автономная некоммерческая образовательная организация
высшего образования
«Региональный финансово-экономический институт»

Кафедра экономики и управления

Утверждаю
Декан экономического факультета
_____ Ю.И. Петренко
«29» мая 2020 г.

Рабочая программа дисциплины

«ПРОГРАММИРОВАНИЕ»

Направление подготовки **38.03.05 Бизнес-информатика**
Профиль **Информационный бизнес**
Квалификация (степень) **Бакалавр**

Факультет экономический

Заочная форма обучения

Курск 2020

Рецензенты:

Аксенов Сергей Леонидович, доктор экономических наук, профессор кафедры экономики и управления;

Бутова Вера Николаевна, кандидат педагогических наук, доцент кафедры экономики и управления.

Рабочая программа составлена в соответствии с Федеральным государственным образовательным стандартом высшего профессионального образования по направлению подготовки 38.03.05 Бизнес-информатика, утвержденного приказом Министерства образования и науки Российской Федерации от от 11 августа 2016 г. N 1002.

Рабочая программа предназначена для методического обеспечения дисциплины образовательной программы 38.03.05 Бизнес-информатика.

«29» мая 2020 г.

Составитель: _____ Смецкой А.С., ст.преподаватель кафедры экономики и управления

Лист согласования рабочей программы
дисциплины «Программирование»

Направление подготовки 38.03.05: **Бизнес-информатика**
Профиль: **Информационный бизнес**

Квалификация (степень): **Бакалавр**

Факультет экономический

Заочная форма обучения

2020/2021 учебный год

Рабочая программа утверждена на заседании кафедры экономики и управления, протокол № 8 от «29» мая 2020 г.

Зав. кафедрой _____ С.Л. Аксенов

Составитель: _____ А.С. Смецкой

Согласовано:

Начальник УМУ
_____ О.И. Петренко, «29» мая 2020 г.

Библиотекарь
_____ Т.А. Котельникова, «29» мая 2020 г.

Председатель методической комиссии по профилю
_____ В.Н. Бутова, «29» мая 2020 г.

Изменения в рабочей программе
дисциплины «Программирование»
на 2021-2022 уч.год

Утверждаю
Декан экономического факультета
_____ Ю.И. Петренко
«25» августа 2021 г.

В рабочую программу вносятся следующие изменения:
1) внесены изменения в список основной литературы

Рабочая программа утверждена на заседании кафедры экономики и управления, протокол № 1 от «25» августа 2021 г.

Зав. кафедрой _____ С.Л. Аксенов

Согласовано:

Начальник УМУ
_____ О.И. Петренко, «25» августа 2021 г.

Председатель методической комиссии по профилю
_____ В.Н. Бутова, «25» августа 2021 г.

Изменения в рабочей программе
дисциплины «Программирование»
на 2022-2023 уч. год

Утверждаю
Декан экономического факультета
_____ Ю.И. Петренко
«26» августа 2022 г.

В рабочую программу вносятся следующие изменения:

1) внесены изменения в перечень вопросов для самоконтроля по самостоятельно изученным темам;
2) внесены изменения в список интернет-ресурсов.

Рабочая программа утверждена на заседании кафедры экономики и управления, протокол № 1 от «26» августа 2022 г.

Зав. кафедрой _____ С.Л. Аксенов

Согласовано:

Начальник УМУ
_____ О.И. Петренко, «26» августа 2022 г.

Председатель методической комиссии по профилю
_____ В.Н. Бутова, «26» августа 2022 г.

Изменения в рабочей программе
дисциплины «Программирование»
на 2023-2024 уч. год

Утверждаю
Декан экономического факультета
_____ Ю.И. Петренко
«25» августа 2023 г.

В рабочую программу вносятся следующие изменения:

1) внесены изменения в перечень вопросов для самоконтроля по самостоятельно изученным темам.

Рабочая программа утверждена на заседании кафедры экономики и управления, протокол № 1 от «25» августа 2023 г.

Зав. кафедрой _____ С.Л. Аксенов

Согласовано:

Начальник УМУ

_____ О.И. Петренко, «25» августа 2023 г.

Председатель методической комиссии по профилю

_____ В.Н. Бутова, «25» августа 2023 г.

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

1. Цель и задачи изучения дисциплины

Целью изучения дисциплины «Программирование» является получение теоретических знаний и практических навыков в разработке программного и информационного обеспечения для информационных систем с использованием объектно-ориентированных языков, применения архитектурных, алгоритмических и программных решений для разрабатываемого программного обеспечения.

В соответствии с обозначенными целями основными задачами данного курса являются:

1. изучение современных методов, средств, стандартов информатики для решения прикладных задач различных классов;
2. исследование архитектуры информационных систем предприятий и организаций;
3. исследование основ методологии и технологии реинжиниринга, проектирования и аудита прикладных информационных систем различных классов;
4. привить студентам навыки применения современных программно-технических средств для решения прикладных задач различных классов; навыки управления проектами по информатизации прикладных процессов и систем.

2. Перечень планируемых результатов обучения по дисциплине (модулю), соотнесенные с планируемыми результатами освоения образовательной программы.

Процесс изучения дисциплины направлен на формирование следующих общекультурных и профессиональных компетенций:

1. обеспечивающие достижение стратегических целей и поддержку бизнес-процессов (ПК-15);
2. осуществлять планирование и организацию проектной деятельности на основе стандартов управления проектами (ПК-16);
3. проектировать архитектуру электронного предприятия (ПК-17);
4. разрабатывать контент и ИТ-сервисы предприятия и Интернет-ресурсов (ПК-18);
5. использовать основные методы естественнонаучных дисциплин в профессиональной деятельности для теоретического и экспериментального исследования (ПК-19).

В результате изучения дисциплины «Объектно-ориентированный анализ и программирование» студент должен:

Знать:

6. современные методы, средства, стандарты информатики для решения прикладных задач различных классов (З-1);
7. основные элементы Java-приложений (З-2);
8. преобразования типов данных при выполнении операций присваивания, объединения строк, вычисления арифметических выражений и вызова метода (З-3);
9. основные технологии программирования (З-4);

Уметь:

10. осознавать сущность и значение информации в развитии современного общества (У-1);
11. работать с информацией из различных источников (У-2);
12. управлять процессами создания и использования информационных сервисов (У-3);
13. использовать соответствующий математический аппарат и инструментальные средства для обработки, анализа и систематизации информации по изучаемой теме (У-4).

Владеть:

14. навыками определения результата применения любого оператора к операндам любого типа, класса, контекста или видимости или к любой их комбинации (В-1);
15. навыками преобразования типов данных при выполнении операций присваивания, объединения строк, вычисления арифметических выражений и вызова метода (В-2);
16. навыками грамотного составления кода программ и управления доступом к объектам, навыками многопоточного программирования, работы с параллельным доступом и взаимодействием между потоками (В-3).

3. Место дисциплины в структуре ООП

Дисциплина включена в базовую часть профессионального цикла ООП.

Для изучения дисциплины необходимы компетенции, сформированные у обучающихся в результате изучения предметов «Информатика» и «Алгебра» в средней образовательной школе. Обучающиеся используют знания, умения, сформированные в ходе изучения дисциплин «Дискретная математика», «Математический анализ».

Знания, умения и виды деятельности, сформированные в результате изучения дисциплины «Программирование» потребуются при изучении дисциплин: «Базы данных», «Электронный бизнес», «Функциональное программирование и интелектуальные системы», «Имитационное моделирование», а также при изучении других дисциплин вариативной части профессионального цикла.

СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

4. Содержание дисциплины (модуля), структурированное по темам (разделам) с указанием отведенного на них количества академических или астрономических часов и видов учебных занятий

Схема распределения учебного времени по видам учебной работы

Общая трудоемкость дисциплины при заочной форме обучения – 6 зачетных единиц (216 академических часов)

Схема распределения учебного времени по семестрам

Виды учебной работы	**Трудоемкость, час**	
	1 курс	**Всего:**
Общая трудоемкость	216	216
Аудиторная работа	14	14
в том числе: лекции практические занятия лабораторные занятия	 4 6 4	 4 6 4
Самостоятельная работа	193	193
Промежуточная аттестация (экзамен)	9	9

Тематический план

№ п/п	Разделы и темы дисциплины	Общая трудоемкость, час	В том числе аудиторных				Самостоятельная работа	Промежуточная аттестация
			всего	из них:				
				лекц	лабор	практ		
1	**Введение в программирование на java**	20	2	1		1	18	
2	**Основы языка**	20	2		1	1	18	
3	**Операторы и присваивания**	18					18	
4	**Объявления и управление доступом**	20	2	1		1	18	
5	**Поток команд управления, управление исключениями и диагностические утверждения**	18	2		1	1	16	
6	**Объектно-ориентированное программирование**	18	1	1			17	
7	Вложенные классы и интерфейсы	18					18	
8	Время жизни объекта	18					18	
9	**Потоки выполнения**	19	1		1		18	
10	**Основные классы**	19	1			1	18	
11	**Коллекции и карты**	21	3	1	1	1	18	
	Промежуточный контроль (экзамен)	9						
	Итого	**216**	**14**	**4**	**4**	**6**	**193**	**9**

Структура и содержание дисциплины

1. Введение в программирование на java.

Введение в концепции объектно-ориентированного программирования, процедуру компилирования и выполнения приложений на Java. Классы

Члены классов: поля и методы. Объекты. Объектные ссылки. Вызовы методов. Терминология для членов класса. Наследование. Агрегация.

JAVA-ПРОГРАММЫ. ОСНОВНЫЕ ПОНЯТИЯ.

Литература:
Основная – 1; 2.
Дополнительная – 1, 2, 3.
Интернет-ресурс: электронный курс для студентов направления «Бизнес-информатика» http://it.rfei.ru/~q
Образовательные технологии, методы и формы обучения: дистанционные образовательные технологии, объяснительно-иллюстративного обучения с элементами проблемного изложения; развивающего обучения, проблемная лекция, практическое занятие, контекстное обучение, ИТ-методы.
Формируемые компетенции: ПК-15; ПК-16; ПК-17.
Образовательные результаты: З-1; З-2; З-3; У-1; У-2; В-1.

2. Основы языка

Основные элементы языка. Константы. Символьные константы. Пробелы и комментарии. Примитивные типы данных. Объявления переменных. Первоначальные значения для переменных. Структура файла исходного кода Java.

Литература:
Основная – 1; 2.
Дополнительная – 1, 2, 3.
Интернет-ресурс: электронный курс для студентов направления «Бизнес-информатика» http://it.rfei.ru/~q
Образовательные технологии, методы и формы обучения: дистанционные образовательные технологии, объяснительно-иллюстративного обучения с элементами проблемного изложения; развивающего обучения, проблемная лекция, практическое занятие, контекстное обучение, ИТ-методы.
Формируемые компетенции: ПК-15; ПК-16; ПК-17.
Образовательные результаты: З-1; З-2; З-3; У-1; У-2; В-1.

3. Операторы и присваивания

ПРИОРИТЕТ И ПРАВИЛА АССОЦИАТИВНОСТИ ДЛЯ ОПЕРАТОРОВ. ПОРЯДОК ВЫЧИСЛЕНИЯ ОПЕРАНДОВ. ЧИСЛОВЫЕ РАСШИРЕНИЯ. ПРОСТОЙ ОПЕРАТОР ПРИСВАИВАНИЯ. АРИФМЕТИЧЕСКИЕ ОПЕРАТОРЫ. УНАРНЫЕ АРИФМЕТИЧЕСКИЕ ОПЕРАТОРЫ. БИНАРНЫЕ ОПЕРАТОРЫ. ОПЕРАТОРЫ ОТНОШЕНИЯ. ССЫЛКИ НА ОБЪЕКТЫ. БУЛЕВЫ ЛОГИЧЕСКИЕ ОПЕРАТОРЫ. ЦЕЛОЧИСЛЕННЫЕ ПОРАЗРЯДНЫЕ ОПЕРАТОРЫ. ОПЕРАТОРЫ СДВИГА. ТРИНАРНЫЙ УСЛОВНЫЙ ОПЕРАТОР. ПЕРЕДАЧА ПАРАМЕТРОВ. ПЕРЕДАЧА ССЫЛОК НА МАССИВ. УПРАЖНЕНИЯ НА ПРОГРАММИРОВАНИЕ.

Литература:
Основная – 1; 2.
Дополнительная – 1, 2, 3.
Интернет-ресурс: электронный курс для студентов направления «Бизнес-информатика» http://it.rfei.ru/~q

Образовательные технологии, методы и формы обучения: дистанционные образовательные технологии, объяснительно-иллюстративного обучения с элементами проблемного изложения; развивающего обучения, проблемная лекция, практическое занятие, контекстное обучение, ИТ-методы.
Формируемые компетенции: ПК-15; ПК-16; ПК-17, ПК-18.
Образовательные результаты: З-1; З-2; З-3; У-1; У-2; В-1, В-2.

4. Объявления и управление доступом

Массивы. Инициализация и использование массивов. Многомерные массивы. Классы и методы. Конструкторы. Правила видимости. Пакеты. Модификаторы доступа для классов и интерфейсов верхнего уровня. Другие модификаторы для классов. Модификаторы доступа членов класса. Другие модификаторы для членов.

Литература:
Основная – 1; 2.
Дополнительная – 1, 2, 3.
Интернет-ресурс: электронный курс для студентов направления «Бизнес-информатика» http://it.rfei.ru/~q
Образовательные технологии, методы и формы обучения: дистанционные образовательные технологии, объяснительно-иллюстративного обучения с элементами проблемного изложения; развивающего обучения, проблемная лекция, практическое занятие, контекстное обучение, ИТ-методы.
Формируемые компетенции: ПК-15; ПК-16; ПК-17, ПК-18.
Образовательные результаты: З-1; З-2; З-3; У-1; У-2; В-1, В-2.

5. Поток команд управления, управление исключениями и диагностические утверждения

Операторы выбора. Операторы цикла. Команды перехода. Исключения в Java. Типы исключений. Обработка исключений. Оператор throw. Assertions. Применение диагностических утверждений. Управление исключениями и диагностические утверждения

Литература:
Основная – 1; 2.
Дополнительная – 1, 2, 3.
Интернет-ресурс: электронный курс для студентов направления «Бизнес-информатика» http://it.rfei.ru/~q
Образовательные технологии, методы и формы обучения: дистанционные образовательные технологии, объяснительно-иллюстративного обучения с элементами проблемного изложения; развивающего обучения, проблемная лекция, практическое занятие, контекстное обучение, ИТ-методы.
Формируемые компетенции: ПК-15; ПК-16; ПК-17, ПК-18.
Образовательные результаты: З-1; З-2; З-3; У-1; У-2; В-1, В-2.

6. Объектно-ориентированное программирование

Одиночное наследование реализации. Концепции ООП. Переопределение и сокрытие методов. Ключевые слова this() и super(). Интерфейсы. Иерархия типов. Преобразование типов значений ссылок. Полиморфизм и динамический поиск метода. Наследование в сравнении с агрегацией

Литература:
Основная – 1; 2.
Дополнительная – 1, 2, 3.
Интернет-ресурс: электронный курс для студентов направления «Бизнес-информатика» http://it.rfei.ru/~q
Образовательные технологии, методы и формы обучения: дистанционные образовательные технологии, объяснительно-иллюстративного обучения с элементами проблемного изложения; развивающего обучения, проблемная лекция, практическое занятие, контекстное обучение, ИТ-методы.
Формируемые компетенции: ПК-15; ПК-16; ПК-17, ПК-18, ПК-19.
Образовательные результаты: З-1; З-2; З-3; У-1; У-2; В-1, В-2.

7. Вложенные классы и интерфейсы

Обзор вложенных классов и интерфейсов. Классы и интерфейсы как статические члены. Классы как нестатические члены. Неявная ссылка на объемлющий контексы. Локальные классы. Безымянные классы. Упражнение на программирование

Литература:
Основная – 1; 2.
Дополнительная – 1, 2, 3.
Интернет-ресурс: электронный курс для студентов направления «Бизнес-информатика» http://it.rfei.ru/~q
Образовательные технологии, методы и формы обучения: дистанционные образовательные технологии, объяснительно-иллюстративного обучения с элементами проблемного изложения; развивающего обучения, проблемная лекция, практическое занятие, контекстное обучение, ИТ-методы.
Формируемые компетенции: ПК-15; ПК-16; ПК-17, ПК-18, ПК-19.
Образовательные результаты: З-1; З-2; З-3; У-1; У-2; В-1, В-2.

8. Время жизни объекта

Сборка мусора. Финализация объекта. Инициализаторы. Блоки статических инициализаторов. Конструирование первоначального состояния объекта.

Литература:
Основная – 1; 2.
Дополнительная – 1, 2, 3.
Интернет-ресурс: электронный курс для студентов направления «Бизнес-информатика» http://it.rfei.ru/~q
Образовательные технологии, методы и формы обучения: дистанционные образовательные технологии, объяснительно-иллюстративного обучения с

элементами проблемного изложения; развивающего обучения, проблемная лекция, практическое занятие, контекстное обучение, ИТ-методы.
Формируемые компетенции: ПК-15; ПК-16; ПК-17, ПК-18, ПК-19.
Образовательные результаты: З-1; З-2; З-3; У-1; У-2; В-1, В-2.

9. Потоки выполнения

Многозадачность. Обзор потоков. Синхронизация. Переходы потоков между состояниями. Приоритеты потоков. Ожидание и уведомление. Блокирование из-за ввода/вывода.

Литература:
Основная – 1; 2.
Дополнительная – 1, 2, 3.
Интернет-ресурс: электронный курс для студентов направления «Бизнес-информатика» http://it.rfei.ru/~q
Образовательные технологии, методы и формы обучения: дистанционные образовательные технологии, объяснительно-иллюстративного обучения с элементами проблемного изложения; развивающего обучения, проблемная лекция, практическое занятие, контекстное обучение, ИТ-методы.
Формируемые компетенции: ПК-15; ПК-16; ПК-17, ПК-18, ПК-19.
Образовательные результаты: З-1; З-2; З-3; У-1; У-2; В-1, В-2.

10. Основыне классы

Класс Object. Классы-оболочки. Класс Math. Класс String. Чтение символов из строки. Класс StringBuffer

Литература:
Основная – 1; 2.
Дополнительная – 1, 2, 3.
Интернет-ресурс: электронный курс для студентов направления «Бизнес-информатика» http://it.rfei.ru/~q
Образовательные технологии, методы и формы обучения: дистанционные образовательные технологии, объяснительно-иллюстративного обучения с элементами проблемного изложения; развивающего обучения, проблемная лекция, практическое занятие, контекстное обучение, ИТ-методы.
Формируемые компетенции: ПК-15; ПК-16; ПК-17, ПК-18, ПК-19.
Образовательные результаты: З-1; З-2; З-3; У-1; У-2; В-1, В-2.

11. Коллекции и карты

Инструментальный набор коллекций. Множества. Списки. Карты. Отсортированные множества и отсортированные карты. Реализация метода equals(). Реализация методов hashCode() и compareTo(). Декораторы коллекций. Другие вспомогательные методы класса Collections.

Литература:
Основная – 1; 2.
Дополнительная – 1, 2, 3.
Интернет-ресурс: электронный курс для студентов направления «Бизнес-

информатика» http://it.rfei.ru/~q
Образовательные технологии, методы и формы обучения: дистанционные образовательные технологии, объяснительно-иллюстративного обучения с элементами проблемного изложения; развивающего обучения, проблемная лекция, практическое занятие, контекстное обучение, ИТ-методы.
Формируемые компетенции: ПК-15; ПК-16; ПК-18, ПК-19.
Образовательные результаты: З-1; З-2; З-3; З-4; У-1; У-2; У-3; У-4; В-1, В-2, В-3.

ПРАКТИЧЕСКИЕ ЗАНЯТИЯ

Практическое занятие № 1

Тема: «Введение в программирование на JAVA»

Цель: формирование у студентов теоретических знаний и практических навыков программирования на языке java; формирование общекультурных и профессиональных компетенций.

Вопросы и задания для практических занятий:

1. **Введение в концепции объектно-ориентированного программирования, процедуру компилирования и выполнения приложений на Java.**
2. **Классы**
3. Члены классов: поля и методы.
4. Объекты. Объектные ссылки.
5. Вызовы методов.
6. Терминология для членов класса.
7. Наследование.
8. Агрегация.
9. Java-программы.

Литература:
Основная – 1; 2.
Дополнительная – 1, 2, 3.
Интернет-ресурс: электронный курс для студентов направления «Бизнес-информатика» http://it.rfei.ru/~q
Образовательные технологии, методы и формы обучения: дистанционные образовательные технологии, объяснительно-иллюстративного обучения с элементами проблемного изложения; развивающего обучения, проблемная лекция, практическое занятие, контекстное обучение, ИТ-методы.
Формируемые компетенции: ПК-15; ПК-16; ПК-17.
Образовательные результаты: З-1; З-2; З-3; У-1; У-2; В-1.
ормы контроля, оценочные средства: текущий контроль: опрос, самостоятельная работа, выполнение тестовых заданий по теории.

Практическое занятие № 2

Тема: «Основы языка»

Цель: формирование у студентов теоретических знаний и практических навыков по работе с языком программирования java; формирование общекультурных и профессиональных компетенций.

***Вопросы и задания для практических занятий**:*

1. **определять правильно построенные объявления пакета, инструкции import, объявления класса (из всех форм, включая внутренние классы), объявления интерфейса, объявления метода (включая метод main, который служит стартовой точкой выполнения программы), объявления переменных и идентификаторы;**
2. **определять классы, которые правильно реализуют интерфейс, будь то интерфейс java.lang.Runnable либо другой явно указанный в вопросе интерфейс;**
3. **обозначать соответствие между значениями индексов массива аргументов, переданного в метод main(), и аргументами командной строки;**
4. **определять все ключевые слова языка программирования Java;**
5. **определять эффект от использования переменной или элемента массива любого типа, если не было выполнено явное присвоение;**
6. **знать диапазон для всех примитивных типов данных и объявлять символьные значения для строк String и всех примитивных типов, используя все разрешенные форматы, основы и представления..**

Литература:
Основная – 1; 2.
Дополнительная – 1, 2, 3.
Интернет-ресурс: электронный курс для студентов направления «Бизнес-информатика» http://it.rfei.ru/~q
Образовательные технологии, методы и формы обучения: дистанционные образовательные технологии, объяснительно-иллюстративного обучения с элементами проблемного изложения; развивающего обучения, проблемная лекция, практическое занятие, контекстное обучение, ИТ-методы.
Формируемые компетенции: ПК-15; ПК-16; ПК-17.
Образовательные результаты: З-1; З-2; З-3; У-1; У-2; В-1.
Формы контроля, оценочные средства: текущий контроль: опрос, самостоятельная работа, выполнение тестовых заданий по теории.

Практическое занятие № 3

Тема: «Операторы и присваивания»

Цель: формирование у студентов понимания приоритета операторов и ассоциативных правил, умение проводить различие между преобразованиями, среди которых преобразование типов, численные преобразования, ведущие к расширению типов, и численные преобразования, ведущие к сужению; определять правила унарного числового расширения и

двоичного числового расширения и контексты, в которых они применяются; понимать преобразования типов для примитивных типов данных при выполнении операций присваивания, объединения строк, вычисления арифметических выражений и вызова метода; формирование общекультурных и профессиональных компетенций.

***Вопросы и задания для практических занятий**:*

1. Определить результат применения любого оператора (включая операторы присваивания и оператор instanceof) к операндам любого типа, класса, контекста или видимости или в их любой комбинации;
2. Определить результат применения логического метода equals (Object) к объектам в любой комбинации классов Java.lang.String, Java.lang.Boolean и Java.lang.Object;
3. В выражении, содержащем операторы &, |, &&, || и переменные с известными значениями, определить, какие операнды вычисляются и само значение выражения;
4. Определить эффект передачи объектов и примитивных значений в методы и выполнения присваивания или другого изменения в этом методе.

Литература:
Основная – 1; 2.
Дополнительная – 1, 2, 3.
Интернет-ресурс: электронный курс для студентов направления «Бизнес-информатика» http://it.rfei.ru/~q
Образовательные технологии, методы и формы обучения: дистанционные образовательные технологии, объяснительно-иллюстративного обучения с элементами проблемного изложения; развивающего обучения, проблемная лекция, практическое занятие, контекстное обучение, ИТ-методы.
Формируемые компетенции: ПК-15; ПК-16; ПК-17.
Образовательные результаты: З-1; З-2; З-3; З-4; У-1; У-2; У-3; У-4; В-1, В-2.
Формы контроля, оценочные средства: текущий контроль: опрос, самостоятельная работа, выполнение тестовых заданий по теории.

Практическое занятие № 4

Тема: «Объявления и управление доступом»

Цель. Определить и научиться использовать анонимные массивы. Понять использование ссылки this в методах экземпляра. Понять термин сигнатура метода и сформулировать условия, при которых метод может быть перегружен. Определить конструктор не по умолчанию и установить, как можно перегружать конструкторы. Определить, как операторы package и import используются для описания пакетов и импорта из пакетов; формирование общекультурных и профессиональных компетенций.

***Вопросы и задания для практических занятий**:*

1. Написать код, который объявляет, создает и инициализирует массивы любого базового типа, пользуясь любым из допустимых способов как для объявления, так и для инициализации;

2. Объявить классы, внутренние классы, методы, переменные экземпляра, статические переменные и автоматические переменные (локальные переменные метода), применяя подходящие разрешенные модификаторы (такие, как public, final, static, abstract);
3. Формулировать важность каждого из этих модификаторов - как по отдельности, так и в комбинации - и понимать влияние пакетных отношений на объявленные элементы, содержащие эти модификаторы;
4. Определить для заданного класса, будет ли для него создан конструктор по умолчанию, и если да, представлять прототип этого конструктора.
5. Определить разрешенные возвращаемые типы для каждого метода по заданным объявлениям всех необходимых методов в этом или родительских классах.

Литература:
Основная – 1; 2.
Дополнительная – 1, 2, 3.
Интернет-ресурс: электронный курс для студентов направления «Бизнес-информатика» http://it.rfei.ru/~q
Образовательные технологии, методы и формы обучения: дистанционные образовательные технологии, объяснительно-иллюстративного обучения с элементами проблемного изложения; развивающего обучения, проблемная лекция, практическое занятие, контекстное обучение, ИТ-методы.
Формируемые компетенции: ПК-15; ПК-16; ПК-17, ПК-18.
Образовательные результаты: З-1; З-2; З-3; З-4; У-1; У-2; У-3; У-4; В-1, В-2.
Формы контроля, оценочные средства: текущий контроль: опрос, самостоятельная работа, выполнение тестовых заданий по теории, выполнение практических заданий.

Практическое занятие № 5

Тема: «Поток команд управления, управление исключениями и диагностические утверждения»

Цель: Знать имена основных классов в иерархии наследования классов исключений. Проводить различие между проверяемыми (checked) и непроверяемыми исключениями. Понимать передачу исключения в стеке выполнения.; формирование общекультурных и профессиональных компетенций.

***Вопросы и задания для практических занятий**:*

1. использовать в коде операторы if и switch и устанавливать допустимые типы аргументов для этих операторов;
2. создавать код с помощью всех видов циклов, включая маркированное (labeled) и немаркированное применение break и continue, определять значения, принимаемые переменными счетчика цикла во время и после выполнения цикла;

3. создавать код, правильно применяя исключения и выражения управления исключениями (try - catch - finally), а также с объявлением методов и переписыванием тех, которые выбрасывают исключения;
4. определять результат исключения, возникающего в заданной точке фрагмента кода. Обратите внимание, что исключение может быть исключением времени выполнения, проверяемым исключением или ошибкой (код может содержать выражения try , catch или finally в любой допустимой комбинации);
5. создавать код, правильно используя диагностические утверждения, а также уметь отличать правильную форму их применения от неправильной;
6. определять правильные утверждения, связанные с механизмом диагностики утверждений (assertion mechanism).

Литература:
Основная – 1; 2.
Дополнительная – 1, 2, 3.
Интернет-ресурс: электронный курс для студентов направления «Бизнес-информатика» http://it.rfei.ru/~q
Образовательные технологии, методы и формы обучения: дистанционные образовательные технологии, объяснительно-иллюстративного обучения с элементами проблемного изложения; развивающего обучения, проблемная лекция, практическое занятие, контекстное обучение, ИТ-методы.
Формируемые компетенции: ПК-15; ПК-16; ПК-17, ПК-18.
Образовательные результаты: З-1; З-2; З-3; З-4; У-1; У-2; У-3; У-4; В-1, В-2.
Формы контроля, оценочные средства: текущий контроль: опрос, самостоятельная работа, выполнение тестовых заданий по теории, выполнение практических заданий.

Практическое занятие № 6

Тема: «Объектно-ориентированное программирование»

Цель: формирование у студентов теоретических знаний и практических навыков в концепции единичного наследования, множественного наследования, отношения подтип-супертип и их использование в объектно-ориентированном программировании (ООП).Понимать цепочку конструкторов, включающую элементы this() и super(). Обозначать правила преобразования для присваивания, явного или неявного преобразования типа или передачи ссылок. Определять на этапе выполнения, является ли объект экземпляром заданного типа ссылки (или подтипом данного типа), с помощью оператора instanceof. Понимать полиморфизм и динамический поиск метода.; формирование общекультурных и профессиональных компетенций.

***Вопросы и задания для практических занятий**:*

1. обозначать преимущества инкапсуляции в объектно-ориентированном проектировании и писать код, реализующий инкапсулированные классы и отношения «есть» (is-a) и «имеет» (has-a);

2. писать код, вызывающий переопределенные или перегруженные методы и родительские или перегруженные конструкторы; описывать результат вызова этих методов;
3. писать код, создающий экземпляры любого конкретного класса, который может быть обычным классом верхнего уровня или вложенным классом.

Литература:
Основная – 1; 2.
Дополнительная – 1, 2, 3.
Интернет-ресурс: электронный курс для студентов направления «Бизнес-информатика» http://it.rfei.ru/~q
Образовательные технологии, методы и формы обучения: дистанционные образовательные технологии, объяснительно-иллюстративного обучения с элементами проблемного изложения; развивающего обучения, проблемная лекция, практическое занятие, контекстное обучение, ИТ-методы.
Формируемые компетенции: ПК-15; ПК-16; ПК-17, ПК-18, ПК-19.
Образовательные результаты: З-1; З-2; З-3; З-4; У-1; У-2; У-3; У-4; В-1, В-2, В-3.
Формы контроля, оценочные средства: текущий контроль: опрос, самостоятельная работа, выполнение тестовых заданий по теории, выполнение практических заданий.

Практическое занятие № 7

Тема: «Вложенные классы и интерфейсы»

Цель: формирование у студентов практических навыков по написанию кода, позволяющего производить различные операции с классами; формирование общекультурных и профессиональных компетенций.

Вопросы и задания для практических занятий:

1. Написать код, который создает экземпляры некоторого конкретного класса, включая обычные классы верхнего уровня и вложенные классы;
2. Перечислить виды вложенных классов и интерфейсов, которые можно определить;
3. Определить контекст, в котором может быть определен вложенный класс или интерфейс;
4. Определить, какие модификаторы доступа разрешены для каждой категории вложенных классов;
5. Определить, какие вложенные классы создают экземпляры, связанные с экземплярами содержащего их контекста;
6. Сформулировать правила доступа, которые влияют на доступность элементов в содержащем их контексте вложенных классов, и писать код, который использует расширенный синтаксис, включающий ключевое слово this для этой цели. Определять, может ли описание вложенного класса содержать статические и нестатические члены;

7. Написать код, порождающий объекты вложенных классов, используя расширенный синтаксис оператора new;
8. Написать код, демонстрирующий импорт и использование вложенных классов;
9. Провести различия между иерархией наследования и контекстом, объемлющим внутренний класс или интерфейс;
10. Написать код для реализации безымянных (anonymous) классов, используя расширение существующего класса или реализацию интерфейса.

Литература:
Основная – 1; 2.
Дополнительная – 1, 2, 3.
Интернет-ресурс: электронный курс для студентов направления «Бизнес-информатика» http://it.rfei.ru/~q
Образовательные технологии, методы и формы обучения: дистанционные образовательные технологии, объяснительно-иллюстративного обучения с элементами проблемного изложения; развивающего обучения, проблемная лекция, практическое занятие, контекстное обучение, ИТ-методы.
Формируемые компетенции: ПК-15; ПК-16; ПК-17, ПК-18, ПК-19.
Образовательные результаты: З-1; З-2; З-3; З-4; У-1; У-2; У-3; У-4; В-1, В-2.
Формы контроля, оценочные средства: текущий контроль: опрос, самостоятельная работа, выполнение тестовых заданий по теории, выполнение практических заданий.

Практическое занятие № 8

Тема: «Время жизни объекта»

Цель: формирование у студентов практических навыков по написанию кода, в котором используется правильный механизм завершения для освобождения ресурсов, перед тем как объект будет собран сборщиком мусора; различать статические инициализаторы и инициализаторы экземпляра, выражения и блоки инициализации, а также то, как они используются для инициализации объекта и инициализации класса; представлять этапы, выполняемые при инициализации состояния объекта, когда объект создается с помощью оператора new; формирование общекультурных и профессиональных компетенций.

Вопросы и задания для практических занятий:

1. Определить поведение, которое гарантируется системой сборки мусора;
2. Написать код, который явно создает объекты, пригодные для сборки мусора;
3. Распознавать место в исходном коде, в котором объект становится пригодным для сборки мусора.

Литература:
Основная – 1; 2.

Дополнительная – 1, 2, 3.
Интернет-ресурс: электронный курс для студентов направления «Бизнес-информатика» http://it.rfei.ru/~q
Образовательные технологии, методы и формы обучения: дистанционные образовательные технологии, объяснительно-иллюстративного обучения с элементами проблемного изложения; развивающего обучения, проблемная лекция, практическое занятие, контекстное обучение, ИТ-методы.
Формируемые компетенции: ПК-15; ПК-16; ПК-17, ПК-18, ПК-19.
Образовательные результаты: З-1; З-2; З-3; З-4; У-1; У-2; У-3; У-4; В-1, В-2.
Формы контроля, оценочные средства: текущий контроль: опрос, самостоятельная работа, выполнение тестовых заданий по теории, выполнение практических заданий.

Практическое занятие № 9

Тема: «Потоки выполнения»

Цель: формирование у студентов практических навыков по написанию кода, позволяющего работать с потоками; формирование общекультурных и профессиональных компетенций.

***Вопросы и задания для практических занятий**:*

1. Написать код, который определяет, инстанцирует и запускает новые потоки с помощью и Java.lang.Thread, и Java.lang.Runnable;
2. Определить условия, которые могут помешать выполнению потока;
3. Написать код, применяя методы synchronized, wait(), notify() и notifyAll() для того, чтобы защититься от проблем с параллельным доступом и взаимодействием между потоками;
4. Определить взаимодействие между потоками и блокировками объекта, когда выполняются методы synchronized, wait(), notify() или notifyAll().

Литература:
Основная – 1; 2.
Дополнительная – 1, 2, 3.
Интернет-ресурс: электронный курс для студентов направления «Бизнес-информатика» http://it.rfei.ru/~q
Образовательные технологии, методы и формы обучения: дистанционные образовательные технологии, объяснительно-иллюстративного обучения с элементами проблемного изложения; развивающего обучения, проблемная лекция, практическое занятие, контекстное обучение, ИТ-методы.
Формируемые компетенции: ПК-15; ПК-16; ПК-17, ПК-18, ПК-19.
Образовательные результаты: З-1; З-2; З-3; З-4; У-1; У-2; У-3; У-4; В-1, В-2.
Формы контроля, оценочные средства: текущий контроль: опрос, самостоятельная работа, выполнение тестовых заданий по теории, выполнение практических заданий.

Практическое занятие № 10

Тема: «Основные классы»

Цель: формирование у студентов практических навыков по написанию кода, позволяющего работать с классами и классами-оболочками; формирование общекультурных и профессиональных компетенций.

***Вопросы и задания для практических занятий**:*

1. Написать код с помощью следующих методов класса java.lang.Math: abs(), ceil(), floor(), max(), min(), random(), round(), sin(), cos(), tan(), sqrt();
2. СФормулировать важность неизменности объектов типа String;
3. понимать значимость классов-оболочек, включая создание подходящих вариантов из классов-оболочек для удовлетворения требованиям определенного поведения, определять результат выполнения фрагмента кода, который включает экземпляр одного из классов-оболочек, и писать код с помощью следующих методов классов-оболочек (т.е. Integer, Double и т.д.):

doubleValue()
floatValue()
intValue()
longValue()
parseXxx()
getXxx()
toString()
toHexString();

4. Понимать функциональность, унаследованную всеми классами от класс Object, который находится на вершине иерархии наследования;
5. Написать код для управления неизменными и динамическими строками, используя возможности, предоставляемые классами String И StringBuffer соответственно.

Литература:
Основная – 1; 2.
Дополнительная – 1, 2, 3.
Интернет-ресурс: электронный курс для студентов направления «Бизнес-информатика» http://it.rfei.ru/~q
Образовательные технологии, методы и формы обучения: дистанционные образовательные технологии, объяснительно-иллюстративного обучения с элементами проблемного изложения; развивающего обучения, проблемная лекция, практическое занятие, контекстное обучение, ИТ-методы.
Формируемые компетенции: ПК-15; ПК-16; ПК-17, ПК-18, ПК-19.
Образовательные результаты: З-1; З-2; З-3; З-4; У-1; У-2; У-3; У-4; В-1, В-2.
Формы контроля, оценочные средства: текущий контроль: опрос, самостоятельная работа, выполнение тестовых заданий по теории, выполнение практических заданий.

Практическое занятие № 11

Тема: «Коллекции и карты»

Цель: формирование у студентов практических навыков по написанию

кода, позволяющего работать с коллекциями и картами; формирование общекультурных и профессиональных компетенций.

***Вопросы и задания для практических занятий**:*

1. выбрать подходящие классы/интерфейсы, удовлетворяющие определенным требованиям поведения;
2. знать различия между правильными и неправильными реализациями методов hashCode() и eguals();
3. идентифицировать базовые интерфейсы коллекций и отношения наследования между ними: Collection, Set, SortedSet, List, Map, SortedMap;
4. сформулировать различия между коллекциями и картами;
5. объяснять действия, выполняемые методами addAll(), removeAll() и retainAll() интерфейса Collection;
6. распознать разрушительные и не приводящие к разрушению информации действия коллекций;
7. понимать, как данные передаются между коллекциями;
8. идентифицировать реализации и понимать использование основных интерфейсов коллекций: HashSet, TreeSet, LinkedHashSet, ArrayList, Vector, LinkedList, HashMap, Hashtable, TreeMap, LinkedHashMap;
9. написать код, чтобы получить представления о коллекциях. использовать итератор для обхода элементов коллекции;
10. знать различия между неупорядоченными, упорядоченными и сортированными списками;
11. понимать естественную упорядоченность элементов, обеспечиваемую интерфейсом Comparable, и всеобщее упорядочивание, навязываемое интерфейсом Comparator;
12. обозначать условия, при которых может быть выброшено исключение UnsupportedOperationException;
13. знать методы в классе Collections, которые можно использовать для создания неизменных и потокобезопасных версий разных типов коллекций;
14. знать главные алгоритмы, предоставленные классом Collections;
15. писать код для передачи данных между массивами и коллекциями.

Литература:
Основная – 1; 2.
Дополнительная – 1, 2, 3.
Интернет-ресурс: электронный курс для студентов направления «Бизнес-информатика» http://it.rfei.ru/~q
Образовательные технологии, методы и формы обучения: дистанционные образовательные технологии, объяснительно-иллюстративного обучения с элементами проблемного изложения; развивающего обучения, проблемная лекция, практическое занятие, контекстное обучение, ИТ-методы.
Формируемые компетенции: ПК-15; ПК-16; ПК-17, ПК-18, ПК-19.
Образовательные результаты: З-1; З-2; З-3; З-4; У-1; У-2; У-3; У-4; В-1, В-2, В-3.

Формы контроля, оценочные средства: текущий контроль: опрос, самостоятельная работа, выполнение тестовых заданий по теории, выполнение практических заданий.

5. Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине (модулю).

Предполагает более углубленное изучение отдельных тем дисциплины “Объектно-ориентированный анализ и программирование”, выполнение следующих заданий для самоконтроля:

Тема “Основные понятия языка программирования java”

Задание 1

Выберите верное утверждение относительно метода.

1. Метод - это форма для создания операций.
2. Метод - это категория объектов.
3. Метод - это атрибут, определяющий свойство конкретной абстракции.
4. Метод реализует абстракцию.
5. Метод - это действие, определяющее поведение для конкретной абстракции.

Задание 2

Выберите верное утверждение относительно объекта.

1. Объект - это то, из чего рождается класс.
2. Объект - это экземпляр класса.
3. Объект - это переменная.
4. Объект - это форма для создания конкретной реализации абстракции.
5. Объект - это ссылка на атрибут.

Задание 3

Какая строка содержит конструктор в описании этого класса?

```
public class Counter {     // (1)
 int current, step;
 public Counter(int startValue, int stepValue) {      // (2)
  set(startValue);
  setStepValue(staepValue);
 }
 public int get() { return current; }       // (3)
 public void set(int value) { current = value; }        // (4)
 public void setStepValue(int stepValue){step=stepValue;}     // (5)
}
```

Выберите один правильный ответ.

1. Код, обозначенный (4), - это конструктор.
2. Код, обозначенный (1), - это конструктор.
3. Код, обозначенный (5), - это конструктор.

4. Код, обозначенный (3), - это конструктор.
5. Код, обозначенный (2), - это конструктор.

Задание 4
Учитывая, что Thing — это класс, как много объектов и как много ссылочных переменных создается в следующем программном коде?
```
Thing item, stuff;
item = new Thing();
Thing entity = new Thing();
```
Выберите два правильных ответа.
1. Один объект создается.
2. Два объекта создается.
3. Три объекта создается.
4. Одна ссылочная переменная создается.
5. Две ссылочные переменные создаются.
6. Три ссылочные переменные создаются.

Задание 5
Какое утверждение о члене экземпляра (instance member) верно? Выберите один правильный ответ.
1. Член экземпляра принадлежит экземпляру, а не классу в целом.
2. Член экземпляра также называется статическим членом.
3. Член экземпляра - это всегда поле.
4. Член класса всегда представляет операцию.
5. Член экземпляра никогда не является методом.

Задание 6
Выберите один правильный ответ.
Объекты передают сообщения в Java при помощи
1. вызова методов экземпляра друг друга.
2. вызова статических методов классов друг друга.
3. изменения статических переменных классов друг друга.
4. изменения полей друг друга.

Задание 7
Учитывая следующий код, какое утверждение верно?
```
class A {
 int value1;
}

class B extends A {
 int value2;
}
```
Выберите два правильных ответа.
1. Класс A расширяет класс B.
2. Класс B - это суперкласс для класса A.

3. Класс A наследует от класса B.
4. Класс B - это подкласс класса A.
5. Объекты класса A имеют поле с именем «value 2».
6. Объекты класса B имеют поле с именем «value 1».

Задание 8
Какую команду из Java 2 SDK следует использовать для того, чтобы скомпилировать следующий код, содержащийся в файле SmallProg.java?

```
public class SmallProg {
public static void main (String[] args)
{System.out.println("Goog luck!"); }
}
```

Выберите один правильный ответ.

1. java SmallProg.main
2. javac SmallProg.java
3. java SmallProg
4. java SmallProg.Java
5. javac SmallProg

Задание 9
Какую команду из Java 2 SDK следует использовать для того, чтобы выполнить метод main () класса с именем SmallProg? Выберите один правильный ответ.

1. java SmallProg
2. java SmallProg.java
3. java SmallProg.main()
4. java SmallProg.class
5. javac SmallProg

Тема “Операторы и присваивания”

Задание 1
Каков самый простой способ для преобразования значения символа c в int? Выберите один правильный ответ.

1. int i = (int) c.
2. int i = c.
3. int i = Character.getNumericValues (c).

Задание 2
Что произойдет в результате попытки откомпилировать и выполнить следующий класс?

```
public class Assignment {
   public static void main(String[] args) {
      int a, b, c;
      b = 10;
      a = b = c = 20;
      System.out.println(a);
```

```
}
```

Выберите один правильный ответ.

1. Компиляция выполнится успешно, и после выполнения будет выведено 20.
2. Компиляция выполнится неудачно, так как компилятор определит, что переменная c в операторе присваивания a = b = c = 20 не была проинициализирована.
3. Компиляция выполнится успешно, и после выполнения будет выведено 10.
4. Компиляция выполнится неудачно, потому что оператор присваивания a = b = c = 20 недопустим.

Задание 3

Что произойдет в результате попытки откомпилировать и выполнить следующую программу?

```
public class MyClass {
  public static void main(String[] args) {
     String a, b, c;
     c = new String("mouse");
     a = new String("cat");
     b = a;
     a = new String("dog");
     c = b;
     System.out.println(c);
  }
```

Выберите один правильный ответ.

1. В результате выполнения программы будет выведено mouse.
2. В результате выполнения программы будет выведено dog.
3. В результате выполнения программы будет выведено cat.
4. В результате выполнения программы с равной вероятностью будет выведено либо cat, либо dog.
5. Программа откомпилируется неудачно.

Задание 4

При вычислении какого из следующих выражений будет использоваться вещественная арифметика? Выберите три правильных ответа.

1. 2.0*3.0
2. 2*3
3. 2/3+5/7
4. 2.4+1.6
5. 0x101L300.0

Задание 5

Каково будет значение выражения (1/2+3/2+0,1)? Выберите один правильный ответ.

1. 1
2. 2.1
3. 1.6

4. 1.1
5. 2

Задание 6

Что будет в результате попытки откомпилировать и выполнить эту программу?

```
public class Integers {
   public static void main(String[] args) {
      System.out.println(0x10 + 10 + 010);
   }
```

Выберите один правильный ответ.

1. После выполнения программа напечатает 34.
2. После выполнения программа напечатает 28.
3. Программа не откомпнлнруется. Компилятор выразит недовольство выражением 0x10 + 10 + 010.
4. После выполнения программа напечатает 101010.
5. После выполнения программа напечатает 30.
6. После выполнения программа напечатает 36.

Задание 7

Какое из следующих выражений допустимо? Выберите три правильных ответа.

1. (- 1 -)
2. (+ + 1)
3. (+-+-+-1)
4. (—1)
5. (1 1)
6. (- -1)

Задание 8

Какое будет получено значение после вычисления выражения (- -1-3 * 10 / 5-1)? Выберите один правильный ответ.

1. 8
2. -6
3. 7
4. -8
5. 10

Задание 9

Какое из этих присваиваний недопустимо?

Выберите один правильный ответ.

1. long l = 012;
2. short s = 12;
3. float f = -123;
4. double d = 0x12345678;
5. int other = (int) true;

Задание 10

Какое из утверждений верно?

Выберите три правильных ответа.

1. Результатом выражения (1 + 2 + «3») будет строка «33».
2. Результатом выражения («1» + 2 + 3) будет строка «15».
3. Результатом выражения (4 + 1.0f) будет float-значение 5.0f.
4. Результатом выражения (10/9) будет int - значение 1.
5. Результатом выражения ('a' + 1) будет char - значение 'b'.

Задание 11

Что произойдет в результате попытки откомпилировать и выполнить следующую программу?

```
public class Prog1 {
   public static void main(String[] args) {
      int k = 1;
      int i = ++k + k++ + + k;
      System.out.println(i);
   }
```

Выберите один правильный ответ.

1. Программа не откомпилируется. Компилятор выразит недовольство выражением ++k + k++ + + k.
2. Программа откомпилируется и после выполнения напечатает значение 4.
3. Программа откомпилируется и после выполнения напечатает значение 7.
4. Программа откомпилируется и после выполнения напечатает значение 8.
5. Программа откомпилируется и после выполнения напечатает значение 3.

Задание 12

Какая неверная строка первой вызовет ошибку на этапе компиляции следующей?

```
public class MyClass {
   public static void main(String[] args) {
      char c;
      int i;
      c = 'a'; // (1)
      i = c;   // (2)
      i++;     // (3)
      c = i;   // (4)
      c++;     // (5)
   }
```

Выберите один правильный ответ.

1. Строка (5).
2. Строка (1).
3. Строка (3).
4. Строка (2).
5. Все строки допустимы. Программа откомпилируется успешно.
6. Строка (4).

Задание 13

Что произойдет в результате попытки откомпилировать и выполнить следующую программу?

```
public class Cast {
   public static void main(String[] args) {
      byte b = 128;
      int  i = b;
      System.out.println(i);
   }
```

Выберите один правильный ответ.

10. Программа откомпилируется и после выполнения напечатает 128.
11. Программа откомпилируется и после выполнения напечатает 255.
12. Программа откомпилируется, но на этапе выполнения будет выброшено исключение ClassCastException.
13. Компилятор откажется компилировать программу, потому что не сможет присвоить byte в int без преобразования типа.
14. Компилятор откажется компилировать программу, потому что 128 выходит за рамки допустимого диапазона для значений byte.

Задание 14

Что напечатает следующая программа во время выполнения?

```
public class EvaluationOrder {
   public static void main(String[] args) {
      int[] array = { 4, 8, 16 };
      int i=1;
      array[++i] = --i;
      System.out.println(array[0] + array[1] + array[2]);
   }
```

Выберите один правильный ответ.

1. 13
2. 20
3. 21
4. 24
5. 14

Задание 15

В результате каких из следующих выражений будет получено true?

Выберите два правильных ответа.

1. (false | true).
2. (null != null).
3. (4 <= 4).
4. (! true).
5. (true & false)

Задание 16

Какие утверждения верны?

Выберите три правильных ответа.

10. Оператор получения остатка % может использоваться только с целыми операндами.
11. Идентификаторы в Java чувствительны к регистру.
12. Арифметические операторы *, / и % имеют приоритет одного и того же уровня.
13. Значения типа short находятся в диапазоне от -128 до +127 включительно.
14. (+15) является разрешенным выражением.

Задание 17

Какое из утверждений относительно того, какие строки будут выведены на консоль следующей программой, верно?

```
public class BoolOp {
   static void op(boolean a, boolean b) {
      boolean c = a != b;
      boolean d = a ^ b;
      boolean e = c == d;
      System.out.println(e);
   }

   public static void main(String[] args) {
      op(false, false);
      op(true, false);
      op(false, true);
      op(true, true);
   }
```

Выберите три правильных ответа.

1. Все строки напечатают одно и то же.
2. По крайней мере одна строка будет содержать false.
3. По крайней мере одна строка будет содержать true.
4. Первая строка будет содержать false.
5. Первая строка будет содержать true.

Задание 18

Что произойдет во время выполнения следующей программы? Выберите один правильный ответ.

```
public class OperandOrder {
   public static void main(String[] args) {
      int i = 0;
      int[] a = {3,6};
      a[i] = i = 9;
      System.out.println(i + " " + a[0] + " " + a[1]);
   }
```

6. Будет выброшено исключение ArraylndexOutOfBoundsException.
7. Будет выведено «9 3 9».

8. Будет выведено «9 0 6».
9. Будет выведено «9 3 6».
10. Будет выведено «9 9 6».

Задание 19

Какое из утверждений верно относительно результата работы следующей программы?

```
public class Logic {
   public static void main(String[] args) {
      int i = 0;
      int j = 0;

      boolean t = true;
      boolean r;

      r = (t &  0<(i+=1));
      r = (t && 0<(i+=2));
      r = (t |  0<(j+=1));
      r = (t || 0<(j+=2));
      System.out.println(i + " " + j);
   }
```

Выберите два правильных ответа.

10. Первая напечатанная цифра будет 1
11. Первая напечатанная цифра будет 2
12. Первая напечатанная цифра будет 3
13. Вторая напечатанная цифра будет 1
14. Вторая напечатанная цифра будет 2
15. Вторая напечатанная цифра будет 3

Задание 20

Что будет отображено на экране во время выполнения следующей программы?

```
public class MyClass {
   public static void main(String[] args) {
      test(1<<32, "1<<32");
      test(1<<31, "1<<31");
      test(1<<30, "1<<30");
      test(1,     "1"    );
      test(0,     "0"    );
      test(-1,    "-1"   );
   }

   public static void test(int i, String exp) {
      if ((i >> 1) != (i >>> 1)) System.out.println(exp);
   }
```

Выберите два правильных ответа.

1. «1<<32».

2. «1<<31».
3. «1<<30».
4. «1».
5. «0»
6. «-1»

Задание 21
Что из следующего не является оператором в Java?
Выберите два правильных ответа.

10. %
11. <<<
12. &
13. %=
14. >>>
15. <=
16. &&=

Задание 22
Допустим, что есть переменная x типа int (которая может содержать отрицательное значение). Какие существуют правильные способы удвоения значения x, кроме тех, в которых промежуточные результаты выходят за пределы допустимого диапазона?
Выберите четыре правильных ответа.

1. x << 1;
2. x = x * 2;
3. x *= 2;
4. x += x;
5. x <<= 1;

Задание 23
Какие из следующих операторов могут использоваться в качестве и целочисленного поразрядного оператора, и булевого логического оператора?
Выберите три правильных ответа.

1. ^
2. !
3. &
4. |
5. ~

Задание 24
Ниже заданы объявления, какое из следующих выражений допустимо?

```
byte  b = 1;
char  c = 1;
short s = 1;
int   i = 1;
```

Выберите три правильных ответа.

1. s = b * 2;
2. i = b << s;
3. b <<= s;
4. c = c + b;
5. s += i;

Задание 25

Что будет отображено на экране во время выполнения следующей программы?

```
public class ParameterPass {
   public static void main(String[] args) {
      int i = 0;
      addTwo(i++);
      System.out.println(i);
   }

   static void addTwo(int i) {
      i += 2;
   }
```

Выберите один правильный ответ.

10. 0
11. 2
12. 1
13. 3

Задание 26

Что будет в результате попытки откомпилировать и выполнить следующий класс?

```
public class Passing {
   public static void main(String[] args) {
      int a = 0; int b = 0;
      int[] bArr = new int[1]; bArr[0] = b;

      inc1(a); inc2(bArr);

      System.out.println("a=" + a + " b=" + b + " bArr[0]=" + bArr[0]);
   }

   public static void inc1(int x) { x++; }

   public static void inc2(int[] x) { x[0]++; }
```

Выберите один правильный ответ.

На этапе компиляции возникнет ошибка, так как x[0]++; является недопустимым выражением.

1. Код откомпилируется, и после выполнения будет отображено «a=0 b=1 bArr[0]=1».
2. Код откомпилируется, и после выполнения будет отображено «a=1 b=1 bArr[0]=1».
3. Код откомпилируется, и после выполнения будет отображено «a=0 b=0 bArr[0]=0».
4. Код откомпилируется, и после выполнения будет отображено «a=0 b=0 bArr[0]=1».

Задание 27

Дан класс:

```
// Filename: Args.java
public class Args {
   public static void main(String[] args) {
      System.out.println(args[0] + " " + args[args.length-1]);
   }
}
```

Каков будет результат после выполнения следующей команды?

java Args In politics stupidity is not a handicap

Выберите один правильный ответ.

12. Будет отображено на экране «In handicap».
13. Будет отображено на экране «Java handicap».
14. Будет отображено на экране «Args handicap».
15. Будет выброшено исключение ArraylndexOutOfBoundsException.
16. Будет отображено на экране «Args a».
17. Будет отображено на экране «In a».

Задание 28

Какое выражение должно вызвать ошибку компиляции, если его добавить в приведенную ниже программу в указанное положение?

```
public class ParameterUse {
   static void main(String[] args) {
      int a = 0;
      final int b = 1;
      int[] c = { 2 };
      final int[] d = { 3 };
      useArgs(a, b, c, d);
   }

   static void useArgs(final int a, int b, final int[] c, int[] d) {
      // INSERT STATEMENT HERE.
   }
```

Выберите два правильных ответа.

1. a++;
2. b++;
3. b = a;

4. c[0]++;
5. d[0]++;
6. c = d;

Тема “Объектно-ориентированный анализ”

Задание 1
Какое утверждение верно? Выберите два правильных ответа.

1. В Java оператор extends используется для задания наследования.
2. Подкласс не абстрактного класса может быть объявлен как abstract.
3. Все члены суперкласса наследуются подклассом.
4. Неизменяемый (final) класс может быть абстрактным.
5. Класс, в котором все члены объявлены как private, не может быть объявлен как public.

Задание 2
Какое утверждение верно? Выберите два правильных ответа.

- Наследование задает отношение «имеет» (has-a) между суперклассом и его подклассом.
- Каждый объект Java имеет публичный (public) метод с именем equals.
- Каждый объект Java имеет публичный (public) метод с именем length.
- Класс может расширять любое количество других классов.
- Класс, не являющийся неизменяемым (final), может расширять любое количество классов.

Задание 3
Какое утверждение верно? Выберите 2 правильных ответа.

- В подклассе должны быть определены все методы из суперкласса
- Возможно определить в подклассе метод с таким же именем и параметрами, как и метод в суперклассе
- Возможно определить в подклассе поле с таким же именем, как и поле в суперклассе
- Для двух классов возможно, чтобы каждый из ни был суперклассом другого

Задание 4
Для представленных ниже классов и объявлений какое из утверждений верно? Выберите 3 правильных ответа.

```
// Классы
class Foo {
   private int i;
   public void f() { /* ... */ }
   public void g() { /* ... */ }
```

```
}

class Bar extends Foo {
   public int j;
   public void g() { /* ... */ }
}

// Объявления:
// ...
   Foo a = new Foo();
   Bar b = new Bar();
```

Класс Bar является корректным подклассом Foo

- Выражение b.f(); допустимо
- Выражение a.j = 5; допустимо
- Выражение a.g(); допустимо
- Выражение b.i = 3; допустимо

Задание 5

Какое утверждение верно? Выберите правильный ответ.

- Переопределенный метод может иметь другой возвращаемый тип, чем в методе, который он переопределяет.
- Подкласс может переопределить любой метод суперкласса
- Список параметров переопределенного метода должен быть подмножеством списка параметров метода, который он переопределяет
- Private-методы нельзя переопределить в подклассе
- В переопределенном методе может быть объявлено, что он может выбросить больше исключений, чем метод, который он переопределяет.

Задание 6

Даны классы A, B и C, причем B расширяет A, C расширяет B и все классы реализуют метод void doIt(). Как можно обратиться к методу doIt() в A из метода класса C? Выберите правильный ответ.

- super.doIt();
- ((A) this).doIt();
- this.super.doIt();
- doIt();
- super.super.doIt();
- A.this.doIt();
- Это не возможно

Задание 7

Что получится в результате компилирования и выполнения следующего

кода?

```
public class MyClass {
   public static void main(String[] args) {
      C c = new C();
      System.out.println(c.max(13, 29));
   }
}

class A {
   int max(int x, int y) { if (x>y) return x; else return y; }
}

class B extends A{
   int max(int x, int y) { return super.max(y, x) - 10; }
}

class C extends B {
   int max(int x, int y) { return super.max(x+10, y+10); }
```

Выберите 1 правильный ответ.

- Компиляция не будет выполнена успешно, потому что метод max() в B передает аргументы при вызове super.max(y, x) в неверном порядке
- Компиляция выполнится успешно, и во время выполнения будет напечатано 29
- Компиляция не будет выполнена успешно, потому что вызов метода max() неоднозначен
- Компиляция выполнится успешно, и во время выполнения будет напечатано 13
- Компиляция выполнится успешно, и во время выполнения будет напечатано 39
- Компиляция выполнится успешно, и во время выполнения будет напечатано 23

Задание 8

Для данного ниже кода какой самый простой оператор print может быть добавлен в метод print()?

```
public class MyClass extends MySuperclass {
   public static void main(String[] args) {
      MyClass object = new MyClass();
      object.print();
   }
   public void print() {
      // Добавьте сюда код который напечатает
      // строку "Hello, world!" из класса Message
```

```
        }
}

class MySuperclass {
   Message msg = new Message();
}

class Message {
   // Сообщение, которое следует напечатать:
   String text = "Hello, world!";
```

Выберите 1 правильный ответ.

- System.out.println(object.super.msg.text);
- System.out.println(text);
- System.out.println(object.msg.text);
- System.out.println(msg.text);
- System.out.println(super.msg.text);
- System.out.println(Message.text);

Задание 9

Какой из конструкторов может быть добавлен в класс MySub, чтобы это не вызвало ошибку при компиляции?

```
class MySuper {
   int number;
   MySuper(int i) { number = i; }
}

class MySub extends MySuper {
   int count;
   MySub(int cnt, int num) {
      super(num);
      count=cnt;
   }

   // Конструктор
```

Выберите 1 правильный ответ.

- MySub() {}
- MySub(int cnt) { super(); count = cnt; }
- MySub(int cnt) { count = cnt; }
- MySub(int cnt) { this(cnt, cnt); }
- MySub(int cnt) { super(cnt); this(cnt, 0); }
- MySub(int cnt) { count = cnt; super(cnt); }

Задание 10

Какое утверждение верно? Выберите правильный ответ.

- Вызов super(), как первый оператор в теле конструктора подкласса, всегда будет выполняться, поскольку все суперклассы имеют конструктор по умолчанию
- Вызов super() или this() должен быть первым оператором в теле конструктора.
- Если ни super(), ни this() не объявлены первыми операторами в теле конструктора, то будет неявно добавлен первым оператором this()
- Если super() является первой строкой в теле конструктора, то вызов this() может быть вторым
- Если и подкласс, и его суперкласс не содержат никаких конструкторов, то во время выполнения неявный конструктор по умолчанию подкласса вызовет super()

Задание 11
Какой будет результат работы программы? Выберите 1 правильный ответ.

```
public class MyClass {
   public static void main(String[] args) {
      B b = new B("Test");
   }
}

class A {
   A() { this("1", "2"); }

   A(String s, String t) { this(s + t); }

   A(String s) { System.out.println(s); }
}

class B extends A {
   B(String s) { System.out.println(s); }

   B(String s, String t) { this(t + s + "3"); }

   B() { super("4"); };
```

- Будет выведено Test, потом Test.
- Будет выведено Test.
- Будет выведено 123, потом Test.
- Будет выведено 4, потом Test.
- Будет выведено 12, потом Test.

Задание 12
Какие утверждения об интерфейсах верны? Выберите 2 правильных ответа.

- Интерфейсы позволяют реализовывать множественное наследование реализации
- Интерфейс может расширить любое количество других интерфейсов
- Члены интерфейса никогда не могут быть static
- Члены интерфейса всегда могут быть объявлены как static

Задание 13
Какие из этих объявлений полей допустимы в теле интерфейса? Выберите 3 правильных ответа.

- public static int answer = 42;
- final static int answer = 42;
- public int answer = 42;
- private final static int answer = 42;

Задание 14
Какое объявление может быть добавлено в указанную строку следующего кода, чтобы это не вызвало ошибок компиляции? Выберите 2 правильных ответа.

```
interface MyConstants {
   int r = 42;
   int s = 69;
   // Код
 final double circumference = 2*Math.PI*r;

 int AREA = r*s;

 public static MAIN = 15;

 protected int CODE = 31337;
```

Задание 15
Какое утверждение относительно следующей программы верно?

```
// Filename: MyClass.java
public class MyClass {
   public static void main(String[] args) {
      A[] arrA;
      B[] arrB;

      arrA = new A[10];
      arrB = new B[20];
      arrA = arrB;       // (1)
      arrB = (B[]) arrA; // (2)
      arrA = new A[10];
      arrB = (B[]) arrA; // (3)
```

```
    }
}

class A {}
class B extends A {}
```

Выберите один правильный ответ.

- Программа успешно откомпилируется и выполнится без ошибок, даже если приведение (B[ ]) в операторах строк (2) и (3) удалить.
- Во время выполнения будет выброшено исключение java.lang.ClassCastException в строке присваивания (2).
- Во время выполнения будет выброшено исключение java.lang.ClassCastException в строке присваивания (3).
- Программа успешно откомпилируется и выполнится без ошибок, даже если приведение (B[ ]) в операторах строк (2) и (3) удалить.
- На этапе компиляции произойдет ошибка из-за присваивания в строке (1).

Задание 16

Какая из строк первой приведет к ошибке на этапе компиляции следующей программы? Выберите правильный ответ.

```
// Filename: MyClass.java
class MyClass {
   public static void main(String[] args) {
      MyClass a;
      MySubclass b;

      a = new MyClass();          // (1)
      b = new MySubclass();       // (2)

      a = b;                      // (3)
      b = a;                      // (4)

      a = new MySubclass();       // (5)
      b = new MyClass();          // (6)
   }
}

class MySubclass extends MyClass {}
```

- Строка (1).
- Строка (2).
- Строка (3).
- Строка (4).
- Строка (5).
- Строка (6).

Задание 17
Даны описания и объявления ссылок. Какая операция присваивания из приведенных в вариантах ответа допустима?

```
// Описания:
interface I1 {}
interface I2 {}
class C1 implements I1 {}
class C2 implements I2 {}
class C3 extends C1 implements I2 {}

// Объявления ссылок:
// ...
C1 obj1;
   C2 obj2;
   C3 obj3;
```

Выберите один правильный ответ.

- I1 b = obj3;
- I1 a = obj2;
- obj3 = obj2;
- obj3 = obj1;
- obj2 = obj1;
- I2 c = obj1;

Задание 18
Даны описания и объявления ссылок. Что можно сказать об операции y=(Sub) x?

```
// Описания классов:
class Super {}
class Sub extends Super {}
// Объявления ссылок:
// ...
Super x;
Sub y;
// ...
```

Выберите один правильный ответ.

- Допустимо для компиляции, но может быть неверно при выполнении.
- Абсолютно правомерно для выполнения, а приведение (Sub) строго обязательно.
- Недопустимо для компиляции.
- Абсолютно правомерно для выполнения, но приведение (Sub) не является строго обязательным.

Задание 19
Даны описания и операторы объявлений. Какая одна из операций присваивания разрешена во время компиляции?

```
// Описания:
interface A {}
class B {}
class C extends B implements A {}
class D implements A {}

// Операторы объявлений:
// [...]
   B b = new B();
   C c = new C();
   D d = new D();
// [...]
```

Выберите один правильный ответ.

- d = (D) c;
- c = d;
- c = b;
- A a = d;
- d = c;

Задание 20
Какие буквы будут напечатаны, когда выполнится следующая программа?

```
// Filename: MyClass.java
public class MyClass {
   public static void main(String[] args) {
      B b = new C();
      A a = b;
      if (a instanceof A) System.out.println("A");
      if (a instanceof B) System.out.println("B");
      if (a instanceof C) System.out.println("C");
      if (a instanceof D) System.out.println("D");
   }
}

class A {}
class B extends A {}
class C extends B {}
class D extends C {}
```

Выберите три правильных ответа.

- A
- C

- D
- B

Задание 21
Даны три класса - A, B и C, где B является подклассом A и C является подклассом B. Какое из следующих логических выражений истинно для случая, когда объект, обозначенный ссылкой o, реально порождается от класса B, а не от A или C? Выберите один правильный ответ.

- (o instanceof B) && (!(o instanceof C))
- (o instanceof B) && !((o instanceof A) || (o instanceof C))
- (o instanceof B) && (!(o instanceof A))
- !((o instanceof A) || (o instanceof B))

Задание 22
Когда программа выполнится, она выведет все следующие буквы I, J, C и D. Это утверждение истинно или ложно?

```
public class MyClass {
   public static void main(String[] args) {
      I x = new D();
      if (x instanceof I) System.out.println("I");
      if (x instanceof J) System.out.println("J");
      if (x instanceof C) System.out.println("C");
      if (x instanceof D) System.out.println("D");
   }
}

interface I{}
interface J{}
class C implements I {}
class D extends C implements J {}
```

- Ложно.
- Истинно.

Задание 23
Что получится, если попробовать откомпилировать и выполнить следующую программу?

```
public class Polymorphism {
   public static void main(String[] args) {
      A ref1 = new C();
      B ref2 = (B) ref1;
      System.out.println(ref2.f());
   }
}
```

```
class A          { int f() { return 0; } }
class B extends A { int f() { return 1; } }
class C extends B { int f() { return 2; } }
```
Выберите один правильный ответ.

- Программа откомпилируется без ошибок, но во время выполнения будет выброшено исключение ClassCastException.
- Произойдет ошибка на этапе компиляции.
- Программа откомпилируется без ошибок, и на этапе выполнения будет выведено 0.
- Программа откомпилируется без ошибок, и на этапе выполнения будет выведено 1.
- Программа откомпилируется без ошибок, и на этапе выполнения будет выведено 2.

Задание 24

Что получится, если попробовать откомпилировать и выполнить следующую программу?

```
public class Polymorphism {
   public static void main(String[] args) {
      A ref1 = new C();
      B ref2 = (B) ref1;
      System.out.println(ref2.f());
   }
}

class A          { int f() { return 0; } }
class B extends A { int f() { return 1; } }
class C extends B { int f() { return 2; } }
```
Выберите один правильный ответ.

- Программа откомпилируется без ошибок, и на этапе выполнения будет выведено 1.
- Программа откомпилируется без ошибок, и на этапе выполнения будет выведено 2.
- Программа откомпилируется без ошибок, и на этапе выполнения будет выведено 0.
- Произойдет ошибка на этапе компиляции.
- Программа откомпилируется без ошибок, и на этапе выполнения будет выведено 3.

Задание 25

Какое утверждение относительно следующего кода истинно? Выберите два правильных ответа.

```
public interface HeavenlyBody { String describe(); }
```

```
class Star {
   String starName;
   public String describe() { return "star " + starName; }
}

class Planet extends Star {
   String name;
   public String describe() {
      return "planet " + name + " orbiting star " + starName;
   }
}
```

- Произойдет ошибка на этапе компиляции, если имя starName заменить именем name в объявлении класса Star.
- Использование наследования обоснованно, поскольку Planet (Планета) является Star (Звездой).
- Экземпляр Planet (Планета) является допустимым экземпляром HeavenlyBody (Небесное тело).
- Произойдет ошибка на этапе компиляции.
- Произойдет ошибка на этапе компиляции, если имя starName заменить на имя bodyName в объявлении класса Star.

Литература:
Основная – 1; 2.
Дополнительная – 1, 2, 3.
Интернет-ресурс: электронный курс для студентов направления «Бизнес-информатика» http://it.rfei.ru/~q
Образовательные технологии, методы и формы обучения: дистанционные образовательные технологии, объяснительно-иллюстративного обучения с элементами проблемного изложения; развивающего обучения, проблемная лекция, практическое занятие, контекстное обучение, ИТ-методы.
Формируемые компетенции: ПК-15; ПК-16; ПК-17, ПК-18, ПК-19.
Образовательные результаты: З-1; З-2; З-3; З-4; У-1; У-2; У-3; У-4; В-1, В-2, В-3.
Формы контроля, оценочные средства: текущий контроль: опрос, самостоятельная работа, выполнение заданий с использованием ЭВМ.

Научно-исследовательская работа студентов

Тема: «Поток команд управления, управление исключениями и диагностические утверждения»

Тема: «Объектно-ориентированное программирование»

Тема: «Потоки выполнения»

Цель: формирование у студентов представления об особенностях работы различных операторов Java, управлении исключениями, приобретение практических навыков в работе с различными приложениями и средами разработки.

Рассмотреть работу операторов, упомянутых в учебном материале, самостоятельно установив дополнительное программное обеспечение:

- *Java Platform Standard Edition Development Kit (JDK) 6.0 и выше.*
- *Среда разработки IntelliJ IDEA 10 и выше.*

Java Development Kit (сокращенно JDK) — бесплатно распространяемый компанией Oracle Corporation (ранее Sun Microsystems) комплект разработчика приложений на языке Java, включающий в себя компилятор Java (javac), стандартные библиотеки классов Java, примеры, документацию, различные утилиты и исполнительную систему Java (JRE). В состав JDK не входит интегрированная среда разработки на Java, поэтому разработчик, использующий только JDK, вынужден использовать внешний текстовый редактор (среду разработки).

IntelliJ IDEA - коммерческая интегрированная среда разработки программного обеспечения на многих языках программирования, в частности Java, JavaScript, Python и др., разработанная компанией JetBrains.

Литература:
Основная – 1; 2.
Дополнительная – 1, 2, 3.
Интернет-ресурс: http://it.rfei.ru/~q, http://www.oracle.com/technetwork/java/javase/downloads/index.html, http://www.jetbrains.com/idea/
Образовательные технологии, методы и формы обучения: дистанционные образовательные технологии, объяснительно-иллюстративного обучения с элементами проблемного изложения; развивающего обучения, проблемная лекция, практическое занятие, контекстное обучение, ИТ-методы.
Формируемые компетенции: ПК-15; ПК-16; ПК-17, ПК-18, ПК-19.
Образовательные результаты: З-1; З-2; З-3; З-4; У-1; У-2; У-3; У-4; В-1, В-2.
Формы контроля, оценочные средства: текущий контроль: опрос, самостоятельная работа, выполнение заданий с использованием ЭВМ.

Примерный перечень вопросов к экзамену

1. Классы

2. Члены классов: поля и методы
3. Объекты
4. Объектные ссылки
5. Вызовы методов
6. Терминология для членов класса
7. Наследование
8. Агрегация
9. Java-программы
10. Основные элементы языка
11. Константы
12. Символьные константы
13. Пробелы и комментарии
14. Примитивные типы данных
15. Объявления переменных
16. Первоначальные значения для переменных
17. Структура файла исходного кода Java
18. Приоритет и правила ассоциативности для операторов
19. Порядок вычисления операндов
20. Числовые расширения
21. Простой оператор присваивания
22. Арифметические операторы
23. Унарные арифметические операторы
24. Бинарные операторы
25. Операторы отношения
26. Ссылки на объекты
27. Булевы логические операторы
28. Целочисленные поразрядные операторы
29. Операторы сдвига
30. Тринарный условный оператор
31. Передача параметров
32. Передача ссылок на массив
33. Массивы
34. Инициализация и использование массивов
35. Многомерные массивы
36. Классы и методы
37. Конструкторы
38. Правила видимости
39. Пакеты
40. Модификаторы доступа для классов и интерфейсов верхнего уровня
41. Другие модификаторы для классов
42. Модификаторы доступа членов класса
43. Другие модификаторы для членов
44. Объявления и управление доступом
45. Операторы выбора
46. Операторы цикла
47. Команды перехода

48. Исключения в Java
49. Типы исключений
50. Обработка исключений
51. Оператор throw
52. Assertions
53. Применение диагностических утверждений
54. Управление исключениями и диагностические утверждения
55. Одиночное наследование реализации
56. Концепции ООП
57. Переопределение и сокрытие методов
58. Ключевые слова this() и super()
59. Интерфейсы
60. Иерархия типов
61. Преобразование типов значений ссылок
62. Полиморфизм и динамический поиск метода
63. Наследование в сравнении с агрегацией
64. Обзор вложенных классов и интерфейсов
65. Классы и интерфейсы как статические члены
66. Классы как нестатические члены
67. Неявная ссылка на объемлющий контекст
68. Локальные классы
69. Безымянные классы
70. Сборка мусора
71. Финализация объекта
72. Инициализаторы
73. Блоки статических инициализаторов
74. Конструирование первоначального состояния объекта
75. Многозадачность
76. Обзор потоков
77. Синхронизация
78. Переходы потоков между состояниями
79. Приоритеты потоков
80. Ожидание и уведомление
81. Блокирование из-за ввода/вывода
82. Класс Object
83. Классы-оболочки
84. Класс Math
85. Класс String
86. Чтение символов из строки
87. Класс StringBuffer
88. Инструментальный набор коллекций
89. Множества
90. Списки
91. Карты
92. Отсортированные множества и отсортированные карты
93. Реализация метода equals()

94. Реализация методов hashCode() и compareTo()
95. Декораторы коллекций
96. Другие вспомогательные методы класса Collections

Вопросы для самоконтроля по самостоятельно изученным темам

1. Классы
2. Члены классов: поля и методы
3. Объекты
4. Объектные ссылки
5. Вызовы методов
6. Терминология для членов класса
7. Наследование
8. Агрегация
9. Java-программы
10. Основные элементы языка
11. Константы
12. Символьные константы
13. Пробелы и комментарии
14. Примитивные типы данных
15. Объявления переменных
16. Первоначальные значения для переменных
17. Структура файла исходного кода Java
18. Приоритет и правила ассоциативности для операторов
19. Порядок вычисления операндов
20. Числовые расширения
21. Простой оператор присваивания
22. Арифметические операторы
23. Унарные арифметические операторы
24. Бинарные операторы
25. Операторы отношения
26. Ссылки на объекты
27. Булевы логические операторы
28. Целочисленные поразрядные операторы
29. Операторы сдвига
30. Тринарный условный оператор
31. Передача параметров
32. Передача ссылок на массив
33. Массивы
34. Инициализация и использование массивов
35. Многомерные массивы
36. Классы и методы
37. Конструкторы
38. Правила видимости
39. Пакеты
40. Модификаторы доступа для классов и интерфейсов верхнего уровня
41. Другие модификаторы для классов

6. Фонд оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине (модулю).

См. Приложение №2 к рабочей программе.

7. Перечень основной и дополнительной литературы, необходимой для освоения дисциплины (модулю).

Литература

Основная

1. Философия Java. 4-е идзание, Б. Эккель, Питер, 2015. - 680 с.
2. Объектно-ориентированное программирование. Автор: Павел Хорев. Издательство: Academia. Серия: Высшее профессиональное образование. Бакалавриат; 2012 г.
3. Алгоритмы на Java. Авторы: Роберт Седжвик, Кевин Уэйн. Издательство: Вильямс; 2015 г.

Дополнительная

1. Управление качеством программного обеспечения: учебник / Б.В. Черников. - М.: ИД "ФОРУМ": ИНФРА-М, 2012. ЭБС:Знаниум Конфликтология.
2. Java. Эффективное программирование. Автор: Джошуа Блох. Издательство: Лори. Серия: Java "из первых рук"; 2014 г.
3. Структуры данных и алгоритмы в Java. Автор: Роберт Лафоре. Переводчик: Е. Матвеев. Издательство: Питер. Серия: Классика Computer Science; 2015 г.
4. Programmer's Guide to Java™ Certification, A: A Comprehensive Primer, Second Edition By Khalid A. Mughal, Rolf W. Rasmussen, Addison Wesley, 2003
5. Философия Java. 4-е идзание, Б. Эккель, Питер, 2008
6. Java 2, П. Ноутон, Г. Шилдт, BHV - Санкт - Петербург, 2007

8. Перечень ресурсов информационно-телекоммуникационной сети «Интернет» (далее – сеть «Интернет»), необходимых для освоения дисциплины (модулю).

1. Электронная библиотека Регионального финансово-экономического института – http://students.rfei.ru/a/students/library.jspx
2. http://www.oracle.com/technetwork/java/javase/downloads/index.html
3. http://www.jetbrains.com/idea/

9. Методические указания для обучающихся по освоению дисциплины (модулю).

Методические указания по изучению дисциплины представляют собой комплекс рекомендаций и объяснений, позволяющих бакалавру оптимальным образом организовать процесс изучения данной дисциплины. Известно, что в структуре учебного плана значительное время отводится на самостоятельное изучение дисциплины. В рабочих программах дисциплин размещается примерное распределение часов аудиторной и внеаудиторной нагрузки по различным темам данной дисциплины.

Для успешного освоения дисциплины бакалавр должен:

1. Прослушать курс лекций по дисциплине.
2. Выполнить все задания, рассматриваемые на практических занятиях, включая решение задач.
3. Выполнить все домашние задания, получаемые от преподавателя.
4. Решить все примерные практические задания, рассчитанные на подготовку к промежуточной аттестации.

При подготовке к промежуточной аттестации особое внимание следует обратить на следующие моменты:

1. Выучить определения всех основных понятий.
2. Повторить все задания, рассматриваемые в течение семестра.
3. Проверить свои знания с помощью тестовых заданий.

Рекомендации по работе на лекционном занятии

На лекциях преподаватель излагает и разъясняет основные, наиболее сложные понятия темы, а также связанные с ней теоретические и практические проблемы, дает рекомендации на семинарское занятие и указания на самостоятельную работу. В ходе лекции бакалавр должен внимательно слушать и конспектировать лекционный материал.

Рекомендации для самостоятельной работы

Самостоятельная работа бакалавров – планируемая учебная, научно-

исследовательская работа, выполняемая во внеаудиторное время по заданию и при методическом руководстве преподавателя, но без его непосредственного участия.

Цель самостоятельной работы бакалавра – научиться осмысленно и самостоятельно работать сначала с учебным материалом, затем с научной информацией, изучить основы самоорганизации и самовоспитания с тем, чтобы в дальнейшем непрерывно повышать свою квалификацию.

Целью самостоятельной работы бакалавров по дисциплине является овладение фундаментальными знаниями, профессиональными умениями и навыками решения задач и теоретическим материалом по дисциплине. Самостоятельная работа способствует развитию самостоятельности, ответственности и организованности, творческого подхода к решению различных проблем.

В зависимости от конкретных видов самостоятельной работы, используемых в каждой конкретной рабочей программе, следует придерживаться следующих рекомендаций.

Одной из форм текущего контроля знаний студентов является контрольная работа. Контрольная работа подразумевает знакомство с основной и дополнительной литературой, включая справочные издания, зарубежные источники, конспект основных положений, требующихся для запоминания и являющихся основополагающими в этой теме.

Выполняя контрольную работу, необходимо внимательно ознакомиться с условиями заданий и написать развернутый и аргументированный ссылками на нормативные акты и литературу ответ. При написании контрольной работы необходимо проанализировать научную и учебную специальную литературу, действующие нормативно-правовые акты, публикации в периодической печати, судебную практику, статистические данные. В процессе выполнения работы необходимо подтверждать свои выводы цифровыми примерами, представленными в виде таблиц, диаграмм, графиков, а также примерами судебной практики. Как правило, контрольные работы проводятся на семинарском занятии.

Подготовка к написанию реферата предполагает поиск литературы и составление списка используемых источников, изложение мнения авторов и своего суждения по выбранному вопросу; формулирование основных аспектов проблемы.

Коллоквиум представляет собой одну из форм учебных занятий, ориентированную на определение качества работы с конспектом лекций, подготовки ответов к контрольным вопросам и др. Коллоквиумы, как правило, проводятся в форме мини-экзамена, имеющего целью уменьшить список тем, выносимых на основной экзамен, и оценить текущий уровень знаний бакалавров.

При подготовке к практикуму/лабораторной работе бакалаврам предлагается выполнить задания, подготовить проекты, составленные преподавателем по каждой учебной дисциплине.

Следует также учитывать краткие комментарии при написании курсовой работы, если она предусмотрена рабочей программой, и подготовке к

итоговому контролю, проводимого в форме зачета и (или) экзамена. Так, написание курсовой работы базируется на изучении научной, учебной, нормативной и другой литературы. Включает отбор необходимого материала, формирование выводов и разработку конкретных рекомендаций по решению поставленных цели и задач, проведение практических исследований по данной теме. Все необходимые требования к оформлению находится в методических указаниях по написанию курсовой работы.

Рекомендации по подготовке к практическому (семинарскому) занятию

Семинарское занятие представляет собой такую форму обучения в учреждениях высшего образования, которая предоставляет студентам возможности для обсуждения теоретических знаний с целью определения их практического применения, в том числе средствами моделирования профессиональной деятельности. Семинарские занятия служат для закрепления изученного материала, развития умений и навыков подготовки докладов, сообщений, приобретения опыта устных публичных выступлений, ведения дискуссии, аргументации и защиты выдвигаемых положений, а также для контроля преподавателем степени подготовленности бакалавров по изучаемой дисциплине. При наличии практических заданий по изучаемой дисциплине бакалавр выполняет все упражнения и задачи, подготовленные преподавателем. Целью практического занятия является более углубленное изучение отдельных тем дисциплины и применение полученных теоретических навыков на практике.

Семинарское занятие не сводится к закреплению или копированию знаний, полученных на лекции. Его задачи значительно шире, сложнее и интереснее. Семинарское занятие одновременно реализует учебное, коммуникативное и профессиональное предназначение. Подготовка к практическому (семинарскому) занятию начинается с тщательного ознакомления с условиями предстоящей работы, т. е. с обращения к планам семинарских занятий.

Подготовка к практическим занятиям должна носить систематический характер. Это позволит бакалавру в полном объеме выполнить все требования преподавателя.

Тщательная подготовка к семинарским занятиям, как и к лекциям, имеет определяющее значение: семинар пройдёт так, как аудитория подготовилась к его проведению.

Самостоятельная работа – столп, на котором держится вся подготовка по изучаемому курсу. Готовясь к практическим занятиям, следует активно пользоваться справочной литературой: энциклопедиями, словарями, альбомами схем и др. Владение понятийным аппаратом изучаемого курса является необходимостью.

При подготовке к семинару бакалавры имеют возможность воспользоваться консультациями преподавателя. Кроме указанных тем бакалавры вправе, по согласованию с преподавателем, избирать и другие интересующие их темы.

Определившись с проблемой, привлекающей наибольшее внимание, следует обратиться к рекомендуемой литературе. Следует иметь ввиду, что в семинаре участвует вся группа, а потому задание к практическому занятию следует распределить на весь коллектив. Задание должно быть охвачено полностью и рекомендованная литература должна быть освоена группой в полном объёме.

Для полноценной подготовки к практическому занятию чтения учебника крайне недостаточно – в учебных пособиях излагаются только принципиальные основы, в то время как в монографиях и статьях на ту или иную тему поднимаемый вопрос рассматривается с разных ракурсов или ракурса одного, но в любом случае достаточно подробно и глубоко. Тем не менее, для того, чтобы должным образом сориентироваться в сути задания, сначала следует ознакомиться с соответствующим текстом учебника – вне зависимости от того, предусмотрена ли лекция в дополнение к данному семинару или нет. Оценив задание, выбрав тот или иной сюжет, и подобрав соответствующую литературу, можно приступать собственно к подготовке к семинару. Для получения более глубоких знаний бакалаврам рекомендуется изучать дополнительную литературу. Следует активно пользоваться справочной литературой: энциклопедиями, словарями, альбомами схем и др. Владение понятийным аппаратом изучаемого курса является необходимостью. В ходе работы студент должен применить приобретенные знания при обобщении теоретического и практического материала, продемонстрировать навыки грамотного изложения своих мыслей с использованием общеправовой и отраслевой терминологии.

Семинар (практическое занятие) предполагает свободный обмен мнениями по избранной тематике. Преподаватель формулирует цель занятия и характеризует его основную проблематику. Заслушиваются сообщения бакалавров. Обсуждение сообщения совмещается с рассмотрением намеченных вопросов. Кроме того заслушиваются сообщения, предполагающие анализ публикаций по отдельным вопросам семинара. Поощряется выдвижение и обсуждение альтернативных мнений. Преподаватель подводит итоги обсуждения и объявляет оценки выступавшим бакалаврами. В целях контроля подготовленности бакалавров и привития им навыков краткого письменного изложения своих мыслей преподаватель в ходе семинарских занятий может осуществлять текущий контроль знаний в виде тестовых заданий.

На семинаре идёт не проверка вашей подготовки к занятию (подготовка есть необходимое условие), но степень проникновения в суть материала, обсуждаемой проблемы. Поэтому беседа будет идти не по содержанию прочитанных работ; преподаватель будет ставить проблемные вопросы, не все из которых могут прямо относиться к обработанной вами литературе.

В ходе практических занятий бакалавры под руководством преподавателя могут рассмотреть различные методы решения задач по дисциплине. Продолжительность подготовки к практическому занятию должна составлять не менее того объема, что определено тематическим планированием в рабочей программе. Практические занятия по дисциплине

могут проводиться в различных формах:

1) устные ответы на вопросы преподавателя по теме занятия; 2) письменные ответы на вопросы преподавателя; 3) групповое обсуждение той или иной проблемы под руководством и контролем преподавателя; 4) заслушивания и обсуждение контрольной работы; 5) решение задач.

При работе необходимо не только привлечь наиболее широкий круг литературы, но и суметь на ее основе разобраться в степени изученности темы. Стоит выявить дискуссионные вопросы, нерешенные проблемы, попытаться высказать свое отношение к ним, привести и аргументировать свою точку зрения или отметить, какой из имеющихся в литературе точек зрения по данной проблематике придерживается автор и почему.

Рекомендации по работе с литературой

Изучение литературы очень трудоемкая и ответственная часть подготовки к семинарскому занятию, написанию эссе, реферата, доклада и т.п. Работа над литературой, статья ли это или монография, состоит из трёх этапов – чтения работы, её конспектирования, заключительного обобщения сути изучаемой работы.

Работа с литературой, как правило, сопровождается записями в следующих формах:

- конспект – краткая схематическая запись основного содержания научной работы. Целью конспектирования является выявление логики, схемы доказательств, основных выводов произведения;
- план – краткая форма записи прочитанного, перечень вопросов, рассматриваемых в книге, статье, составление плана раскрывает логику произведения, способствует ориентации в его содержании;
- выписки – либо цитаты из произведения, либо дословное изложение мест из источника, способствуют более глубокому пониманию читаемого текста;
- тезисы – сжатое изложение основных мыслей и положений прочитанного материала;
- аннотация – очень краткое изложение содержания прочитанной работы, составляется после полного прочтения и осмысливания работы;
- резюме – краткая оценка прочитанного произведения, отражает наиболее общие выводы и положения работы, ее концептуальные итоги.

Прежде, чем браться за конспектирование, скажем, статьи, следует её хотя бы однажды прочитать, чтобы составить о ней предварительное мнение, постараться выделить основную мысль или несколько базовых точек, опираясь на которые можно будет в дальнейшем работать с текстом.

Конспектирование – дело очень тонкое и трудоёмкое, в общем виде может быть определено как фиксация основных положений и отличительных черт рассматриваемого труда вкупе с творческой переработкой идей, в нём содержащихся. Конспектирование – один из эффективных способов усвоения

письменного текста. Хотя само конспектирование уже может рассматриваться как обобщение, тем не менее есть смысл выделить последнее особицей, поскольку в ходе заключительного обобщения идеи изучаемой работы окончательно утверждаются в сознании изучающего. Достоинством заключительного обобщения как самостоятельного этапа работы с текстом является то, что здесь читатель, будучи автором обобщений, отделяет себя от статьи, что является гарантией независимости читателя от текста.

Если программа занятия предусматривает работу с источником, то этой стороне подготовки к семинару следует уделить пристальное внимание. В сущности, разбор источника не отличается от работы с литературой – то же чтение, конспектирование, обобщение.

Рекомендации к написанию реферата

Использование реферата в качестве промежуточного или итогового отчета студента о самостоятельном изучении какой-либо темы учебного курса предполагает, прежде всего, установление целей и задач данной работы, а также его функциональной нагрузки в процессе обучения.

Реферат – это композиционно-организованное, обобщенное изложение содержания источника информации (в учебной ситуации – статей, монографий, материалов конференции, официальных документов и др., но не учебника по данной дисциплине). Тема реферата может быть предложена преподавателем или выбрана студентом из рабочей программы соответствующей дисциплины.

Возможно, после консультации с преподавателем, обоснование и формулирование собственной темы.

Тема реферата должна отражать проблему, которая достаточно хорошо исследована в науке. Как правило, внутри такой проблемы выбирается для анализа какой-либо единичный аспект.

Тематика может носить различный характер:

- межпредметный,
- внутрипредметный,
- интегративный,
- быть в рамках программы дисциплины или расширять ее содержание (рассмотрение истории проблемы, новых теорий, новых аспектов проблемы).

Целью реферата является изложение какого-либо вопроса на основе обобщения, анализа и синтеза одного или нескольких первоисточников. Другими словами, реферат отвечает на вопрос «какая информация содержится в первоисточнике, что излагается в нем?».

Принимая во внимание, что реферат – одна из форм интерпретации исходного текста одного или нескольких первоисточников, следует сформулировать задачу, стоящую перед студентами: создать новый текст на основе имеющихся текстов, т.е. текст о тексте. Новизна в данном случае

подразумевает собственную систематизацию материала при сопоставлении различных точек зрения авторов и изложении наиболее существенных положений и выводов реферируемых источников.

1. Требования к рефератам.

Прежде всего, следует помнить, что реферат не должен отражать субъективных взглядов референта (студента) на излагаемый вопрос, а также давать оценку тексту.

Основными требованиями к реферату считаются:

1. информативность и полнота изложения основных идей первоисточника;
2. точность изложения взглядов автора – неискаженное фиксирование всех положений первичного текста,
3. объективность – реферат должен раскрывать концепции первоисточников с точки зрения их авторов;
4. изложение всего существенного – «чтобы уметь схватить новое и существенное в сочинениях» (М.В. Ломоносов);
5. изложение в логической последовательности в соответствии с обозначенной темой и составленным планом;
6. соблюдение единого стиля – использование литературного языка в его научно-стилевой разновидности;
7. корректность в характеристике авторского изложения материала.

2. Виды рефератов.

По характеру воспроизведения информации различают рефераты репродуктивные и продуктивные.

Репродуктивные рефераты воспроизводят содержание первичного текста:

- реферат-конспект содержит в обобщенном виде фактографическую информацию, иллюстративный материал, сведения о методах исследования, о полученных результатах и возможностях их применения;
- реферат-резюме приводит только основные положения, тесно связанные с темой текста.

Продуктивные рефераты предполагают критическое или творческое осмысление литературы:

- реферат-обзор охватывает несколько первичных текстов, дает сопоставление разных точек зрения по конкретному вопросу;
- реферат-доклад дает анализ информации, приведенной в первоисточниках, и объективную оценку состояния проблемы.

По количеству реферируемых источников:

- монографические – один первоисточник;
- обзорные – несколько первичных текстов одной тематики.

По читательскому назначению:

– общие – характеристика содержания в целом; ориентация на широкую аудиторию;
– специализированные – ориентация на специалистов.

3. Этапы работы над рефератом.

1. Выбор темы.
2. Изучение основных источников по теме.
3. Составление библиографии.
4. Конспектирование необходимого материала или составление тезисов.
5. Систематизация зафиксированной и отобранной информации.
6. Определение основных понятий темы и анализируемых проблем.
7. Разработка логики исследования проблемы, составление плана.
8. Реализация плана, написание реферата.
9. Самоанализ, предполагающий оценку новизны, степени раскрытия сущности проблемы, обоснованности выбора источников и оценку объема реферата.
10. Проверка оформления списка литературы.
11. Редакторская правка текста.
12. Оформление реферата и проверка текста с точки зрения грамотности и стилистики.

4. Структура реферата.

В структуре реферата выделяются три основных компонента: библиографическое описание, собственно реферативный текст, справочный аппарат.

Библиографическое описание предполагает характеристику имеющихся на эту тему работ, теорий; историографию вопроса; выделение конкретного вопроса (предмета исследования); обоснование использования избранных первоисточников.

Собственно реферативный текст:

Введение – обоснование актуальности темы, проблемы; предмет, цели и задачи реферируемой работы, предварительное формулирование выводов.

Основная часть – содержание, представляющее собой осмысление текста, аналитико-синтетическое преобразование информации, соответствующей теме реферата.

Основную часть рекомендуется разделить на два-три вопроса. В зависимости от сложности и многогранности темы, вопросы можно разделить на параграфы. Чрезмерное дробление вопросов или, наоборот, их отсутствие приводят к поверхностному изложению материала. Каждый вопрос должен заканчиваться промежуточным выводом и указывать на связь с последующим вопросом.

Заключение – обобщение выводов автора, область применения результатов работы.

Справочный аппарат:

Список литературы – список использованных автором реферата работ (может состоят из одного и более изданий).

Приложения (необязательная часть) – таблицы, схемы, графики,

фотографии и т.д.

Реферат как образец письменной научной речи

1. Качества научной речи.

Функциональные стили различаются:

- характером предаваемой информации;
- сферой функционирования;
- адресатом;
- использованием языковых средств различных уровней.

Главной коммуникативной задачей реферата является выражение научных понятий и умозаключений.

Реферат должен быть написан научным стилем, что предполагает:

- передачу информации научного характера;
- функционирование в образовательной среде;
- в качестве адресата преподавателя, т.е. специалиста, или студентов,
- заинтересованных в получении данной информации;
- демонстрацию характерных языковых особенностей письменной разновидности научно-учебного подстиля литературного языка.

Научный стиль обладает рядом экстралингвистических характеристик, или качеств:

- точность – строгое соответствие слов обозначаемым предметам и явлениям действительности (знание предмета и умение выбирать необходимую лексику);
- понятность – доступность речи для тех, кому она адресована (правильное использование терминов, иностранных слов, профессионализмов);
- логичность, последовательность – четкое следование в изложении логике и порядку связей в действительности (первоисточнике);
- объективность – отсутствие субъективных суждений и оценок в изложении информации;
- абстрактность и обобщенность – отвлеченность от частных, несущественных признаков;
- преобладание рассуждения как типа речи над описанием и повествованием;
- графическая информация наличие схем, графиков, таблиц, формул и т.п.

2. Особенности письменной научной речи

Письменная речь, в отличие от устной, подразумевает:

- определенную степень подготовленности к работе;
- возможность исправления и доработки текста;
- наличие композиции строения, соотношения и взаимного расположения частей реферата;

– выдержанность стиля изложения; строгое следование лексическим и грамматическим нормам.

Доминирующим фактором организации языковых средств в научном стиле является их обобщенно-отвлеченный характер на лексическом и грамматическом уровнях языковой системы.

Лексический уровень предполагает:

– использование абстрактной лексики, преобладающей над конкретной: мышление, отражение, изменяемость, преобразование, демократизация и т.п.;
– отсутствие единичных понятий и конкретных образов, что подчеркивается употреблением слов обычно, постоянно, регулярно, систематически, каждый и т.п.;
– преобладание терминов различных отраслей науки: лексикология, коммуникация, эмпиризм, гносеология, адаптация и т.п.;
– использование слов общенаучного употребления: функция, качество, значение, элемент, процесс, анализ, доказательство и т.п.;
– употребление многозначных слов в одном (реже двух) значениях: предполагать (считать, допускать); окончание (завершение), рассмотреть (разобрать, обдумать, обсудить) и т.п.;
– наличие специфических фразеологизмов: рациональное зерно, демографический взрыв, магнитная буря и т.п.;
– клиширование: представляет собой..., включает в себя..., относится к..., заключается в... и т.п.;
– преобладание отвлеченных существительных над однокоренными глаголами: взаимодействие, зависимость, классификация, систематизация и т.п.

Грамматический уровень:

– использование аналитической степени сравнения: более сложный, наиболее простой, менее известный и т.п. в отличие от эмоционально окрашенных: наиважнейший, сложнейший, ближайший и т.п.;
– преимущественное употребление глаголов 3 лица ед. и мн.ч. настоящего времени (реже 1 лица будущего времени сравним, рассмотрим): исследуются, просматривается, подразумевается, доказывает и т.п.;
– активность союзов, предлогов, предложных сочетаний: в связи..., в соответствии..., в качестве..., в отношении..., сравнительно с ... и т.п.;
– преобладание пассивных (страдательных) конструкций: рассмотрены вопросы,
– описаны явления, сделаны выводы, отражены проблемы и т.п.;

– выражение четкой связи между частями сложного предложения: следует сказать, что...; наблюдения показывают, что..., необходимо подчеркнуть, что... и т.п.;
– усиленная связующая функция наречий и наречных выражений: поэтому, итак, таким образом, наконец... и т.п.;
– осложнение предложений обособленными конструкциями: «Стремлением к смысловой точности и информативности обусловлено употребление в научной речи конструкций с несколькими вставками и пояснениями, уточняющими содержание высказывания, ограничивающими его объем, указывающими источник информации и т.д.».

Обобщая отличительные языковые особенности письменного научного стиля, можно сказать, что он характеризуется:

– употреблением книжной, нейтральной и терминологической лексики;
– преобладанием абстрактной лексики над конкретной;
– увеличением доли интернационализмов в терминологии;
– относительной однородностью, замкнутостью лексического состава;
– неупотребительностью разговорных и просторечных слов; слов с эмоционально-экспрессивной и оценочной окраской;
– наличием синтаксических конструкций, подчеркивающих логическую связь и последовательность мыслей.

Оформление реферата. Критерии оценки.

Правила оформления реферата регламентированы. Объем – не более 10-15 стр. машинописного текста, напечатанного в формате Word 7,0, 8,0; размер шрифта – 14; интервал – 1,5, формат бумаги А 4, сноски постраничные, сплошные; поле (верхнее, нижнее, левое, правое) 2 мм; выравнивание – по ширине; ориентация книжная; шрифт Times New Roman Cyr.

Работа должна иметь поля; каждый раздел оформляется с новой страницы.

Титульный лист оформляется в соответствии с установленной формой.

На первой странице печатается план реферата, включающий в себя библиографическое описание; введение, разделы и параграфы основной части, раскрывающие суть работы, заключение; список литературы; приложения.

В конце реферата представляется список использованной литературы с точным указанием авторов, названия, места и года ее издания.

Критерии оценки реферата.

1. Степень раскрытия темы предполагает:

– соответствие плана теме реферата;
– соответствие содержания теме и плану реферата;
– полноту и глубину раскрытия основных понятий;

– обоснованность способов и методов работы с материалом;
– умение работать с литературой, систематизировать и структурировать материал;
– умение обобщать, делать выводы, сопоставлять различные
– точки зрения по рассматриваемому вопросу.

2. Обоснованность выбора источников оценивается:
– полнотой использования работ по проблеме;
– привлечением наиболее известных и новейших работ по проблеме (журнальные публикации, материалы сборников научных трудов и т.д.).

3. Соблюдение требований к оформлению определяется:
– правильным оформлением ссылок на используемую литературу;
– оценкой грамотности и культуры изложения;
– владением терминологией и понятийным аппаратом проблемы;
– соблюдением требований к объему реферата;
– культурой оформления.

Защита реферата

Рефераты обычно представляются на заключительном этапе изучения дисциплины как результат итоговой самостоятельной работы студента. Защита реферата осуществляется или на аудиторных занятиях, предусмотренных учебным планом, или на зачете как один из вопросов билета (последнее определяется преподавателем).

Если реферат подразумевает публичную защиту, то выступающему следует заранее подготовиться к реферативному сообщению, а преподавателю и возможным оппонентам – ознакомиться с работой.

Реферативное сообщение отличается от самого реферата прежде всего объемом и стилем изложения, т.к. учитываются особенности устной научной речи и публичного выступления в целом. В реферативном сообщении содержание реферата представляется подробно (или кратко) и, как правило, вне оценки, т.е. изложение приобретает обзорный характер и решает коммуникативную задачу (передать в устной форме информацию, которая должна быть воспринята слушателями). Учитывая публичный характер высказываний, выступающий должен:
– составить план и тезисы выступления;
– кратко представить проблематику, цель, структуру и т.п.;
– обеспечить порционную подачу материала не в соответствии с частями, разделами и параграфами, а сегментировать в зависимости от новизны информации;
– соблюдать четкость и точность выражений, их произнесение; обращать внимание на интонацию, темп, громкость и т.п. особенности публичного выступления;

– демонстрировать подготовленный характер высказываний, допуская, как в любой другой устной речи, словесную импровизацию.

Рекомендации по написанию эссе

Эссе – средство, позволяющее оценить умение обучающегося письменно излагать суть поставленной проблемы, самостоятельно проводить анализ этой проблемы с использованием концепций и аналитического инструментария соответствующей дисциплины, делать выводы, обобщающие авторскую позицию по поставленной проблеме.

Цель эссе состоит в развитии таких навыков, как самостоятельное творческое мышление и письменное изложение собственных мыслей.

Структура эссе определяется предъявляемыми требованиями:

– мысли автора по проблеме излагаются в форме кратких тезисов.

– мысль должна быть подкреплена доказательствами – поэтому за тезисом следуют аргументы.

Аргументы – это факты, явления общественной жизни, события, жизненные ситуации и жизненный опыт, научные доказательства, ссылки на мнение ученых и др.

Эссе обычно имеет кольцевую структуру (количество тезисов и аргументов зависит от темы, избранного плана, логики развития мысли):

– вступление

– тезис, аргументы

– тезис, аргументы

– тезис, аргументы

– заключение.

При написании эссе надо учитывать следующее:

Вступление и заключение должны фокусировать внимание на проблеме (во вступлении она ставится, в заключении – резюмируется мнение автора).

Необходимо выделение абзацев, красных строк, установление логической связи абзацев: так достигается целостность работы.

Стиль изложения: эмоциональность, экспрессивность, художественность.

Правила написания эссе:

– из формальных правил можно назвать только одно – наличие заголовка;

– внутренняя структура может быть произвольной. Поскольку это малая форма письменной работы, то не требуется обязательное повторение выводов в конце, они могут быть включены в основной текст или в заголовок;

– аргументация может предшествовать формулировке проблемы. Формулировка проблемы может совпадать с окончательным выводом.

В качестве примера можете познакомиться с широко известными эссе И.А. Бунина («Недостатки современной поэзии»), Д.С. Мережковского («О

причинах упадка и новых течениях современной русской литературы»), К.Д. Бальмонта («Элементарные слова о символической поэзии»), В.Я. Брюсова («Ключи тайн»), Вяч. Иванова («Символизм как миропонимание»), А.А. Блока («О лирике»).

Учебно-методические указания к выполнению тестовых заданий.

Тестовый контроль отличается от других методов контроля (устные и письменные экзамены, зачеты, контрольные работы и т.п.) тем, что он представляет собой специально подготовленный контрольный набор заданий, позволяющий надежно и адекватно количественно оценить знания обучающихся посредством статистических методов.

Все вышеуказанные преимущества тестового контроля могут быть достигнуты лишь при использовании теории педагогических тестов, которая сложилась на стыке педагогики, психологии и математической статистики. Основными достоинствами применения тестового контроля являются:

- объективность результатов проверки, так как наличие заранее определенного эталона ответа (ответов) каждый раз приводит к одному и тому же результату;
- повышение эффективности контролирующей деятельности со стороны преподавателя за счет увеличения её частоты и регулярности;
- возможность автоматизации проверки знаний учащихся, в том числе с использованием компьютеров;
- возможность использования в системах дистанционного образования.

Тест – инструмент, состоящий из системы тестовых заданий с описанными системами обработки и оценки результата, стандартной процедуры проведения и процедуры для измерения качеств и свойств личности, изменение которых возможно в процессе систематического обучения.

Преимущество тестового контроля состоит в том, что он является научно обоснованным методом эмпирического исследования и в определенной сфере позволяет преодолеть умозрительные оценки знаний студентов. Следует отметить, что задания, используемые многими преподавателями и называемые ими тестовыми, на самом деле таковыми вовсе не являются. В отличие от обычных задач тестовые задания имеют четкий однозначный ответ и оцениваются стандартно на основе ценника. В самом простом случае оценка студента есть сумма баллов за правильно выполненные задания. Тестовые задания должны быть краткими, ясными и корректными, не допускающими двусмысленности. Сам же тест представляет собой систему заданий возрастающей трудности. Тестовый контроль может применяться как средство текущего, тематического и рубежного контроля, а в некоторых случаях и итогового.

Текущее тестирование осуществляется после изучения отдельной темы или группы тем. Текущее тестирование, прежде всего, является одним из элементов самоконтроля и закрепления слушателем пройденного учебного

материала.

Виды тестовых заданий

Тестовое задание (ТЗ) может быть представлено в одной из следующих стандартизированных форм:

- закрытое ТЗ, предполагающее выбор ответов (испытуемый выбирает правильный ответ (ответы) из числа готовых, предлагаемых в задании теста);
- открытое ТЗ (испытуемый сам формулирует краткий или развернутый ответ);
- ТЗ на установление правильной последовательности;
- ТЗ на установление соответствия между элементами двух множеств.

Закрытое тестовое задание

Закрытое ТЗ состоит из неполного тестового утверждения с одним ключевым элементом и множеством допустимых вариантов ответов, один или несколько из которых являются правильными. Тестируемый студент определяет правильные ответы из данного множества. Рекомендуется пять или шесть вариантов ответов, из которых два или три являются правильными.

Открытое тестовое задание

Открытое ТЗ имеет вид неполного утверждения, в котором отсутствует один или несколько ключевых элементов и требует самостоятельной формулировки ответа тестируемого. В качестве отсутствующих ключевых элементов могут быть: число, буква, слово или словосочетание. При формулировке задания на месте ключевого элемента необходимо поставить прочерк или многоточие.

Тестовое задание на установление правильной последовательности

ТЗ на установление правильной последовательности состоит из однородных элементов некоторой группы и четкой формулировки критерия упорядочения этих элементов.

Тестовое задание на установление соответствия

ТЗ на установление соответствия состоит из двух групп элементов и четкой формулировки критерия выбора соответствия между ними. Внутри каждой группы элементы должны быть однородными. Количество элементов во второй группе должно превышать количество элементов первой группы, но не более чем в 2 раза. Максимально допустимое количество элементов во второй группе не должно превышать 10. Количество же элементов в первой группе должно быть не менее двух.

Требования к тестовым заданиям

Для обеспечения адекватности оценки знаний тесты должны обладать следующими свойствами:

- тест должен быть **репрезентативным** с точки зрения изучаемого материала (ответы на вопросы, поставленные в тесте, не должны выходить за пределы данной учебной дисциплины);

- тест должен быть **уместным**: формулировка и состав вопросов должны соответствовать основной цели дисциплины (при тестировании по определенной теме вопросы должны соответствовать одной из основных задач дисциплины, упомянутых в программе курса);
- тест должен быть **объективным**, что заключается в неизбежности выбора правильного варианта ответа различными экспертами, а не только преподавателем, оставившим тест;
- тест должен быть **специфичным**, т.е. в тесте не должно быть таких вопросов, на которые мог бы ответить человек, не знающий данной дисциплины, но обладающий достаточной эрудицией;
- тест должен быть **оперативным**, что предусматривает возможность быстрого ответа на отдельный вопрос, поэтому вопросы формулируются коротко и просто и не должны включать редко используемые слова, конечно, если эти слова не являются понятиями, знание которых предусмотрено в учебной дисциплине.

Перечисленные свойства тестовых заданий обеспечивают необходимый качественный уровень проведения итогового контроля, к которому предъявляются следующие требования.

Процесс тестирования должен быть **валидным** (значимым), когда результаты подтверждают конкретные навыки и знания, которые экзамен подразумевает проверить.

Тестирование является **объективным**, если результаты не отражают мнения или снисходительность проверяющего.

Убедиться в **надежности** тестирования можно, если результаты повторно подтверждены последующими контрольными мероприятиями.

Эффективность тестирования определяется, если его выполнение и оценивание не занимает больше времени или денег, чем необходимо.

Тестирование можно считать **приемлемым**, если студенты и преподаватели воспринимают контрольное мероприятие адекватно его значимости.

Изучение динамики процесса проверки знаний с помощью тестов позволяет установить индивидуальное время тестирования для каждого конкретного набора тестовых заданий. Нередко время тестирования для различных дисциплин устанавливается одинаковым на основании некоторого стандарта, не принимая во внимание специфику конкретной дисциплины и ее раздела.

Указания по подготовке к зачету/экзамену

Формой итогового контроля знаний и умений, полученных в процессе изучения дисциплины является зачет и экзамен.

Экзамен (зачет) дает возможность преподавателю:

- выяснить уровень освоения студентами учебной программы дисциплины;

– оценить формирование у студентов определенных знаний и навыков их использования, необходимых и достаточных для будущей профессиональной деятельности;
– оценить умение студентов творчески мыслить и логически правильно излагать ответы на поставленные вопросы.

При подготовке к экзамену (зачету) необходимо ориентироваться на конспекты лекций, рекомендуемую литературу и др. Сдача экзамена и (или) зачета предполагает полное понимание, запоминание и применение изученного материала на практике. Для успешной подготовки к промежуточной аттестации студентам необходимо вновь обратиться к пройденному материалу. Литература для подготовки к экзамену (зачету) рекомендуется преподавателем, либо указана в рабочей программе по дисциплине.

При подготовке к промежуточной аттестации в качестве ориентира студент может использовать перечень контрольных вопросов для самопроверки. Подготовка ответов на эти вопросы позволит:
– выяснить уровень освоения студентами учебных программ;
– оценить формирование у студентов определенных знаний и навыков их использования, необходимых и достаточных для будущей профессиональной деятельности;
– оценить умение студентов творчески мыслить и логически правильно излагать ответы на поставленные вопросы.

Оценка знаний студентов должна опираться на строго объективные критерии, научно обоснованные педагогикой и обязательные для выполнения всех преподавателей.

Среди таких критериев важнейшими являются принципы подхода к оценке. В наиболее общем виде эти принципы можно представить следующим образом:
– глубокие знания и понимание существа вопроса, но не всех его деталей, а лишь основных;
– степень сознательного и творческого усвоения изучаемых наук как базы личных убеждений и полезных обществу действий;
– понимание сущности науки, места каждой темы в общем курсе и её связи с предыдущими и последующими темами;
– выделение коренных проблем науки и умение правильно использовать это знание в самостоятельной научной деятельности или практической работе по специальности.

Экзамен (зачет) может проводиться в устной, письменной форме и с применением тестов. Экзамен (зачет) проводится по вопросам, охватывающим весь пройденный материал. По окончанию экзамена (зачета) преподаватель может задать студенту дополнительные и уточняющие вопросы.

Студентам необходимо тщательно готовиться к итоговому экзамену. Процесс подготовки к итоговому экзамену начинается, по существу, с самого первого этапа изучения предмета. Он включает в себя самостоятельную работу над рекомендованной литературой. Как правило, он начинается за полтора-два месяца до экзаменационной сессии. Изучив и законспектировав рекомендованные источники, выполнив предусмотренные учебным планом письменные работы и имея рецензии на них, студент начинает непосредственную подготовку к экзамену с тщательной отработки курса в соответствии с требованиями учебной программы и выполнения рекомендаций преподавателя, данных в рецензии. На этом этапе студент должен повторить изученное по учебникам и учебным пособиям, личным конспектам, записям лекций и другим материалам. При этом особое внимание должно быть обращено на тщательную отработку тех конкретных вопросов и тем учебной программы, которые слабо усвоены.

При повторении материала перед итоговым экзаменом необходима самопроверка или взаимная проверка знаний. В этом случае по каждой теме надо еще раз хорошо продумать материал, найти соответствующие статьи из нормативных актов, подобрать примеры. Вполне себя оправдывает групповая взаимная проверка. Для этого рекомендуется собираться по 3-4 человека и проводить разбор вопросов по курсу. Экзамен проводится по билетам. Если какой-либо из поставленных в билете вопросов студенту кажется неясным, он может обратиться к преподавателю за разъяснением. Пользоваться наглядными пособиями, словарями или справочниками можно только с разрешения преподавателя. При подготовке к ответу, а также при ответе не обязательно придерживаться той последовательности вопросов, которая дана в билетах. Записи ответов лучше делать в виде развернутого плана, их можно дополнить цифрами, примерами, фактами, а также сослаться на необходимые нормативные акты и другие источники.

Ответ должен быть построен в форме свободного рассказа. Важно не только верно изложить соответствующее положение, но и дать его глубокое теоретическое обоснование. При ответах надо избегать больших выступлений, отклонений от существа вопросов, но не следует вдаваться и в такую крайность, как погоня за краткостью. Такой ответ не раскроет содержания вопроса и не даст возможности преподавателю правильно судить о знаниях студента. После ответов на вопросы билета преподаватель может задать дополнительные вопросы, на которые студент обязан ответить.

Экзаменатор оценивает знания по четырехбалльной системе: «отлично», «хорошо», «удовлетворительно», «неудовлетворительно». Все положительные оценки записываются в экзаменационную ведомость и зачетную книжку. Неудовлетворительные оценки проставляются в экзаменационную ведомость.

10. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине (модулю), включая перечень программного обеспечения и информационных

справочных систем.

При осуществлении образовательного процесса используется ряд информационных технологий обеспечения дистанционного обучения, включающий, но не исчерпывающийся, технологиями онлайн и оффлайн распространения образовательной информации (почтовая рассылка печатных материалов и бланков тестирования или электронных версий образовательных материалов на физических носителях, либо интерактивный доступ к материалам через интернет, доступ к электронно-библиотечным системам института и сторонних поставщиков), технологиями взаимодействия студентов с преподавателем (видео-лекции и семинары, групповые и индивидуальные консультации через интернет, индивидуальные консультации по телефону), технологиями образовательного контроля (интерактивные онлайн тесты в интернет, оффлайн тесты с использованием персональных печатных бланков).

Для реализации указанных технологий используется набор программного обеспечения и информационных систем, включающий, но не ограничивающийся, следующим списком.

4. операционные системы Microsoft Windows (различных версий);
5. операционная система GNU/Linux;
6. свободный фисный пакет LibreOffice;
7. система управления процессом обучения «Lete e-Learning Suite» (собственная разработка);
8. система электронного обучения студентов направления подготовки «Бизнес-информатика» EduTerra.pro
9. система интерактивного онлайн тестирования (собственная разработка);
10. система телефонной поддержки и консультаций сотрудниками колл-центра «Центральная служба поддержки» (собственная разработка);
11. система онлайн видео конференций Adobe Connect;
12. электронно-библиотечная система «Айбукс»;
13. электронно-библиотечная система «Издательства «Лань»;
14. интернет-версия справочника «КонсультантПлюс»;
15. приложение для мобильных устройств «КонсультантПлюс: Студент»;
16. справочная правовая система «Гарант»;
17. иные ИСС.

11. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине (модулю).

7. Аудиторная база (лекционная аудитория, аудитория для проведения

практических занятий, виртуальные классные комнаты на портале РФЭИ)

8. Организационно-технические средства и аудиовизуальный фондовый материал, мультимедийное оборудование.
9. Комплекты видеофильмов, аудиокниг, CD-дисков по проблемам дисциплины.
10. Интернет.

Приложение 1. Соотнесение результатов обучения по дисциплине соотнесенные с планируемыми результатами освоения образовательной программы.

Название дисциплины	Планируемые результаты обучения		Компетенции					
	код	описание	ОК-1	ОК-12	ОК-13	ОК-17	ПК-16	ПК-19
Программирование	В-1	Владеет навыками поиска, систематизации и представления информации	x	x	x	x	x	x
	З-1	Способен публично выступать, аргументировать свою точку зрения						
	З-2	Способен использовать основные элементы Java-приложений						
	З-3	Способен использовать преобразования типов данных при выполнении операций присваивания, объединения строк, вычисления арифметических выражений и вызова метода						
	З-4	Знает основные технологии программирования						
	У-1	осознает сущность и значение информации в развитии современного общества						
	У-2	Работает с информацией из различных источников						
	У-3	Способен управлять процессами создания и использования информационных сервисов						
	У-4	Умеет использовать соответствующий математический аппарат и инструментальные средства для обработки, анализа и систематизации информации по изучаемой теме						
	В-2	Владеет навыками преобразования типов данных при выполнении операций присваивания, объединения строк, вычисления арифметических выражений и вызова метода						
	В-3	Владеет навыками грамотного составления кода программ и управления доступом к объектам, навыками многопоточного программирования, работы с параллельным доступом и взаимодействием между потоками						

Приложение 2. Фонд оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине, входящей в состав рабочей программы дисциплины Программирование

Направление подготовки **38.03.05 (080500) Бизнес-информатика**
Профиль **Информационный бизнес**
Квалификация (степень) **Бакалавр**
Утверждена **21 декабря 2015 г.**

1. Перечень компетенций с указанием этапов их формирования в процессе освоения образовательной программы

Региональный финансово-экономический институт при формировании компетенций студентов направления подготовки 38.03.05 (080500) «Бизнес-информатика» выделяет три этапа формирования компетенции:

- **начальный**. На этом этапе формируются знаниевые и инструментальные основы компетенции, осваиваются основные категории, формируются базовые умения. В целом, знания и умения носят репродуктивный характер. Студент воспроизводит термины, факты, методы, понятия, принципы и правила. На этом этапе он решает задачи, преимущественно, по образцу. Если студент удовлетворительно отвечает этим требованиям, можно говорить об освоении им базового (начального) уровня компетенции;
- **основной** этап – знания, умения, навыки, обеспечивающие формирование компетенции, значительно возрастают, но ещё не достигают целевых (итоговых) значений. На этом этапе студент осваивает действия с предметными знаниями в конкретной дисциплине и, часто, в междисциплинарном характере действий. Способен самостоятельно решать учебные задачи, внося коррективы в алгоритм своих действий, осуществлять саморегуляцию в ходе работы, переносить знания и умения на новые, возникающие в ходе выполнения работ, условия. Успешное прохождение этого этапа позволяет достичь удовлетворительного уровня сформированности компетенции;
- **завершающий** этап – на этом этапе студент достигает итоговых (целевых) показателей по заявленной компетенции. Он осваивает весь необходимый объём знаний, овладевает всеми умениями и навыками в сфере заявленной компетенции. Он способен использовать эти знания, умения и навыки при решении реальных задач и в нестандартных учебных условиях.

Дисциплина имеет целью участие в формировании следующих компетенций (список в соответствии с РУП направления подготовки, составленным в соответствии с Федеральным государственным образовательным стандартом высшего профессионального образования по направлению подготовки 080500 Бизнес-информатика, утвержденного приказом Министерства образования и науки Российской Федерации от 14 января 2010 г., № 27, в редакции Приказа Министерства образования и науки Российской Федерации от 31.05.2011 № 1975):

1. ОК-1
2. ОК-12
3. ОК-13
4. ОК-17
5. ПК-16
6. ПК-19

Этапы формирования компетенций обычно распределены следующим образом:

1. **Начальный** – формируется в процессе изучения отдельных разделов дисциплины, а успешность его освоения определяется с помощью критериев оценивания компетенции, подробно описанной в разделе [2] этого документа.
2. **Основной** – формируется на этапе успешного завершения всех дисциплин, участвующих в процессе формирования компетенции.
3. **Завершающий** – достигается на основании комплексной междисциплинарной работы, в ходе итоговых практик, экзаменов, выполнении дипломной работы и подтверждении успешного овладения компетенцией.

Завершение дисциплины с точки зрения показателей раздела [2] означает успешное освоение как минимум начального уровня овладения компетенцией.

2. Описание показателей и критериев оценивания компетенций на различных этапах их формирования, описание шкал оценивания

Контроль достижения целевых критериев на этапе текущего формирования компетенции при изучении любых дисциплин направления подготовки осуществляется на основании следующих инструментов (средств оценивания):

1. индивидуальные задания расчётного типа;
2. индивидуальные задания графического типа;
3. индивидуальные задания вербального типа;
4. индивидуальные задания расчётно-графического типа;
5. индивидуальные темы рефератов по заданной теме;

6. индивидуальные темы эссе по заданной теме;
7. индивидуальные задания для выполнения контрольных работ;
8. тесты в ЭИОС по темам дисциплины:
    a. базовый уровень
    b. высокий уровень
    c. повышенный уровень
9. задания для выполнения лабораторных работ;
10. вопросы для защиты лабораторных работ;
11. задания для подготовки и защиты докладов;
12. сценарии ролевых игр;
13. сценарии мастер-классов;
14. задания для выполнения курсовых работ (проектов);
15. задания для выполнения научно-исследовательских работ;
16. задания для прохождения практик;
17. вопросы к экзамену;
18. вопросы к государственному экзамену;
19. задания для выполнения выпускных квалификационных работ.

Основными типами промежуточного контроля являются тестирования вербального и невербального типов в ЭИОС РФЭИ.

Эти тесты различаются по характеру стимульного материала.

В вербальных типах заданий основным содержанием работы испытуемых являются операции с понятиями, мыслительные действия, осуществляемые в словеснологической форме. Составляющие эти методики задания апеллируют к памяти, воображению, мышлению в их опосредованной языковой форме. Они очень чувствительны к различиям в языковой культуре, уровню образования, профессиональным особенностям. Вербальный тип заданий наиболее распространён в компетентностных тестах, тестах достижений, при оценке специальных способностей. Невербальные тесты — это такой тип методик, в которых тестовый материал представлен в наглядной форме (в виде картинок, чертежей, графических изображений и т. п.). От испытуемых требуется понимание вербальных инструкций, само же выполнение заданий опирается на перцептивные и моторные функции.

Невербальные тесты уменьшают влияние языковых различий на результат испытания. Они также облегчают процедуру тестирования испытуемых с нарушением речи, слуха или с умеренным уровнем подготовки. Невербальные тесты широко используются при оценке начального этапа формирования компетенции.

Программа изучения дисциплины составлена таким образом, что успешное её освоение возможно с различными результатами. Все задания разделены на обязательные и необязательные. Успешное выполнение всех обязательных заданий означает достижение удовлетворительного уровня по освоению дисциплины.

Количество обязательных заданий текущего контроля не менее 65% от

общего количества заданий. Все обязательные задания предполагают возможность повторного выполнения (как автоматически, так и в ряде случаев по согласованию/дополнительному разрешению). Успешное выполнение всех обязательных заданий гарантирует студенту оценку «удовлетворительно» в зачётной книжке, если изучение этой дисциплины предполагает выставление оценки.

Необязательный уровень включает задания высокой и повышенной (относительно высокой) сложности. Их успешное выполнение необязательно для студента, однако их выполнение непосредственно влияет на оценку по дисциплине, а также более глубокий уровень освоения предметной областью дисциплины. Успешное завершение всех заданий высокой сложности предполагает получение оценки «хорошо», а повышенной сложности «отлично» при оценивании результатов освоения дисциплины.

Текущий подход является формализованным для всех дисциплин направления подготовки «Бизнес-информатика» и **обязателен к применению в рамках текущей дисциплины**.

В связи с различиями в части применения дисциплины на разных формах обучения и конкретных профилях здесь приводится полные сведения о способе формирования оценки.

1. Если по дисциплине в РУПе не предусмотрен промежуточный контроль (в РУПе по дисциплине указан только ОДИН итоговый экзамен)

Накопленная оценка по дисциплине рассчитывается с помощью взвешенной суммы оценок за отдельные формы текущего контроля знаний следующим образом:

$О_{накопленная}=n_1 О_{текущий1} + n_2 \cdot О_{текущий\ 2} + n_3 \cdot О_{текущий\ 3} + \ldots + n_i \cdot О_{текущий\ i,}$ где

$О_{текущий1}$ – оценка за текущее компьютерное тестирование (базовый, минимальный уровень)

$О_{текущий2}$ – оценка за текущее компьютерное тестирование (высокий уровень освоения)

$О_{текущий3}$ – оценка за текущее компьютерное тестирование (повышенной сложности)

$О_{текущий4}$ – оценка за эссе

…

$О_{текущий\ i}$ – оценка за реферат, доклад и т.п.

$n_1, n_2, n_3 \ldots n_i$ - веса оценок за отдельные формы текущего контроля ($О_{текущий1}, О_{текущий2}, О_{текущий3} \ldots О_{текущий\ i}$)

n_1=**0.6**, n_2=**0.2**, n_3=**0.1** n_4=**0.1**

Сумма весов оценок за отдельные формы текущего контроля, которые учитываются в накопленной оценке, должна быть равна единице (нормализуются):

$\sum ni = 1$

Способ округления накопленной оценки текущего контроля: **в пользу студента.**

Результирующая оценка по дисциплине (которая пойдёт в диплом и является критерием оц) рассчитывается следующим образом:

$О_{результ} = k_1 \cdot О_{накопл} + k_2 \cdot О_{экз}$, где

$O_{накопл}$ – накопленная оценка по дисциплине

$O_{экз}$ – оценка за экзамен

k_1 – вес накопленной оценки по дисциплине

k_2 – вес экзаменационной оценки по дисциплине

Сумма весов ($k_1 + k_2$) должна быть равна единице: $\sum k_i = 1$, при этом, $0,2 \leq k_1 \leq 0,8$. Вес итоговой аттестации не может быть менее 20% от всей дисцпилины.

Для текущей дисциплины $\boldsymbol{k_1 = 0,8}$

Способ округления экзаменационной и результирующей оценок: среднее арифметическое.

2. Если по дисциплине в РУПе предусмотрен промежуточный контроль (в РУПе по дисциплине указано БОЛЕЕ одного экзамена)

Итоговая накопленная оценка по дисциплине рассчитывается следующим образом:

$$O_{\text{накопленная Итоговая}} = (O_{\text{промежуточная 1}} + O_{\text{промежуточная 2}} + \ldots + O_{\text{накопленная i}}) : \text{на число этапов,}$$

$O_{промежуточная\ 1}$ – промежуточная оценка 1 этапа/модуля

$$O_{\text{промежуточная 1}} = m_1 \cdot O_{\text{накопленная 1 этапа}} + m_2 \cdot O_{\text{промежуточный экзамен 1 этапа}}$$

Сумма весов ($m_1 + m_2$) должна быть равна единице, при этом, $0,2 \leq m_1 \leq 0,8$

$O_{промежуточная\ 2}$ – промежуточная оценка 2 этапа/модуля

$$O_{\text{промежуточная 2}} = m_3 \cdot O_{\text{накопленная 2 этапа}} + m_4 \cdot O_{\text{промежуточный экзамен 2 этапа}}$$

Сумма весов ($m_3 + m_4$) должна быть равна единице, при этом, $0,2 \leq m_3 \leq 0,8$

$O_{накопленная\ 1\ этапа}$, $O_{накопленная\ 2\ этапа}$ рассчитываются по приведенной выше формуле расчета накопленной оценки (за каждый этап)

$O_{накопленная\ i}$ – накопленная оценка последнего этапа/модуля перед итоговым экзаменом

$O_{накопленная\ i}$ рассчитывается по приведённой выше формуле расчёта накопленной оценки **(для последнего этапа/модуля перед итоговым экзаменом)**

Результирующая оценка по дисциплине (которая идёт в диплом и является одним из критериев оценивания достижения основного этапа освоения компетенции) рассчитывается следующим образом:

$$O_{\text{результ}} = k_1 \cdot O_{\text{накопленная Итоговая}} + k_2 \cdot O_{\text{Итоговый экз}}$$

$O_{Итоговый\ экз}$ – оценка за **итоговый** экзамен

Сумма весов ($k_1 + k_2$) должна быть равна единице: $\sum k_i = 1$, при этом, $0,2 \leq k_1 \leq 0,8$

Способ округления накопленных, промежуточных, экзаменационных и результирующей оценок: **среднее арифметическое**

3. Типовые контрольные задания и иные материалы, необходимые для оценки знаний, умений, навыков и опыта деятельности, характеризующие этапы формирования компетенций в процессе освоения образовательной деятельности.

В соответствии с описанием показателей и критериев оценивания, подробно описанные в пункте 2 этого документа, здесь приводится неполный список **примеров** тестовых заданий.

См. приложение 2.1 «Типовые контрольные задания», являющееся частью рабочей программы дисциплины.

4. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и опыта деятельности, характеризующие этапы формирования компетенций

Система текущего контроля успеваемости и промежуточной аттестации студентов предусматривает решение следующих задач:

- оценка качества освоения студентами основной профессиональной образовательной программы (оцениваются знания, умения и навыки);
- аттестация студентов на соответствие их персональных достижений поэтапным требованиям соответствующей основной профессиональной образовательной программы;
- поддержание постоянной обратной связи и принятие оптимальных решений в управлении качеством обучения студентов на уровне преподавателя, кафедры, факультета и института целиком.

Текущий контроль успеваемости и промежуточная аттестация является основным механизмом оценки качества подготовки студентов (согласно требованиям ФГОС) и формой контроля учебной работы студентов.

Оценка качества подготовки студентов осуществляется в двух основных направлениях: оценка уровня освоения дисциплины и оценка компетенций студентов. Предметом оценивания являются знания, умения, компетенции обучающихся.

Промежуточная аттестация студентов проводится по учебной дисциплине в сроки, предусмотренные учебными планами и годовыми календарными учебными графиками в порядке, утверждённом в вузе.

Каждая компетенция формируется на всех этапах обучения студента в процессе изучения ряда дисциплин, а после, использования междисциплинарных знания для выполнения дипломной работы и практик.

Знания, умения и навыки постепенно формируют целевую компетенцию. Поэтому существенно отличаются и методы контроля промежуточной и итоговой оценки достижения компетенций.

Промежуточные методы контроля включают в себя автоматические и

неавтоматические методы контроля, такие как тестирование или аттестация/не аттестация по выполнению требуемых видов работ.

С целью определения уровня овладения компетенциями, в заданные логикой преподавания дисциплины сроки проводится текущий и промежуточный контроль знаний, умений и навыков каждого обучающегося. Все виды текущего контроля осуществляются в соответствии с формой задания (см. п.2 «описание показателей и критериев оценивания…»).

Процедура оценивания компетенций обучающихся основана на следующих условиях:

1. Периодичность проведения оценки (минимум 1 раз на каждую рассматриваемую тему в дисциплине).
2. Многоступенчатость: оценка (как автоматически с помощью ЭИОС или преподавателем) и самооценка обучающегося, обсуждение результатов и комплекс мер по устранению недостатков.
3. Единство используемой технологии для всех обучающихся, выполнение условий сопоставимости результатов оценивания.

5. Показатели и критерии оценивания сформированности компетенций

Основным критерием итоговой сформированности любой компетенции является успешное завершение обучения студентом, выполнение и защита дипломной работы и государственного экзамена, прохождение и защита практик.

Успешное завершение дисциплины означает достижение очередного шага в формировании компетенции. Критерием успешного завершения дисциплины является как минимум выполнение всех обязательных требований (заданий) из перечня в пункте 2 этого документа. Критерии успешного завершения каждого из заданий определяются в самих заданиях. Примеры заданий можно посмотреть в п.3 этого документа.

Шкалы оценивания предусматривают детальный ответ на вопрос об уровне освоения дисциплины и, посредством оценивания процедур знаний, умений и навыков, показателей оценивания сформированности компетенции.

Урок 1. Введение в программирование на JAVA

Тест по теме «Введение в Java»

Общая группа

Выберите верное утверждение относительно метода.

- Метод — это атрибут, определяющий свойство конкретной абстракции.
- Метод реализует абстракцию.
- Метод — это категория объектов.
- Метод — это действие, определяющее поведение для конкретной абстракции.
- Метод — это форма для создания операций.

Выберите верное утверждение относительно объекта.

- Объект — это ссылка на атрибут.
- Объект — это то, из чего рождается класс.
- Объект — это экземпляр класса.
- Объект — это переменная.
- Объект — это форма для создания конкретной реализации абстракции.

Какая строка содержит конструктор в описании этого класса?

```
public class Counter {  // (1)
```

```
  int current, step;

  public Counter(int startValue, int stepValue) {  // (2)
    set(startValue);
    setStepValue(stepValue);
  }

  public int get() {  // (3)
    return current;
  }

  public void set(int value) {  // (4)
    current = value;
  }

  public void setStepValue(int stepValue) {  // (5)
    step = stepValue;
  }

}
```

Выберите один правильный ответ.

- Код, обозначенный (1), — это конструктор.
- Код, обозначенный (2), — это конструктор.
- Код, обозначенный (3), — это конструктор.
- Код, обозначенный (4), — это конструктор.
- Код, обозначенный (5), — это конструктор.

Учитывая, что `Thing` — это класс, как много объектов и как много ссылочных переменных создается в следующем программном коде?

```
Thing item, stuff;
item = new Thing();
Thing entity = new Thing();
```

Выберите два правильных ответа.

- Один объект создается.
- Два объекта создается.
- Три объекта создается.

- Одна ссылочная переменная создается.
- Две ссылочные переменные создаются.
- Три ссылочные переменные создаются.

Какое утверждение о члене экземпляра (instance member) верно?

Выберите один правильный ответ.

- Член экземпляра принадлежит экземпляру, а не классу в целом.
- Член экземпляра — это всегда поле.
- Член экземпляра также называется статическим членом.
- Член класса всегда представляет операцию.
- Член экземпляра никогда не является методом.

Выберите один правильный ответ.

Объекты передают сообщения в Java при помощи

- вызова статических методов классов друг друга.
- вызова методов экземпляра друг друга.
- изменения статических переменных классов друг друга.
- изменения полей друг друга.

Учитывая следующий код, какое утверждение верно?

```
class A {
  int value1;
}

class B extends A {
  int value2;
}
```

Выберите два правильных ответа.

- ☐ Класс A расширяет класс B.
- ☐ Класс A наследует от класса B.
- ☐ Класс B — это подкласс класса A.
- ☐ Объекты класса A имеют поле `value2`.
- ☐ Объекты класса B имеют поле `value1`.
- ☐ Класс B — это суперкласс для класса A.

Какую команду следует использовать для того, чтобы скомпилировать следующий код, содержащийся в файле `SmallProg.java`?

```
public class SmallProg {

  public static void main (String[] args) {
    System.out.println("Goog luck!");
  }

}
```

Выберите один правильный ответ.

- `java SmallProg`
- `javac SmallProg.java`
- `java SmallProg.main`
- `java SmallProg.Java`
- `javac SmallProg`

Какую команду следует использовать для того, чтобы выполнить метод `main()` класса с именем `SmallProg`?

Выберите один правильный ответ.

- `java SmallProg.main()`
- `java SmallProg`
- `java SmallProg.java`
- `javac SmallProg`
- `java SmallProg.class`

Урок 2. Основы языка

Тест по теме «Основы языка»

Общая группа

Что из следующего не является разрешенным идентификатором? Выберите один правильный ответ.

- a2z.
- 52pickup.
- _class.
- ca$h,
- dipus.
- total#.

Выберите одно верное утверждение.

Ключевые слова в языке Java — это

- `exit`, `class` и `while`
- `static`, `unsigned` и `long`
- `try`, `catch` и `thrown`
- `new` и `delete`
- `for`, `while` и `next`
- `return`, `goto` и `default`

Является ли это завершенным и допустимым комментарием?

```
/* // */
```

Выберите один правильный ответ.

- ○ Эта комбинация комментариев неверна, и компилятор отклонит ее.
- ○ Нет, блок комментария (`/* ... */`) не заканчивается, так как однострочный комментарий (`//` ...) комментирует закрывающую часть.
- ○ Это полностью верный комментарий. Компилятор игнорирует часть `//` .

Что из перечисленного не представляет примитивное значение в Java? Выберите два правильных ответа,

- ☐ “t”.
- ☐ 'к'.
- ☐ false.
- ☐ “hello”.
- ☐ 50.5F.

Какие из следующих примитивных типов данных не являются целыми типами? Выберите три правильных ответа.

- ☐ Тип double.
- ☐ Tim float.
- ☐ Тип short.
- ☐ Тип boolean.
- ☐ Тип byte.

Какой целый тип в Java имеет диапазон в точности от -2147483648 (-2^{31}) до 2147483647 (2^{31-1}) включительно?

Выберите один правильный ответ.

- Тип int.
- Тип char.
- Тип long.
- Тип byte.
- Тип short.

Какая из следующих строк содержит правильные объявления?

Выберите три правильных ответа.

- `ch\u0061r a = 'a';`
- `ch'a'r a = 'a';`
- `char 'a' = 'a';`
- `char \u0061 = 'a';`
- `char a = '\u0061';`

Если данный код находится внутри метода, то какое из утверждений верно?

```
int a, b;
b = 5;
```

Выберите один правильный ответ.

- Локальная переменная b проинициализирована, но не объявлена.
- Локальная переменная b не объявлена.
- Локальная переменная a не объявлена.
- Локальная переменная a объявлена, но не проинициализирована.
- Локальная переменная b объявлена, но не проинициализирована.

В каком из этих объявлений переменной она остается непроинициализированной до момента явной инициализации?

Выберите один правильный ответ.

- Объявление переменной экземпляра типа `int`.
- Объявление статической переменной типа `float`.
- Объявление статической переменной типа `Object`.
- Объявление локальной переменной типа `float`.
- Объявление переменной экземпляра типа `int[ ]`.

Что получится в результате компиляции этого класса?

```
import java.util.*;
package com.acme.toolkit;

public class AClass {
   public Other anInstance;
}

class Other {
   int value;
}
```

Выберите один правильный ответ.

- Компиляция класса выполнится неудачно, так как класс `Other` должен быть объявлен как `public`.
- Компиляция класса выполнится неудачно, так как объявление `package` не должно употребляться после оператора `import`.
- Компиляция класса выполнится неудачно, так как класс `Other` должен быть определен в файле, названном `Other.java`.
- Компиляция файла произойдет без ошибок.
- Компиляция класса выполнится неудачно, так как оператор `import` никогда не должен располагаться на вершине файла.
- Компиляция класса выполнится неудачно, так как класс `Other` не был еще объявлен, когда класс `AClass` на него сослался.

Верно ли утверждение: “Пустой файл является корректным файлом исходного кода”?

- Да
- Нет

Какое из нижеприведенных объявлений метода `main ()` правильно, если необходимо запустить Java-приложение на выполнение? Выберите два правильных ответа.

- ☐ `public static int main(String[] args) { /* ... */ }`
- ☐ `public static void main(String args) { /* ... */ }`
- ☐ `final public static void main(String[] arguments) { /* ... */ }`
- ☐ `public int main(String[] args, int args) { /* ... */ }`
- ☐ `static void main(String[] args) { /* ... */ }`
- ☐ `static public void main(String[] args) { /* ... */ }`

Что из перечисленного относится к зарезервированным ключевым словам?

Выберите три правильных ответа.

- ☐ void.
- ☐ String.
- ☐ public.
- ☐ args.
- ☐ main.
- ☐ static.

Урок 3. Операторы и присваивания

Операторы и присваивания

Общая группа

Каков самый простой способ для преобразования значения символа `c` в `int`? Выберите один правильный ответ.

- `int i = c.`
- `int i = Character.getNumericValues (c).`
- `int i = (int) c.`

Что произойдет в результате попытки откомпилировать и выполнить следующий класс?

```
public class Assignment {
    public static void main(String[] args) {
        int a, b, c;
        b = 10;
        a = b = c = 20;
        System.out.println(a);
    }
}
```

Выберите один правильный ответ.

- Компиляция выполнится успешно, и после выполнения будет выведено `20`.
- Компиляция выполнится неудачно, потому что оператор присваивания `a = b = c = 20` недопустим.
- Компиляция выполнится неудачно, так как компилятор определит, что переменная c в операторе присваивания `a = b = c = 20` не была проинициализирована.
- Компиляция выполнится успешно, и после выполнения будет выведено `10`.

Что произойдет в результате попытки откомпилировать и выполнить следующую программу?

```
public class MyClass {
    public static void main(String[] args) {
        String a, b, c;
        c = new String("mouse");
        a = new String("cat");
        b = a;
        a = new String("dog");
        c = b;

        System.out.println(c);
    }
}
```

Выберите один правильный ответ.

- ○ Программа откомпилируется неудачно.
- ○ В результате выполнения программы будет выведено `dog`.
- ○ В результате выполнения программы будет выведено `mouse`.
- ○ В результате выполнения программы с равной вероятностью будет выведено либо `cat`, либо `dog`.
- ○ В результате выполнения программы будет выведено `cat`.

При вычислении какого из следующих выражений будет использоваться вещественная арифметика? Выберите три правильных ответа.

- ☐ `2 * 3`
- ☐ `0x10 * 1L * 300.0`
- ☐ `2 / 3 + 5 / 7`
- ☐ `2.4 + 1.6`
- ☐ `2.0 * 3.0`

Каково будет значение выражения `1 / 2 + 3 / 2 + 0.1`? Выберите один правильный ответ.

- ○ `1.6`
- ○ `2.1`
- ○ `1.1`
- ○ `2`
- ○ `1`

Что будет в результате попытки откомпилировать и выполнить эту программу?

```
public class Integers {
```

```
    public static void main(String[] args) {
        System.out.println(0x10 + 10 + 010);
    }
}
```

Выберите один правильный ответ.

- ○ После выполнения программа напечатает `30`.
- ○ После выполнения программа напечатает `101010`.
- ○ После выполнения программа напечатает `28`.
- ○ После выполнения программа напечатает `36`.
- ○ Программа не откомпилируется. Компилятор выразит недовольство выражением `0x10 + 10 + 010`.
- ○ После выполнения программа напечатает `34`.

Какое из следующих выражений допустимо? Выберите три правильных ответа.

- ☐ `(1 * * 1)`
- ☐ `(- -1)`
- ☐ `(+ + 1)`
- ☐ `(+-+-+-1)`
- ☐ `(--1)`
- ☐ `(- 1 -)`

Какое будет получено значение после вычисления выражения `(- -1-3 * 10 / 5-1)` ? Выберите один правильный ответ.

- `-6`
- `10`
- `7`
- `8`
- `-8`

Какое из этих присваиваний недопустимо?

Выберите один правильный ответ.

- `double d = 0x12345678;`
- `int other = (int) true;`
- `long l = 012;`
- `float f = -123;`
- `short s = 12;`

Какое из утверждений верно?

Выберите три правильных ответа.

- ☐ Результатом выражения `(4 + 1.0f)` будет `float`-значение `5.0f`.
- ☐ Результатом выражения `(1 + 2 + "3")` будет строка `33`.
- ☐ Результатом выражения `('a' + 1)` будет `char`-значение `'b'`.
- ☐ Результатом выражения `(10 / 9)` будет `int`-значение `1`.
- ☐ Результатом выражения `("1" + 2 + 3)` будет строка `15`.

Что произойдет в результате попытки откомпилировать и выполнить следующую программу?

```
public class Prog1 {
    public static void main(String[] args) {
        int k = 1;
        int i = ++k + k++ + + k;
        System.out.println(i);
    }
}
```

Выберите один правильный ответ.

- Программа не откомпилируется. Компилятор выразит недовольство выражением `++k + k++ + + k`.
- Программа откомпилируется и после выполнения напечатает значение `8`.
- Программа откомпилируется и после выполнения напечатает значение `4`.
- Программа откомпилируется и после выполнения напечатает значение `3`.
- Программа откомпилируется и после выполнения напечатает значение `7`.

Какая неверная строка первой вызовет ошибку на этапе компиляции следующей?

```
public class MyClass {
    public static void main(String[] args) {
        char c;
        int i;
        c = 'a'; // (1)
        i = c;   // (2)
        i++;     // (3)
        c = i;   // (4)
        c++;     // (5)
    }
}
```

Выберите один правильный ответ.

- Строка `(4)`.
- Строка `(2)`.
- Все строки допустимы. Программа откомпилируется успешно.
- Строка `(3)`.
- Строка `(5)`.
- Строка `(1)`.

Что произойдет в результате попытки откомпилировать и выполнить следующую программу?

```
public class Cast {
    public static void main(String[] args) {
        byte b = 128;
        int  i = b;
        System.out.println(i);
    }
}
```

Выберите один правильный ответ.

- Компилятор откажется компилировать программу, потому что `128` выходит за рамки допустимого диапазона для значений `byte`.
- Компилятор откажется компилировать программу, потому что не сможет присвоить `byte` в `int` без преобразования типа.
- Программа откомпилируется, но на этапе выполнения будет выброшено исключение `ClassCastException`.
- Программа откомпилируется и после выполнения напечатает `255`.
- Программа откомпилируется и после выполнения напечатает `128`.

Что напечатает следующая программа во время выполнения?

```
public class EvaluationOrder {
    public static void main(String[] args) {
        int[] array = { 4, 8, 16 };
        int i = 1;
        array[++i] = --i;
        System.out.println(array[0] + array[1] + array[2]);
    }
}
```

Выберите один правильный ответ.

- ○ `14`
- ○ `24`
- ○ `20`
- ○ `13`
- ○ `21`

В результате каких из следующих выражений будет получено `true`?

Выберите два правильных ответа.

- ☐ `(null != null)`.
- ☐ `(4 <= 4)`.
- ☐ `(true & false)`
- ☐ `(false | true)`.
- ☐ `(! true)`.

Какие утверждения верны?

Выберите три правильных ответа.

- ☐ Оператор получения остатка `%` может использоваться только с целыми операндами.
- ☐ Значения типа short находятся в диапазоне от `-128` до `+127` включительно.
- ☐ `(+15)` является разрешенным выражением.
- ☐ Идентификаторы в Java чувствительны к регистру.
- ☐ Арифметические операторы `*`, `/` и `%` имеют приоритет одного и того же уровня.

Какое из утверждений относительно того, какие строки будут выведены на консоль следующей программой, верно?

```
public class BoolOp {

    static void op(boolean a, boolean b) {
        boolean c = a != b;
        boolean d = a ^ b;
        boolean e = c == d;
        System.out.println(e);
    }

    public static void main(String[] args) {
        op(false, false);
        op(true, false);
        op(false, true);
        op(true, true);
    }

}
```

Выберите три правильных ответа.

- ☐ Первая строка будет содержать `false`.
- ☐ По крайней мере одна строка будет содержать `true`.
- ☐ По крайней мере одна строка будет содержать `false`.
- ☐ Первая строка будет содержать `true`.
- ☐ Все строки напечатают одно и то же.

Что произойдет во время выполнения следующей программы? Выберите один правильный ответ.

```
public class OperandOrder {
    public static void main(String[] args) {
        int i = 0;
        int[] a = { 3, 6 };
        a[i] = i = 9;
        System.out.println(i + " " + a[0] + " " + a[1]);
    }
}
```

- ○ Будет выведено `"9 3 6"`.
- ○ Будет выброшено исключение `ArrayIndexOutOfBoundsException`.
- ○ Будет выведено `"9 9 6"`.
- ○ Будет выведено `"9 3 9"`.
- ○ Будет выведено `"9 0 6"`.

Какое из утверждений верно относительно результата работы следующей программы?

```
public class Logic {
    public static void main(String[] args) {
        int i = 0;
        int j = 0;
```

```
        boolean t = true;
        boolean r;

        r = (t & 0 < (i += 1));
        r = (t && 0 < (i += 2));
        r = (t | 0 < (j += 1));
        r = (t || 0 < (j += 2));
        System.out.println(i + " " + j);
    }

}
```

Выберите два правильных ответа.

- ☐ Вторая напечатанная цифра будет 2
- ☐ Первая напечатанная цифра будет 2
- ☐ Первая напечатанная цифра будет 3
- ☐ Вторая напечатанная цифра будет 1
- ☐ Первая напечатанная цифра будет 1
- ☐ Вторая напечатанная цифра будет 3

Что будет отображено на экране во время выполнения следующей программы?

```
public class MyClass {

    public static void main(String[] args) {
        test(1 << 32, "1 << 32");
        test(1 << 31, "1 << 31");
        test(1 << 30, "1 << 30");
        test(1, "1");
        test(0, "0");
        test(-1, "-1");
    }

    public static void test(int i, String exp) {
        if ((i >> 1) != (i >>> 1)) System.out.println(exp);
    }
```

```
}
```

Выберите два правильных ответа.

- ☐ `1`.
- ☐ `1 << 32`.
- ☐ `-1`
- ☐ `1 << 30`.
- ☐ `1 << 31`.
- ☐ `0`

Что из следующего не является оператором в Java?

Выберите два правильных ответа.

- ☐ `&`
- ☐ `<<<`
- ☐ `>>>`
- ☐ `&&=`
- ☐ `<=`
- ☐ `%`
- ☐ `%=`

Допустим, что есть переменная `x` типа `int` (которая может содержать отрицательное значение). Какие существуют правильные способы удвоения значения `x`, кроме тех, в которых промежуточные результаты выходят за пределы допустимого диапазона?

Выберите четыре правильных ответа.

- ☐ `x += x;`
- ☐ `x *= 2;`
- ☐ `x <<= 1;`
- ☐ `x = x * 2;`
- ☐ `x << 1;`

Какие из следующих операторов могут использоваться в качестве и целочисленного поразрядного оператора, и булевого логического оператора?

Выберите три правильных ответа.

- ☐ `|`
- ☐ `~`
- ☐ `!`
- ☐ `^`
- ☐ `&`

Ниже заданы объявления, какое из следующих выражений допустимо?

```
byte  b = 1;
char  c = 1;
short s = 1;
int   i = 1;
```

Выберите три правильных ответа.

- ☐ `c = c + b;`
- ☐ `s += i;`
- ☐ `s = b * 2;`
- ☐ `i = b << s;`
- ☐ `b <<= s;`

Что будет отображено на экране во время выполнения следующей программы?

```
public class ParameterPass {
    public static void main(String[] args) {
        int i = 0;
        addTwo(i++);
        System.out.println(i);
    }

    static void addTwo(int i) {
        i += 2;
    }
}
```

Выберите один правильный ответ.

- ○ 0
- ○ 3
- ○ 2
- ○ 1

Что будет в результате попытки откомпилировать и выполнить следующий класс?

```
public class Passing {
    public static void main(String[] args) {
        int a = 0;
        int b = 0;
```

```
        int[] bArr = new int[1];
        bArr[0] = b;

        inc1(a);
        inc2(bArr);

        System.out.println("a=" + a + " b=" + b + " bArr[0]=" +
bArr[0]);
    }

    public static void inc1(int x) {
        x++;
    }

    public static void inc2(int[] x) {
        x[0]++;
    }
}
```

Выберите один правильный ответ.

- Код откомпилируется, и после выполнения будет отображено `a=1 b=1 bArr[0]=1`.
- Код откомпилируется, и после выполнения будет отображено `a=0 b=1 bArr[0]=1`.
- Код откомпилируется, и после выполнения будет отображено `a=0 b=0 bArr[0]=0`.
- На этапе компиляции возникнет ошибка, так как `x[0]++;` является недопустимым выражением.
- Код откомпилируется, и после выполнения будет отображено `a=0 b=0 bArr[0]=1`.

Дан класс:

```
// Filename: Args.java
public class Args {
    public static void main(String[] args) {
```

```
        System.out.println(args[0] + " " + args[args.length - 1]);
    }
}
```

Каков будет результат после выполнения следующей команды?

```
java Args In politics stupidity is not a handicap
```

Выберите один правильный ответ.

- Будет отображено на экране `In handicap`.
- Будет отображено на экране `Args handicap`.
- Будет отображено на экране `In a`.
- Будет отображено на экране `Args a`.
- Будет отображено на экране `Java handicap`.
- Будет выброшено исключение `ArrayIndexOutOfBoundsException`.

Какое выражение должно вызвать ошибку компиляции, если его добавить в приведенную ниже программу в указанное положение?

```
public class ParameterUse {
    public static void main(String[] args) {
        int a = 0;
        final int b = 1;
        int[] c = { 2 };
        final int[] d = { 3 };
        useArgs(a, b, c, d);
    }

    static void useArgs(final int a, int b, final int[] c, int[] d) {
        // INSERT STATEMENT HERE.
    }
}
```

Выберите два правильных ответа.

- ☐ `b = a;`
- ☐ `d[0]++;`
- ☐ `c = d;`
- ☐ `a++;`
- ☐ `c[0]++;`
- ☐ `b++;`

Урок 4. Объявления и управление доступом

Объявления и управление доступом

Общая группа

Учитывая данное ниже объявление, определите, какое выражение вернет размер массива, допуская, что массив был проинициализирован?

```
int[] array;
```

Выберите один правильный ответ.

- `array.length`
- `array[].length()`
- `array[].length`
- `array[].size()`
- `array.size()`
- `array.length()`

Возможно ли создать массив нулевой длины? Выберите один правильный ответ.

- Нет, нельзя создать массивы нулевой длины, но в метод main() возможно передать массив ссылок типа String нулевой длины, когда аргументы программы не определены.
- Да, но только для массивов со ссылками на объекты.
- Да, можно создать массивы любого типа нулевой длины.
- Да, но только для примитивных типов.
- Нет, в Java невозможно создать массивы нулевой длины.

Какое из следующих предложений объявления массива недопустимо? Выберите один правильный ответ.

- `int [][]a = new int [4][4]`
- `int []a[] = new int [4][]`
- `int []a[] = new int [4][4]`
- `int a[][] = new int [4][4]`
- `int a[][] = new int [][4]`

Какие из этих предложений объявления массива недопустимы? Выберите два правильных ответа.

- `int i[][] = { { 1, 2 }, new int[2] }`
- `int i[4] = { 1, 2, 3, 4 }`
- `int i[] = new int[2] { 1, 2 }`
- `int[] i[] = { { 1, 2 }, { 1 }, { }, { 1, 2, 3 } }`
- `int i[][] = new int[][]{ { 1, 2, 3 }, { 4, 5, 6 } }`

Что получится, если попытаться откомпилировать и выполнить следующую

программу?

```
class MyClass {
    public static void main(String[] args) {
        int size = 20;
        int[] arr = new int[size];

        for (int i = 0; i < size; ++i) {
            System.out.println(arr[i]);
        }
    }
}
```

Выберите один правильный ответ.

- Программа скомпилируется и выполнится без ошибок, но не создаст выходной поток.
- Программа скомпилируется и выполнится без ошибок, а также напечатает двадцать раз `null`.
- На этапе компиляции произойдет ошибка, потому что выражение типа массива `int[]` неправильно.
- Программа скомпилируется и выполнится без ошибок, а также напечатает числа от `0` до `19`.
- Программа скомпилируется и выполнится без ошибок, а также напечатает двадцать раз `0`.
- Программа скомпилируется, но на этапе выполнения будет выброшено исключение `ArayIndexOutOfBoundsException`.

Дана следующая программа, какое предложение верно?

```
class MyClass {
    public static void main(String[] args) {
        String[] numbers = { "one", "two", "three", "four" };

        if (args.length == 0) {
```

```
            System.out.println("no arguments");
        } else {
            System.out.println(numbers[args.length] + " arguments");
        }
    }
}
```

Выберите один правильный ответ.

- При вызове программы с пустой строкой аргументов или с тремя аргументами она соответственно выведет `no arguments` или `two arguments`.
- На этапе компиляции произойдет ошибка.
- При вызове программы с пустой строкой аргументов или с тремя аргументами она соответственно выведет `no arguments` или `four arguments`.
- Программа выбросит исключение `NullPointerException`, если при выполнении в нее не будут переданы аргументы.
- При вызове программы с пустой строкой аргументов или с тремя аргументами она соответственно выведет `no arguments` или `three arguments`.

Что получится, если попытаться откомпилировать и выполнить следующую программу?

```
public class DefaultValuesTest {
    int[] ia = new int[1];
    boolean b;
    int i;
    Object o;

    public static void main(String[] args) {
        DefaultValuesTest instance = new DefaultValuesTest();
        instance.print();
    }
```

```
    public void print() {
        System.out.println(ia[0] + " " + b + " " + i + " " + o);
    }
}
```

Выберите один правильный ответ.

- Программа выведет на экран `null false 0 null`.
- Программа выведет на экран `null 0 0 null`.
- Программа выведет на экран `0 false 0 null`.
- На этапе выполнения будет выброшено исключение `java.lang.NullPointerException`.
- Произойдет ошибка на этапе компиляции, потому что переменные не проинианализированы.
- Программа выведет на экран `0 false NaN null`.

Какое объявление метода из представленных ниже допустимо? Выберите один правильный ответ.

- `void method3(void) { /* ... */ }`
- `void method1 { /* ... */ }`
- `method4() { /* ... */ }`
- `void method2() { /* ... */ }`
- `method5(void) { /* ... */ }`

В приведенном ниже коде какие операторы можно поместить в указанную позицию, чтобы это не вызвало ошибок компиляции?

```
public class ThisUsage {
    int planets;
    static int suns;
```

```
    public void gaze() {
        int i;
        // ... Добавьте операторы сюда ...
    }
}
```

- ☐ `this.i = 4;`
- ☐ `this.suns = planets;`
- ☐ `i = this.planets;`
- ☐ `this = new ThisUsage();`
- ☐ `i = this.suns;`

В приведенных ниже парах объявлений методов какие операторы верны?

```
void fly(int distance) {}
int  fly(int time, int speed) { return time*speed; }
void fall(int time) {}
int  fall(int distance) { return distance; }
void glide(int time) {}
void Glide(int time) {}
```

Выберите два правильных ответа.

- ☐ Вторая пара методов скомпилируется корректно, и перегружается метод `fall`.
- ☐ Первая пара методов скомпилируется корректно, и перегружается метод `fly`.
- ☐ Третья пара методов не скомпилируется корректно.
- ☐ Вторая пара методов не скомпилируется корректно.
- ☐ Третья пара методов скомпилируется корректно, и перегружается метод `glide`.

Имеется класс с именем `Book`, какое из представленных объявлений конструкторов допустимо для класса? Выберите один правильный ответ.

- `abstract Book() {}`
- `Book Book() {}`
- `private final Book() {}`
- `void Book() {}`
- `Book(Book b)`
- `public static void Book(String [] args) {}`

Какое утверждение верно? Выберите два правильных ответа.

- ☐ Конструктор может вернуть значение.
- ☐ Конструктор может получить доступ к нестатическим членам класса.
- ☐ Все классы должны определять конструктор.
- ☐ Конструктор должен приниициализировать все поля в классе.
- ☐ Конструктор можно объявить как `private`.

Что получится в результате попытки скомпилировать следующую программу?

```
public class MyClass {

    long var;

    public void MyClass(long param) { var = param; }  // (1)

    public static void main(String[] args) {
        MyClass a, b;
        a = new MyClass();                            // (2)
        b = new MyClass(5);                           // (3)
    }
}
```

- На этапе компиляции в строке (1) произойдет ошибка, так как в конструкторе не задается возвращаемое значение.
- На этапе компиляции в строке (3) произойдет ошибка, так как класс не содержит конструктора, который получает в качестве параметра один аргумент типа `int`.
- На этапе компиляции в строке (2) произойдет ошибка, так как класс не содержит конструктора по умолчанию.
- Программа скомпилируется успешно.

Выберите один правильный ответ.

Для заданного класса какой из способов обращения к нему извне пакета `net.basemaster` является допустимым?

```
package net.basemaster;

public class Base {
    // ...
}
```

Выберите два правильных ответа.

- ☐ Обращение к классу как `Base`.
- ☐ Обращение к классу как `basemaster.Base`.
- ☐ Импорт `net.basemaster.*` и обращение к классу как `Base`.
- ☐ Обращение к классу как `net.basemaster.Base`.
- ☐ Импорт `net.*` и обращение к классу как `basemaster.Base`.

Какое из следующих определений класса является верным определением класса, который не может быть инстанцирован? Выберите один правильный ответ.

- `abstract class Ghost { void haunt() {};}`
- `abstract Ghost {abstract void haunt();}`
- `abstract class Ghost {void haunt();}`
- `class Ghost {abstract void haunt();}`
- `static class Ghost {abstract haunt();}`

Какое из следующих определений класса является верным определением класса, который нельзя расширить? Выберите один правильный ответ.

- `static class Link { }`
- `abstract final class Link { }`
- `final class Link { }`
- `private class Link { }`
- `native class Link { }`
- `abstract class Link { }`
- `class Link { }`

По предложенному ниже определению класса скажите, какие поля доступны извне пакета `com.corporation.project` ? Выберите два правильных ответа.

```
package com.corporation.project;

public class MyClass {
              int i;
    public    int j;
    protected int k;
    private   int l;
}
```

- ☐ Поле `j` доступно из всех классов других пакетов.
- ☐ Поле `l` доступно во всех классах других пакетов.
- ☐ Поле `l` доступно в подклассах только других пакетов.
- ☐ Поле `k` доступно из подклассов других пакетов.
- ☐ Поле `i` доступно из всех классов других пакетов.
- ☐ Поле `k` доступно из всех классов других пакетов.

Оцените степень строгости доступа по умолчанию по сравнению с доступами `public` (общедоступный), `protected` (защищенный) и `private` (закрытый). Выберите один правильный ответ.

- ○ Более строг, чем `protected`, но менее, чем `private`.
- ○ Более строг, чем `public`, но менее, чем `protected`.
- ○ Более строг, чем `private`.
- ○ Менее строг, чем `protected`, в случае, когда оба рассматриваются внутри одного и того же пакета, и более строг, чем `protected`, в случае внешнего пакета.
- ○ Менее строг, чем `public`.

Какое из утверждений о доступности членов верно? Выберите один правильный ответ.

- `Private` -члены вообще недоступны.
- `Private` -члены могут быть доступны только из кода внутри класса этого члена
- Член с уровнем доступа по умолчанию может быть доступен для любого подкласса того класса, в котором он определен.
- Доступ пакетный/по умолчанию для члена можно задать с помощью ключевого слова `default`.
- `Private` -члены всегда доступны из того же пакета.

Какое из утверждений об использовании модификаторов верно? Выберите два правильных ответа.

- ☐ Сами объекты не имеют каких-либо модификаторов доступа, их имеют только объектные ссылки.
- ☐ Подклассы класса должны располагаться в том же пакете, что и класс, который они расширяют.
- ☐ Вы не можете задать модификатор доступа для локальных переменных. Они доступны только внутри блока, в котором объявлены.
- ☐ Если ни один модификатор доступа (`public`, `protected` или `private`) не задан в объявлении члена, то к этому члену можно обратиться только из классов его же пакета и изо всех его подклассов.
- ☐ Локальные переменные можно объявить как `static`.

В предложенном коде какую строку комментария можно раскомментировать, чтобы это не вызвало ошибок?

```
abstract class MyClass {
    abstract void f();
    final    void g() {}
 // final    void h() {}                          // (1)
```

```
    protected static int i;
    private          int j;
}

final class MyOtherClass extends MyClass {
 // MyOtherClass(int n) { m = n; }               // (2)

    public static void main(String[] args) {
        MyClass mc = new MyOtherClass();
    }

    void f() {}
    void h() {}
 // void k() { i++; }                            // (3)
 // void l() { j++; }                            // (4)

    int m;
}
```

Выберите один правильный ответ.

- `MyOtherClass(int n) { m = n; } // (2)`
- `void k() { i++; } // (3)`
- `final void h() // (1)`
- `void l() { j++; } // (4)`

Что получится в результате попытки откомпилировать и выполнить следующую программу?

```
class MyClass {
    static MyClass ref;
    String[] arguments;

    public static void main(String[] args) {
        ref = new MyClass();
        ref.func(args);
    }

    public void func(String[] args) {
```

```
            ref.arguments = args;
        }
    }
```

Выбирете один правильный ответ.

- Произойдет ошибка на этапе компиляции, так как в нестатическом методе `func()` нельзя обращаться к статической переменной `ref`.
- Произойдет ошибка на этапе компиляции, так как аргумент `args`, передаваемый в статический метод `main()`, не может быть передан в нестатический метод `func()`.
- Программа откомпилируется и выполнится успешно.
- Произойдет ошибка на этапе компиляции, так как в статическом методе `main()` не может быть вызван нестатический метод `func()`.
- Произойдет ошибка на этапе компиляции, так как в методе `func()` нельзя присвоить значение статической переменной `args` нестатической переменной `arguments`.
- Программа откомпилируется, но на этапе выполнения выбросит исключение.

Даны несколько объявлений. Какое из утверждений верно?

```
int a;                              // (1)
static int a;                       // (2)
int f() { return a; }               // (3)
static int f() { return a; }        // (4)
```

- Объявления (2) и (3) не могут находиться в одном и том же классе.
- Объявления (2) и (4) не могут находиться в одном и том же классе.
- Объявления (1) и (3) не могут находиться в одном и том же классе.
- Объявления (1) и (4) не могут находиться в одном и том же классе.

Выберите один правильный ответ.

Какое утверждение верно? Выберите один правильный ответ.

- Методы экземпляра могут получить доступ к локальным переменным статических методов.
- Во все методы классов при вызове неявно передается параметр `this`.
- Класс может содержать как статические, так и нестатические переменные, как статические, так и нестатические методы.
- Статический метод может вызвать нестатические методы того же класса с помощью ключевого слова `this`.
- Каждый объект класса имеет собственный экземпляр каждой статической переменной.

Что (если есть) неправильно в этом коде?

```
abstract class MyClass {
    transient int j;
    synchronized int k;

    final void MyClass() {}

    static void f() {}
}
```

Выберите один правильный ответ.

- Поле `j` нельзя объявить как `transient`.
- Поле `k` нельзя объявить как `synchronized`.
- Метод `f()` нельзя объявить как `static`.
- Метод `MyClass()` нельзя объявить как `final`.
- Класс `MyClass` нельзя объявить как `abstract`.
- Все правильно, код откомпилируется без ошибок.

Что из следующего не является допустимым объявлением члена внутри класса? Выберите один правильный ответ.

- `abstract int t;`
- `static int a;`
- `final Object[] fudge = { null };`
- `final static private double PI = 3.14159265358979323846;`
- `native void sneeze();`

Какое утверждение о модификаторах справедливо? Выберите два правильных ответа.

- ☐ Абстрактные классы можно объявить как `final`.
- ☐ Поля можно объявить как `native`.
- ☐ Абстрактные классы не могут содержать неизменяемые методы.
- ☐ Неабстрактные методы можно объявить в абстрактном классе.
- ☐ Классы можно объявить как `native`.

Какое утверждение справедливо? Выберите один правильный ответ.

- `Transient` -поля нельзя сохранить во время сериализации.
- Подкласс класса с абстрактным методом должен обеспечить реализацию для абстрактного метода.
- Исходное состояние объекта массива, создаваемого выражением `int[]a = new int[10]`, будет зависеть от переменной массива `a`
- Конструктор можно объявить как `abstract`.
- Только статические методы могут обращаться к статическим переменным.

Урок 5. Поток команд управления, управление исключениями и диагностические утверждения

Управление исключениями и диагностические утверждения

Общая группа

Что получится в результате попытки откомпилировать и выполнить следующий класс?

```
public class IfTest {
    public static void main(String[] args) {
        if (true)
        if (false)
        System.out.println("a");
        else
        System.out.println("b");
    }
}
```

Выберите один правильный ответ.

- ○ Произойдет ошибка на этапе компиляции, так как компилятор не сможет определить, какому оператору `if` принадлежит какой оператор `else`.
- ○ Произойдет ошибка на этапе компиляции, так как синтаксис оператора `if` неверен.
- ○ Код откомпилируется, и будет выведена на консоль буква `b`.
- ○ Код откомпилируется, но на консоли ничего не будет отображено.
- ○ Код откомпилируется, и будет выведена на консоль буква `a`.

Какое утверждение верно? Выберите три правильных ответа.

- □ Только выражения, значение которых принадлежит типу `boolean`, могут использоваться в качестве условий в операторе `if`.
- □ Оператор `if` может иметь как выражение `if`, так и `else`.
- □ В условном выражении в операторе `if` могут быть вызовы методов.
- □ Если переменные `a` и `b` типа `boolean`, то выражение `(a = b)` может быть условным выражением в операторе `if`.
- □ Оператор `if (false) ; else ;` недопустим.

Что (если есть) неправильно в следующем коде?

```
void test(int x) {
    switch (x) {
        case 1:
        case 2:
        case 0:
        default:
        case 4:
    }
}
```

Выберите один правильный ответ.

- В коде нет ничего неправильного.
- Метка `case 0` должна предшествовать метке `case 1`.
- Каждый блок case должен заканчиваться оператором `break`.
- Метка `default` должна быть самой последней в операторе `switch`.
- Тело оператора `switch` должно содержать по крайней мере один оператор.
- Переменная `x` имеет неправильный тип для выражения под `switch`.

Какие из этих комбинаций типов выражений `switch` и значений меток в префиксах `case` допустимы внутри оператора `switch`? Выберите один правильный ответ.

- Выражение `switch` типа `float` и тип значения метки `case int`.
- Выражение `switch` типа `byte` и тип значения метки `case float`
- Выражение `switch` типа `char` и тип значения метки `case long`.
- Выражение `switch` типа `boolean` и тип значения метки `case boolean`.
- Выражение `switch` типа `int` и тип значения метки `case char`.

Что будет в результате попытки откомпилировать и выполнить следующий код?

```
class MyClass {
    public static void main(String[] args) {
        boolean b = false;
        int i = 1;
        do {
            i++;
            b = !b;
        } while (b);
        System.out.println(i);
    }
}
```

Выберите один правильный ответ.

- ○ Произойдет ошибка на этапе компиляции, так как присваивание `b = !b` недопустимо.
- ○ Код откомпилируется, и на этапе выполнения будет выведено `1`.
- ○ Произойдет ошибка на этапе компиляции, так как `b` является недопустимым условным выражением для оператора `do-while`.
- ○ Код откомпилируется, и на этапе выполнения будет выведено `2`.
- ○ Код откомпилируется, и на этапе выполнения будет выведено `3`.

Что будет отображено при выполнении следующей программы?

```
public class MyClass {
    public static void main(String[] args) {
        int i = 0;
        int j;
        for (j = 0; j < 10; ++j) { i++; }
        System.out.println(i + " " + j);
    }
}
```

Выберите два правильных ответа.

- ☐ Вторая напечатанная цифра будет 10.
- ☐ Первая напечатанная цифра будет 11.
- ☐ Вторая напечатанная цифра будет 9.
- ☐ Первая напечатанная цифра будет 10.
- ☐ Вторая напечатанная цифра будет 11.
- ☐ Первая напечатанная цифра будет 9.

Какое из следующих утверждений допустимо? Выберите один правильный ответ.

- `int j = 10; for (int i = 0, j += 90; i < j; i++) { j--; }`
- `int i = 100; for ( (i > 0); i--) {}`
- `int i, j; for (j = 100; i < j; j--) { i += 2; }`
- `for (int i = 10; i = 0; i--) {}`
- `for (int i = 0, j = 100; i < j; i++, --j) {;}`

Что получится в результате попытки откомпилировать и выполнить следующую программу?

```
class MyClass {
    public static void main(String[] args) {
        int i = 0;
        for (     ; i < 10; i++) ;       // (1)
        for (i = 0;       ; i++) break; // (2)
        for (i = 0; i < 10;    ) i++;    // (3)
        for (     ;       ;    ) ;       // (4)
    }
}
```

- Код откомпилируется и программа выполнится и остановится без вывода данных.
- Произойдет ошибка на этапе компиляции, так как в операторе цикла `for` в строке (2) пропущено выражение в средней секции.
- Код откомпилируется, но на этапе выполнения никогда не завершится.
- Произойдет ошибка на этапе компиляции, так как в операторе `for` в строке (1) пропущено выражение в первой секции.
- Произойдет ошибка на этапе компиляции, так как оператор цикла `for` в строке (4) недопустим.
- Произойдет ошибка на этапе компиляции, так как в операторе цикла `for` в строке (3) пропущено выражение в последней секции.

Выберите один правильный ответ.

Какое утверждение корректно, считая, что оно встречается независимо? Выберите три правильных ответа.

- ☐ `do { break; } while (true);`
- ☐ `while () break;`
- ☐ `switch (1) { default: break; }`
- ☐ `for (;true;) break;`
- ☐ `if (true) { break; }`

По заданному ниже фрагменту программы определите, какие строки будут частью вывода программы?

```
outer:
for (int i = 0; i < 3; i++) {
    for (int j = 0; j < 2; j++) {
        if (i == j) {
            continue outer;
        }
        System.out.println("i=" + i + ", j=" + j);
    }
}
```

Выберите два правильных ответа.

- ☐ i=1, j=2
- ☐ i=2, j=1
- ☐ i=3, j=2
- ☐ i=1, j=0
- ☐ i=3, j=3
- ☐ i=2, j=2
- ☐ i=0, j=1

Что получится в результате попытки откомпилировать и выполнить следующий код?

```
class MyClass {
    public static void main(String[] args) {
        for (int i = 0; i < 10; i++) {
            switch (i) {
                case 0:
                    System.out.println(i);
            }
            if (i) {
                System.out.println(i);
            }
        }
    }
}
```

Выберите один правильный ответ.

- Произойдет ошибка на этапе компиляции из-за неправильного условного выражения в операторе `if`.
- Код откомпилируется и напечатает `0` дважды.
- Код откомпилируется и напечатает ряд от `1` до `10`.
- Код откомпилируется и напечатает `0`.
- Код откомпилируется и напечатает ряд от `0` до `10`.
- Произойдет ошибка на этапе компиляции из-за неправильного выражения под `switch` в операторе `switch`.

Какая из следующих реализаций метода `max()` корректно возвратит наибольшее значение?

```
// (1)
int max(int x, int y) {
    return (if (x > y) { x; } else { y; });
}
// (2)
int max(int x, int y) {
    return (if (x > y) { return x; } else { return y; });
}
// (3)
int max(int x, int y) {
    switch (x < y) {
        case true:
            return y;
        default:
            return x;
    };
}
// (4)
int max(int x, int y) {
    if (x > y) return x;
    return y;
}
```

Выберите один правильный ответ.

- Реализация (3).
- Реализация (2).
- Реализация (1).
- Реализация (4).

Относительно следующего кода какое из утверждений верно?

```
class MyClass {
    public static void main(String[] args) {
        int k = 0;
        int l = 0;
        for (int i = 0; i <= 3; i++) {
            k++;
            if (i == 2) break;
            l++;
        }
        System.out.println(k + ", " + l);
    }
}
```

Выберите один правильный ответ.

- Программа не откомпилируется, если `break` просто удалить.
- Программа выведет `4, 3`, если `break` заменить на `continue`.
- Программа выведет `3, 3`.
- Программа не откомпилируется, если `break` заменить на `return`.
- Программа не откомпилируется.

Какое утверждение справедливо? Выберите два правильных ответа.

- `block: { break block; }` представляет собой допустимый блок операторов
- Оператор `break` может использоваться только в цикле (`while` , `do-while` или `for`) или в операторе `switch` .
- `block: { continue block; }` представляет собой допустимый блок операторов
- `{ continue; }` представляет собой допустимый блок операторов.
- `{{}}` представляет собой допустимый блок операторов.

Какие цифры и в каком порядке будут выведены на экран следующей программой?

```
public class MyClass {
    public static void main(String[] args) {
        int k = 0;
        try {
            int i = 5 / k;
        } catch (ArithmeticException e) {
            System.out.println("1");
        } catch (RuntimeException e) {
            System.out.println("2");
            return;
        } catch (Exception e) {
            System.out.println("3");
        } finally {
            System.out.println("4");
        }
        System.out.println("5");
    }
}
```

Выберите один правильный ответ.

- ○ Будет выведено только `1` и `4` в таком порядке.
- ○ Будет выведено только `1`, `4` и `5` в таком порядке.
- ○ Будет выведено только `1`, `2` и `4` в таком порядке.
- ○ Будет выведено только `1`, `2`, `4` и `5` в таком порядке.
- ○ Будет выведено только `5`.
- ○ Будет выведено только `3` и `5` в таком порядке.

Какое утверждение относительно следующей программы истинно?

```
public class Exceptions {
    public static void main(String[] args) {
        try {
            if (args.length == 0) return;
            System.out.println(args[0]);
        } finally {
            System.out.println("The end");
        }
    }
}
```

Выберите два правильных ответа.

- ☐ Если программа будет запущена с одним аргументом, то его же просто и выведет.
- ☐ Если программа будет запущена без аргументов, то выведет `The end`.
- ☐ Если программа будет запущена с одним аргументом, то выведет его, а затем фразу `The end`.
- ☐ Если программа будет запущена без аргументов, то ничего не выведет.
- ☐ Будет выброшено на этапе выполнения исключения `ArrayIndexOutOfBoundsException`.

Что получится в результате компиляции и выполнения программы?

```
public class MyClass {
    public static void main(String[] args) {
        RuntimeException re = null;
        throw re;
    }
}
```

Выберите один правильный ответ.

- Произойдет ошибка на этапе компиляции, поскольку нельзя выбросить `re`.
- Программа откомпилируется без ошибок, выполнится и завершится без какого-либо вывода.
- Программа откомпилируется без ошибок, на этапе выполнения будет выброшено исключение `java.lang.NullpointerException`.
- Программа откомпилируется без ошибок, на этапе выполнения будет выброшено исключение `java.lang.RuntimeException`.
- Произойдет ошибка на этапе компиляции, поскольку в прототипе метод `main()` не объявлено, что он выбрасывает исключение `RuntimeException`.

Какое утверждение истинно? Выберите два правильных ответа.

- ☐ Блоки `finally` выполняются тогда и только тогда, когда исключение выбрасывается из соответствующего блока `try`.
- ☐ В переопределенном методе должны быть объявлены в `throw` классы исключений, как в методе, который он переопределяет.
- ☐ В методе `main()` программы может быть объявлено, что он выбрасывает проверяемые исключения.
- ☐ Если исключение не перехватывается в методе, то метод будет прерван, и будет продолжено нормальное выполнение.
- ☐ Метод, объявляющий о том, что он выбрасывает исключение некоторого типа, может выбросить экземпляр любого подкласса этого класса исключения.

Какие цифры и в каком порядке будут выведены следующей программой?

```
public class MyClass {
    public static void main(String[] args) {
        try {
            f();
        } catch (InterruptedException e) {
            System.out.println("1");
            throw new RuntimeException();
        } catch (RuntimeException e) {
            System.out.println("2");
            return;
        } catch (Exception e) {
            System.out.println("3");
        } finally {
            System.out.println("4");
        }
        System.out.println("5");
    }

    // InterruptedException подкласс Exception.
    static void f() throws InterruptedException {
        throw new InterruptedException("Time for lunch.");
    }
}
```

- ○ Будет выведено `1`, `2`, `4` и `5` в таком порядке.
- ○ Будет выведено `1`, `4` и `5` в таком порядке.
- ○ Будет выведено `1`, `2` и `4` в таком порядке.
- ○ Будет выведено `1` и `4` в таком порядке.
- ○ Будет выведено `5`.

Выберите один правильный ответ.

Какие цифры и в каком порядке будут выведены?

```
public class MyClass {
    public static void main(String[] args) throws
InterruptedException {
        try {
            f();
            System.out.println("1");
        } finally {
            System.out.println("2");
        }
        System.out.println("3");
    }

    // InterruptedException is a direct subclass of Exception.
    static void f() throws InterruptedException {
        throw new InterruptedException("Time to go home.");
    }
}
```

Выберите один правильный ответ.

- Будет выведено `1`, `2` и `3` в таком порядке.
- Будет выведено `3` и `2` в таком порядке.
- Будет выведено `2` и `throw InterruptedException`.
- Будет выведено `1` и `2` в таком порядке.
- Будет выведено `2` и `3` в таком порядке.
- Будет выведено `1` и `3` в таком порядке.

Что неправильно в следующем коде? Выберите один правильный ответ.

```
public class MyClass {
    public static void main(String[] args) throws A {
        try {
            f();
        } finally {
            System.out.println("Done.");
        } catch (A e) {
            throw e;
        }
    }

    public static void f() throws B {
        throw new B();
    }
}

class A extends Throwable {}

class B extends A {}
```

- Объявление класса `A` некорректно.
- Единственный блок `try` не может сопровождаться и блоком `finally` и блоком `catch`.
- Блок `finally` должен идти после блока `catch` в методе `main()`.
- В методе `main()` должно быть объявлено, что он выбрасывает `B`.
- В блоке `catch` в методе `main()` должно быть объявлено, что он отлавливает а не `A`.

Какой минимальный список исключений должен быть задан в переопределенном методе `f()` в следующем коде в операторе `throws`, чтобы компиляция прошла успешно?

```
class A {
   // InterruptedException подкласс Exception.
    void f() throws ArithmeticException, InterruptedException {
        div(5, 5);
    }
    int div(int i, int j) throws ArithmeticException {
        return i/j;
    }
}

public class MyClass extends A {
    void f() /* throws [...list of exceptions...] */ {
        try {
            div(5, 0);
        } catch (ArithmeticException e) {
            return;
        }
        throw new RuntimeException("ArithmeticException was
expected.");
    }
}
```

- ○ Требуется определить, что он выбрасывает `InterruptedException`.
- ○ Никакие исключения не требуется задавать.
- ○ Требуется определить, что он выбрасывает и `ArithmeticException`, и `InterruptedException`.
- ○ Требуется определить, что он выбрасывает `ArithmeticException`.
- ○ Требуется определить, что он выбрасывает `RuntimeException`.

Полагая, что диагностические утверждения разрешены, какое из следующих предложений приведет к ошибке? Выберите два правильных ответа.

- ☐ `assert false : true;`
- ☐ `assert false : false;`
- ☐ `assert true : false;`
- ☐ `assert true : true;`

Какое имя класса исключения будет выброшено диагностическим утверждением? Выберите один правильный ответ.

- ○ `AssertionError`
- ○ Зависит от диагностического утверждения.
- ○ `FailedAssertion`
- ○ `Error`
- ○ `RuntimeException`
- ○ `AssertionException`

Что может послужить причиной игнорирования диагностических утверждений? Выберите один правильный ответ.

- Использование как опций компиляции, так и опций выполнения.
- Использование соответствующих опций компиляции.
- Ничего.
- Использование соответствующих опций выполнения.

Для данного ниже метода какие его вызовы приведут к выбрасыванию исключения, полагая, что диагностические утверждения включены?

```
static int inv(int value) {
    assert value > -50 : value < 100;
    return 100 / value;
}
```

Выберите два правильных ответа.

- ☐ `inv(100)`.
- ☐ `inv(50)`.
- ☐ `inv(150)`.
- ☐ `inv(-50)`.
- ☐ `inv(0)`.

Что получится в результате компиляции и выполнения следующего кода с включенным механизмом диагностических утверждений?

```
public class TernaryAssertion {
    public static void assertBounds(int low, int high, int value) {
        assert ( value > low ? value < high : false )
            : (value < high ? "too low" : "too high" );
    }
    public static void main(String[] args) {
        assertBounds(100, 200, 150);
    }
}
```

- ○ Класс откомпилируется, и программа выполнится без ошибок.
- ○ Произойдет ошибка на этапе компиляции, потому что оператор `assert` некорректный.
- ○ Класс откомпилируется, и при выполнении будет выброшено исключение `AssertionError` с сообщением об ошибке `too high`.
- ○ Класс откомпилируется, и при выполнении будет выброшено исключение `AssertionError` с сообщением об ошибке `too tow`.
- ○ Произойдет ошибка на этапе компиляции, потому что имя метода `assertBounds` не может начинаться с ключевого слова `assert`.

Выберите один правильный ответ.

Что справедливо о классе `AssertionError`? Выберите два правильных ответа.

- ☐ Имеет метод `getErrorMessage()`.
- ☐ Это проверяемое исключение.
- ☐ Имеет метод `toString()`.
- ☐ Может быть отловлено конструкцией `try-catch`.

Какой из этих классов является суперклассом для `AssertionError?` Выберите один правильный ответ.

- ○ `Exception`.
- ○ `Error`.
- ○ `Object`.
- ○ `Throwable`.

Для каких классов следующие команды включат диагностические утверждения?

```
java -ea -da:com... net.example.LaunchTranslator
```

Выберите два правильных ответа.

- ☐ `java.lang.String`
- ☐ `java.lang.AssertionError`
- ☐ `com.example.Translator`
- ☐ `dot.com.Boom`
- ☐ `net.example.LaunchTranslator`

Урок 6. Объектно-ориентированное программирование

Объектно-ориентированное программирование

Общая группа

Какое утверждение верно? Выберите два правильных ответа.

- ☐ В Java оператор `extends` используется для задания наследования.
- ☐ Неизменяемый (`final`) класс может быть абстрактным.
- ☐ Класс, в котором все члены объявлены как `private` , не может быть объявлен как `public` .
- ☐ Все члены суперкласса наследуются подклассом.
- ☐ Подкласс не абстрактного класса может быть объявлен как `abstract` .

Какое утверждение верно? Выберите два правильных ответа.

- ☐ Класс, не являющийся неизменяемым (`final`), может расширять любое количество классов.
- ☐ Класс может расширять любое количество других классов.
- ☐ Каждый объект Java имеет публичный (`public`) метод с именем equals.
- ☐ Наследование задает отношение «имеет» (has-a) между суперклассом и его подклассом.
- ☐ Каждый объект Java имеет публичный (`public`) метод с именем length.

Какое утверждение верно? Выберите 2 правильных ответа.

- ☐ Для двух классов возможно, чтобы каждый из ни был суперклассом другого
- ☐ Возможно определить в подклассе метод с таким же именем и параметрами, как и метод в суперклассе
- ☐ В подклассе должны быть определены все методы из суперкласса
- ☐ Возможно определить в подклассе поле с таким же именем, как и поле в суперклассе

Для представленных ниже классов и объявлений какое из утверждений верно? Выберите 3 правильных ответа.

```
// Классы
class Foo {
    private int i;
    public void f() { /* ... */ }
    public void g() { /* ... */ }
}

class Bar extends Foo {
    public int j;
```

```
    public void g() { /* ... */ }
}

// Объявления:
// ...
    Foo a = new Foo();
    Bar b = new Bar();
// ...
```

- ☐ Выражение `b.i = 3;` допустимо
- ☐ Выражение `a.j = 5;` допустимо
- ☐ Выражение `a.g();` допустимо
- ☐ Класс `Bar` является корректным подклассом `Foo`
- ☐ Выражение `b.f();` допустимо

Какое утверждение верно? Выберите правильный ответ.

- ○ `Private` -методы нельзя переопределить в подклассе
- ○ Список параметров переопределенного метода должен быть подмножеством списка параметров метода, который он переопределяет
- ○ Переопределенный метод может иметь другой возвращаемый тип, чем в методе, который он переопределяет.
- ○ В переопределенном методе может быть объявлено, что он может выбросить больше исключений, чем метод, который он переопределяет.
- ○ Подкласс может переопределить любой метод суперкласса

Даны классы `A`, `B` и `C`, причем `B` расширяет `A`, `C` расширяет `B` и все классы реализуют метод `void doIt()`. Как можно обратиться к методу `doIt()` в `A` из метода класса `C`? Выберите правильный ответ.

- `super.doIt();`
- `super.super.doIt();`
- `A.this.doIt();`
- `((A) this).doIt();`
- Это не возможно
- `this.super.doIt();`
- `doIt();`

Что получится в результате компилирования и выполнения следующего кода?

```
public class MyClass {
    public static void main(String[] args) {
        C c = new C();
        System.out.println(c.max(13, 29));
    }
}

class A {
    int max(int x, int y) { if (x>y) return x; else return y; }
}

class B extends A{
    int max(int x, int y) { return super.max(y, x) - 10; }
}

class C extends B {
    int max(int x, int y) { return super.max(x+10, y+10); }
}
```

Выберите 1 правильный ответ.

- Компиляция выполнится успешно, и во время выполнения будет напечатано 39
- Компиляция не будет выполнена успешно, потому что вызов метода `max()` неоднозначен
- Компиляция выполнится успешно, и во время выполнения будет напечатано 23
- Компиляция выполнится успешно, и во время выполнения будет напечатано 13
- Компиляция выполнится успешно, и во время выполнения будет напечатано 29
- Компиляция не будет выполнена успешно, потому что метод `max()` в `B` передает аргументы при вызове `super.max(y, x)` в неверном порядке

Для данного ниже кода какой самый простой оператор `print` может быть добавлен в метод print()?

```
public class MyClass extends MySuperclass {
    public static void main(String[] args) {
        MyClass object = new MyClass();
        object.print();
    }
    public void print() {
        // Добавьте сюда код который напечатает
        // строку "Hello, world!" из класса Message
    }
}

class MySuperclass {
    Message msg = new Message();
}

class Message {
    // Сообщение, которое следует напечатать:
    String text = "Hello, world!";
}
```

Выберите 1 правильный ответ.

- `System.out.println(text);`
- `System.out.println(Message.text);`
- `System.out.println(object.super.msg.text);`
- `System.out.println(msg.text);`
- `System.out.println(super.msg.text);`
- `System.out.println(object.msg.text);`

Какой из конструкторов может быть добавлен в класс `MySub`, чтобы это не вызвало ошибку при компиляции?

```
class MySuper {
    int number;
    MySuper(int i) { number = i; }
}

class MySub extends MySuper {
    int count;
    MySub(int cnt, int num) {
        super(num);
        count=cnt;
    }

    // Конструктор
}
```

Выберите 1 правильный ответ.

- `MySub(int cnt) { super(); count = cnt; }`
- `MySub(int cnt) { count = cnt; super(cnt); }`
- `MySub(int cnt) { this(cnt, cnt); }`
- `MySub()`
- `MySub(int cnt) { super(cnt); this(cnt, 0); }`
- `MySub(int cnt) { count = cnt; }`

Какое утверждение верно? Выберите правильный ответ.

- Вызов `super()` или `this()` должен быть первым оператором в теле конструктора.
- Если ни `super()`, ни `this()` не объявлены первыми операторами в теле конструктора, то будет неявно добавлен первым оператором `this()`
- Если и подкласс, и его суперкласс не содержат никаких конструкторов, то во время выполнения неявный конструктор по умолчанию подкласса вызовет `super()`
- Если `super()` является первой строкой в теле конструктора, то вызов `this()` может быть вторым
- Вызов `super()`, как первый оператор в теле конструктора подкласса, всегда будет выполняться, поскольку все суперклассы имеют конструктор по умолчанию

Какой будет результат работы программы? Выберите 1 правильный ответ.

```
public class MyClass {
    public static void main(String[] args) {
        B b = new B("Test");
    }
}
```

```
class A {
    A() { this("1", "2"); }

    A(String s, String t) { this(s + t); }

    A(String s) { System.out.println(s); }
}

class B extends A {
    B(String s) { System.out.println(s); }

    B(String s, String t) { this(t + s + "3"); }

    B() { super("4"); };
}
```

- ○ Будет выведено `Test`.
- ○ Будет выведено `4`, потом `Test`.
- ○ Будет выведено `Test`, потом `Test`.
- ○ Будет выведено `123`, потом `Test`.
- ○ Будет выведено `12`, потом `Test`.

Какие утверждения об интерфейсах верны? Выберите 2 правильных ответа.

- ☐ Члены интерфейса всегда могут быть объявлены как `static`
- ☐ Интерфейсы могут быть расширены любым количеством других интерфейсов
- ☐ Интерфейсы позволяют реализовывать множественное наследование реализации
- ☐ Интерфейс может расширить любое количество других интерфейсов
- ☐ Члены интерфейса никогда не могут быть `static`

Какие из этих объявлений полей допустимы в теле интерфейса? Выберите 3

правильных ответа.

- ☐ `public static int answer = 42;`
- ☐ `public int answer = 42;`
- ☐ `final static int answer = 42;`
- ☐ `private final static int answer = 42;`

Какое объявление может быть добавлено в указанную строку следующего кода, чтобы это не вызвало ошибок компиляции? Выберите 2 правильных ответа.

```
interface MyConstants {
    int r = 42;
    int s = 69;
    // Код
}
```

- ☐ `public static MAIN = 15;`
- ☐ `int AREA = r * s;`
- ☐ `protected int CODE = 31337;`
- ☐ `final double circumference = 2 * Math.PI * r;`

Какое утверждение относительно следующей программы верно?

```
// Filename: MyClass.java
public class MyClass {
    public static void main(String[] args) {
        A[] arrA;
        B[] arrB;

        arrA = new A[10];
        arrB = new B[20];
        arrA = arrB;       // (1)
        arrB = (B[]) arrA; // (2)
        arrA = new A[10];
```

```
        arrB = (B[]) arrA; // (3)
    }
}

class A {}

class B extends A {}
```

Выберите один правильный ответ.

- На этапе компиляции произойдет ошибка из-за присваивания в строке (1).
- Во время выполнения будет выброшено исключение `java.lang.ClassCastException` в строке присваивания (2).
- Во время выполнения будет выброшено исключение `java.lang.ClassCastException` в строке присваивания (3).
- Программа успешно откомпилируется и выполнится без ошибок, даже если приведение (`B[ ]`) в операторах строк (2) и (3) удалить.
- Программа успешно откомпилируется и выполнится без ошибок, даже если приведение (`B[ ]`) в операторах строк (2) и (3) удалить.

Какая из строк первой приведет к ошибке на этапе компиляции следующей программы? Выберите правильный ответ.

```
// Filename: MyClass.java
class MyClass {
    public static void main(String[] args) {
        MyClass a;
        MySubclass b;

        a = new MyClass();              // (1)
        b = new MySubclass();           // (2)

        a = b;                          // (3)
        b = a;                          // (4)
```

```
        a = new MySubclass();          // (5)
        b = new MyClass();             // (6)
    }
}

class MySubclass extends MyClass {}
```

- Строка (6).
- Строка (5).
- Строка (2).
- Строка (4).
- Строка (3).
- Строка (1).

Даны описания и объявления ссылок. Какая операция присваивания из приведенных в вариантах ответа допустима?

```
// Описания:
interface I1 {}
interface I2 {}
class C1 implements I1 {}
class C2 implements I2 {}
class C3 extends C1 implements I2 {}

// Объявления ссылок:
// ...
C1 obj1;
C2 obj2;
C3 obj3;
```

Выберите один правильный ответ.

- `obj3 = obj2;`
- `obj2 = obj1;`
- `obj3 = obj1;`
- `I2 c = obj1;`
- `I1 b = obj3;`
- `I1 a = obj2;`

Даны описания и объявления ссылок. Что можно сказать об операции `y = (Sub) x` ?

```
// Описания классов:
class Super {}
class Sub extends Super {}
// Объявления ссылок:
// ...
Super x;
Sub y;
// ...
```

Выберите один правильный ответ.

- Недопустимо для компиляции.
- Допустимо для компиляции, но может быть неверно при выполнении.
- Абсолютно правомерно для выполнения, но приведение (Sub) не является строго обязательным.
- Абсолютно правомерно для выполнения, а приведение (Sub) строго обязательно.

Даны описания и операторы объявлений. Какая одна из операций присваивания разрешена во время компиляции?

```
// Описания:
```

```
interface A {}
class B {}
class C extends B implements A {}
class D implements A {}

// Операторы объявлений:
// [...]
    B b = new B();
    C c = new C();
    D d = new D();
// [...]
```

Выберите один правильный ответ.

- `d = (D) c;`
- `c = b;`
- `A a = d;`
- `d = c;`
- `c = d;`

Какие буквы будут напечатаны, когда выполнится следующая программа?

```
// Filename: MyClass.java
public class MyClass {
    public static void main(String[] args) {
        B b = new C();
        A a = b;
        if (a instanceof A) System.out.println("A");
        if (a instanceof B) System.out.println("B");
        if (a instanceof C) System.out.println("C");
        if (a instanceof D) System.out.println("D");
    }
}

class A {}
class B extends A {}
class C extends B {}
class D extends C {}
```

Выберите три правильных ответа.

- ☐ `A`
- ☐ `B`
- ☐ `C`
- ☐ `D`

Даны три класса - `A`, `B` и `C`, где `B` является подклассом `A` и `C` является подклассом `B`. Какое из следующих логических выражений истинно для случая, когда объект, обозначенный ссылкой `o`, реально порождается от класса `B`, а не от `A` или `C`? Выберите один правильный ответ.

- ○ `(o instanceof B) && !((o instanceof A) || (o instanceof C))`
- ○ `(o instanceof B) && (!(o instanceof A))`
- ○ `(o instanceof B) && (!(o instanceof C))`
- ○ `!((o instanceof A) || (o instanceof B))`

Когда программа выполнится, она выведет все следующие буквы `I`, `J`, `C` и `D`. Это утверждение истинно или ложно?

```
public class MyClass {
    public static void main(String[] args) {
        I x = new D();
        if (x instanceof I) System.out.println("I");
        if (x instanceof J) System.out.println("J");
        if (x instanceof C) System.out.println("C");
        if (x instanceof D) System.out.println("D");
    }
}

interface I{}
interface J{}
```

```
class C implements I {}
class D extends C implements J {}
```

- Истинно.
- Ложно.

Что получится, если попробовать откомпилировать и выполнить следующую программу?

```
public class Polymorphism {
    public static void main(String[] args) {
        A ref1 = new C();
        B ref2 = (B) ref1;
        System.out.println(ref2.f());
    }
}

class A           { int f() { return 0; } }
class B extends A { int f() { return 1; } }
class C extends B { int f() { return 2; } }
```

Выберите один правильный ответ.

- Программа откомпилируется без ошибок, но во время выполнения будет выброшено исключение `ClassCastException`.
- Программа откомпилируется без ошибок, и на этапе выполнения будет выведено `0`.
- Программа откомпилируется без ошибок, и на этапе выполнения будет выведено `1`.
- Программа откомпилируется без ошибок, и на этапе выполнения будет выведено `2`.
- Произойдет ошибка на этапе компиляции.

Что получится, если попробовать откомпилировать и выполнить следующую

программу?

```
public class Polymorphism2 {
    public static void main(String[] args) {
        A ref1 = new C();
        B ref2 = (B) ref1;
        System.out.println(ref2.g());
    }
}

class A {
    private int f() { return 0; }
    public int g() { return 3; }
}

class B extends A {
    private int f() { return 1; }
    public int g() { return f(); }
}

class C extends B {
    public int f() { return 2; }
}
```

Выберите один правильный ответ.

- Программа откомпилируется без ошибок, и на этапе выполнения будет выведено `2`.
- Программа откомпилируется без ошибок, и на этапе выполнения будет выведено `3`.
- Произойдет ошибка на этапе компиляции.
- Программа откомпилируется без ошибок, и на этапе выполнения будет выведено `0`.
- Программа откомпилируется без ошибок, и на этапе выполнения будет выведено `1`.

Какое утверждение относительно следующего кода истинно? Выберите два

правильных ответа.

```
public interface HeavenlyBody { String describe(); }

class Star {
    String starName;
    public String describe() { return "star " + starName; }
}

class Planet extends Star {
    String name;
    public String describe() {
        return "planet " + name + " orbiting star " + starName;
    }
}
```

- ☐ Произойдет ошибка на этапе компиляции, если имя `starName` заменить именем `name` в объявлении класса `Star`.
- ☐ Произойдет ошибка на этапе компиляции, если имя `starName` заменить на имя `bodyName` в объявлении класса `Star`.
- ☐ Экземпляр `Planet` (Планета) является допустимым экземпляром `HeavenlyBody` (Небесное тело).
- ☐ Произойдет ошибка на этапе компиляции.
- ☐ Использование наследования обоснованно, поскольку `Planet` (Планета) является `Star` (Звездой).

Урок 7. Вложенные классы и интерфейсы

Вложенные классы и интерфейсы

Общая группа

Что получится в результате попытки откомпилировать и выполнить следующий код? Выберите один правильный ответ.

```
public class MyClass {
    public static void main(String[] args) {
        Outer objRef = new Outer();
        System.out.println(objRef.createInner().getSecret());
    }
}

class Outer {
    private int secret;
    Outer() { secret = 123; }

    class Inner {
        int getSecret() { return secret; }
    }

    Inner createInner() { return new Inner(); }
}
```

- ○ Код успешно откомпилируется, и будет выведено на экран во время выполнения `123`.
- ○ На этапе компиляции произойдет ошибка, потому что поле `secret` не может быть доступно из метода `getSecret()`.
- ○ На этапе компиляции произойдет ошибка, потому что метод `getSecret()` невидим из метода `main()` класса `MyClass`.
- ○ На этапе компиляции произойдет ошибка, потому что класс `Inner` нельзя объявить внутри класса `Outer`.
- ○ На этапе компиляции произойдет ошибка, потому что недопустимо передавать из метода `createInner()` объекты класса `Inner` в методы за пределами класса `Outer`.

Какое утверждение о вложенных классах верно? Выберите два правильных ответа.

- ☐ Класс-статический член может содержать нестатические поля.
- ☐ Для каждого экземпляра внешнего класса может существовать много экземпляров классов-нестатических членов.
- ☐ Интерфейс-статический член может содержать нестатические поля.
- ☐ Интерфейс-статический член имеет собственный внешний экземпляр.
- ☐ Экземпляр класса - статического члена имеет собственный внешний экземпляр.

Что получится, если откомпилировать и выполнить следующий код? Выберите один правильный ответ.

```
public class MyClass {
    public static void main(String[] args) {
        State st = new State();
        System.out.println(st.getValue());
        State.Memento mem = st.memento();
        st.alterValue();
```

```
        System.out.println(st.getValue());
        mem.restore();
        System.out.println(st.getValue());
    }

    public static class State {
        protected int val = 11;

        int getValue() { return val; }
        void alterValue() { val = (val + 7) % 31; }
        Memento memento() { return new Memento(); }

        class Memento {
            int val;

            Memento() { this.val = State.this.val; }
            void restore() { ((State) this).val = this.val; }
        }
    }
}
```

- Программа успешно откомпилируется, и во время выполнения на экран будет выведено 11, 18 и 11.
- На этапе компиляции произойдет ошибка, поскольку выражение `((State) this).val` в методе `restore()` класса `Memento` недопустимо.
- На этапе компиляции произойдет ошибка, поскольку в статическом методе `main()` пытаются создать новый экземпляр класса - статического члена `State`.
- На этапе компиляции произойдет ошибка, поскольку класс - нестатический член Memento содержит поле с таким же именем, как поле во внешнем классе `State`.
- На этапе компиляции произойдет ошибка, поскольку объявление класса `State.Memento` недоступно из метода `main()`.
- На этапе компиляции произойдет ошибка, поскольку выражение `State.this.val` в конструкторе класса `Memento` недопустимо.

Что получится, если откомпилировать и выполнить следующую программу? Выберите один правильный ответ.

```
public class Nesting {
    public static void main(String[] args) {
        B.C obj = new B().new C();
    }
}


class A {
    int val;
    A(int v) { val = v; }
}

class B extends A {
    int val = 1;
    B() { super(2); }

    class C extends A {
        int val = 3;
        C() {
            super(4);
            System.out.println(B.this.val);
            System.out.println(C.this.val);
            System.out.println(super.val);
        }
    }
}
```

- Программа успешно откомпилируется, и во время выполнения на экран будет выведено `1`, `3` и `4` в таком порядке.
- На этапе компиляции произойдет ошибка.
- Программа успешно откомпилируется, и во время выполнения на экран будет выведено `2`, `3` и `4` в таком порядке.
- Программа успешно откомпилируется, и во время выполнения на экран будет выведено `3`, `2` и `1` в таком порядке.
- Программа успешно откомпилируется, и во время выполнения на экран будет выведено `1`, `4` и `2` в таком порядке.

Какое утверждение о следующей программе верно? Выберите два правильных ответа.

```
public class Outer {
    public void doIt() {
    }
    public class Inner {
        public void doIt() {
        }
    }
    public static void main(String[] args) {
        new Outer().new Inner().doIt();
    }
}
```

- ☐ Полное имя класса `Inner` — это `Outer.Inner`.
- ☐ На этапе компиляции произойдет ошибка.
- ☐ Метод `doIt()` в классе `Inner` скрывает метод `doIt()` класса `Outer`.
- ☐ Метод `doIt()` в классе `Inner` перегружает метод `doIt()` класса `Outer`.
- ☐ Метод `doIt()` в классе `Inner` переопределяет метод `doIt()` класса `Outer`.

Какое утверждение верно? Выберите правильный ответ.

- ○ Все вложенные классы можно объявить как статические.
- ○ Методы во всех вложенных классах можно объявить статическими.
- ○ Все вложенные классы можно объявить как классы-статические члены.
- ○ Классы-нестатические члены должны иметь или модификатор доступа по умолчанию, или `public`.
- ○ Классы-статические члены могут содержать нестатические методы.

Дано объявление `interface IntHolder { int getInt(); }`

Какие из следующих методов допустимы?

```
//----(1)----
    IntHolder makeIntHolder(int i) {
        return new IntHolder() {
            public int getInt() { return i; }
        };
    }


//----(2)----
    IntHolder makeIntHolder(final int i) {
```

```
        return new IntHolder {
            public int getInt() { return i; }
        };
    }


//----(3)----
    IntHolder makeIntHolder(int i) {
        class MyIH implements IntHolder {
            public int getInt() { return i; }
        }
        return new MyIH();
    }


//----(4)----
    IntHolder makeIntHolder(final int i) {
        class MyIH implements IntHolder {
            public int getInt() { return i; }
        }
        return new MyIH();
    }


//----(5)----
    IntHolder makeIntHolder(int i) {
        return new MyIH(i);
    }
    static class MyIH implements IntHolder {
        final int j;
        MyIH(int i) { j = i; }
        public int getInt() { return j; }
    }
```

Выберите два правильных ответа.

- ☐ Метод отмечен как (1).
- ☐ Метод отмечен как (2).
- ☐ Метод отмечен как (3).
- ☐ Метод отмечен как (4).
- ☐ Метод отмечен как (5).

Какие утверждения верны? Выберите два правильных ответа.

- ☐ Если класс-нестатический член вложен внутрь класса с именем `Outer`, то методы этого класса-нестатического члена должны использовать префикс `Outer.this` для доступа к членам класса `Outer`.
- ☐ Если `objRef` определяет экземпляр какого-либо вложенного класса внутри класса `Outer`, то выражение `(objRef instanceof Outer)` будет верно.
- ☐ Никакие другие статические члены, за исключением неизменяемых статических полей, не могут быть объявлены внутри класса-нестатического члена.
- ☐ Безымянный класс не может содержать конструктор.
- ☐ Все поля в каждом вложенном классе должны быть объявлены как неизменяемые.

Какое утверждение верно? Выберите один правильный ответ.

- Классы верхнего уровня можно объявить статическими.
- Безымянные классы не могут быть статическими.
- Локальные классы можно объявить статическими.
- Никакие классы не могут быть объявлены как статические.
- Классы, объявленные как члены класса верхнего уровня, можно объявить статическими.

Урок 8. Время жизни объекта

Время жизни объекта

Общая группа

Какое утверждение верно? Выберите правильный ответ.

- Как только объект станет недостижим, будет немедленно подобран сборщиком мусора.
- Если объект `obj1` имеет доступ к объекту `obj2`, который пригоден для сборки мусора, то `obj1` также пригоден для сборки мусора.
- Если объект `obj1` доступен из объекта `obj2` и объект `obj2` доступен из объекта `obj1`, то объекты `obj1` и `obj2` непригодны для сборки мусора.
- Как только объект становится пригодным для сборки мусора, он остается в таком состоянии пригодности, пока не будет уничтожен.
- Объекты можно явно уничтожить с помощью ключевого слова `delete`.

Определите место в следующей программе, в которой объект, первоначально доступный по ссылке `arg1`, становится пригодным для сборки мусора. Выберите правильный ответ.

```
public class MyClass {
    public static void main(String[] args) {
        String msg;
        String pre = "This program was called with ";
        String post = " as first argument.";

        String arg1 = new String((args.length > 0) ?
                          "'" + args[0] + "'" :
                          "<no argument>");
        msg = arg1;
```

```
            arg1 = null;              // (1)
            msg = pre + msg + post;   // (2)
            pre = null;               // (3)

            System.out.println(msg);

            msg = null;               // (4)
            post = null;              // (5)
            args = null;              // (6)
        }
    }
```

- После строки (3).
- После строки (2).
- После строки (5).
- После строки (1).
- После строки (6).
- После строки (4).

Какое утверждение верно? Выберите правильный ответ.

- У всех объектов есть метод `finalize()`.
- Объекты могут быть разрушены явным вызовом метода `finalize()`.
- Код класса, переопределенный метод `finalize()` которого не содержит явно вызова метода `finalize()` своего суперкласса, не будет скомпилирован.
- Если выбрасывается исключение во время выполнения метода `finalize()` пригодного объекта, то исключение игнорируется и объект уничтожается.
- Метод `finalize()` можно объявить с любым уровнем доступа.

Какое утверждение верно? Выберите правильный ответ.

- В теле метода `finalize()` можно обращаться только к тем объектам, которые пригодны для сборки мусора.
- Метод `finalize()` может быть перегружен.
- Метод `finalize()` должен быть объявлен с модификатором доступа `protected`.
- Переопределенный метод `finalize()` в каждом классе может выбрасывать проверяемые исключения.
- Код, в котором явно вызывается метод `finalize()`, не будет скомпилирован.

Какое утверждение характеризует гарантированное поведение сборки мусора и механизма уничтожения? Выберите правильный ответ.

- Объект, пригодный для сборки мусора, в конце концов будет уничтожен сборкой мусора.
- Объект, ставший однажды пригодным для сборки мусора, больше никогда не станет доступным живому потоку.
- Метод `finalize()` никогда не будет вызван более одного раза у объекта.
- Если объект `A` становится пригодным для сборки мусора перед объектом `B`, то объект `A` будет разрушен перед объектом `B`.
- Объекты не будут уничтожены, пока на них существуют ссылки.

Какие из этих блоков статических инициализаторов можно добавить в данный класс после комментария?

```
public class MyClass {
    private static int count = 5;
    final static int STEP = 10;
    boolean alive;
    // Добавьте блок статического инициализатора здесь
```

```
}
```

Выберите три правильных ответа.

- ☐ `static {;}`
- ☐ `static { count = 1; }`
- ☐ `static ;`
- ☐ `static { STEP = count; }`
- ☐ `static { alive = true; count = 0; }`
- ☐ `static { count += STEP; }`

Что получится, если попробовать откомпилировать и выполнить следующий код?

```
public class MyClass {
    public static void main(String[] args) {
        MyClass obj = new MyClass(l);
    }
    static int i = 5;
    static int l;
    int j = 7;
    int k;

    public MyClass(int m) {
        System.out.println(i + ", " + j + ", " + k + ", " + l + ", "
+ m);
    }
    { j = 70; l = 20; }
    static { i = 50; }
}
```

Выберите один правильный ответ.

- Произойдет ошибка на этапе компиляции, поскольку поле `k` не будет проинициализировано, когда оно используется.
- Произойдет ошибка на этапе компиляции, поскольку блок инициализатора экземпляра пытается присвоить значение статическому полю.
- Компиляция произойдет успешно, и будет выведено при выполнении `5, 7, 0, 20, 0.`
- Компиляция произойдет успешно, и будет выведено при выполнении `50, 70, 0, 20, 20.`
- Компиляция произойдет успешно, и будет выведено при выполнении `5, 70, 0, 20, 0.`
- Компиляция произойдет успешно, и будет выведено при выполнении `50, 70, 0, 20, 0.`

Для данного класса какой блок инициализатора экземпляра, добавленный в указанное положение, позволит откомпилировать класс без ошибок?

```
public class MyClass {
    static int gap = 10;
    double length;
    final boolean active;

    // Добавьте код здесь
}
```

Выберите правильный ответ.

- `{ active = (gap > 5); length = 5.5 + gap;}`
- `instance { active = true; }`
- `{ length = 4.2; }`
- `{ ; }`
- `MyClass { gap += 5; }`
- `{ gap = 5; length = (active ? 100 : 200) + gap; }`

Что получится, если попробовать откомпилировать и выполнить следующую программу?

```
public class Initialization {
    private static String msg(String msg) {
        System.out.println(msg); return msg;
    }

    public Initialization() { m = msg("1"); }

    { m = msg("2"); }

    String m = msg("3");

    public static void main(String[] args) {
        Object obj = new Initialization();
    }
}
```

Выберите правильный ответ.

- Компиляция пройдет успешно, и будет выведено при выполнении `1`, `2` и `3`.
- Компиляция пройдет успешно, и будет выведено при выполнении `1`, `3` и `2`.
- Компиляция пройдет успешно, и будет выведено при выполнении `3`, `1` и `2`.
- Компиляция пройдет успешно, и будет выведено при выполнении `2`, `3` и `1`.
- Программа не откомпилируется

Какая из отмеченных строк в следующем коде может быть раскомментирована с помощью удаления символов `//`, так что код будет по-прежнему компилироваться без ошибок?

```
class GeomInit {
//  int width = 14;             /* Строка А */
    {
//      area = width * height;  /* Строка Б */
    }
    int width = 37;
    {
//      height = 11;            /* Строка В */
    }
    int height, area;
//  area = width * height;      /* Строка Г */
    {
//      int width = 15;         /* Строка Д */
        area = 100;
    }
};
```

Выберите два правильных ответа.

- ☐ Строка Г.
- ☐ Строка А.
- ☐ Строка Б.
- ☐ Строка В.
- ☐ Строка Д.

Урок 9. Потоки выполнения

Потоки выполнения

Общая группа

Какой из способов запуска нового потока правильный?

Выберите правильный ответ.

- Создать новый объект `Thread` и вызвать метод `start()`.
- Создать новый объект Thread и вызвать метод `resume()`.
- Создать новый объект `Thread` и вызвать метод `run()`.
- Создать новый объект `Thread` и вызвать метод `begin()`.
- Просто создать новый объект `Thread`. Поток запустится автоматически.

При расширении класса `Thread` для того, чтобы получить поведение потока, какой метод следует переопределить? Выберите правильный ответ.

- `resume()`
- `behavior()`
- `start()`
- `begin()`
- `run()`

Какое утверждение верно? Выберите два правильных ответа.

- ☐ Вызов метода `run()` у объекта, реализующего `Runnable`, создаст новый поток.
- ☐ Класс `Thread` реализует `Runnable`.
- ☐ Классы, реализующие интерфейс `Runnable`, должны определить метод `start()`
- ☐ Программа завершится, когда закончит свое выполнение последний не-демон.
- ☐ Класс `Thread` — это абстрактный класс.

Что получится, если попытаться скомпилировать и выполнить следующую программу?

```
public class MyClass extends Thread {
    public MyClass(String s) { msg = s; }
    String msg;
    public void run() {
        System.out.println(msg);
    }

    public static void main(String[] args) {
        new MyClass("Hello");
        new MyClass("World");
    }
}
```

Выберите правильный ответ.

- Программа скомпилируется неуспешно.
- Программа скомпилируется без ошибок и будет печатать во время выполнения `Hello` и `World`, но порядок непредсказуем.
- Программа скомпилируется без ошибок и будет печатать никогда не завершающийся поток `Hello` и `World`.
- Программа скомпилируется без ошибок и будет печатать каждый раз во время выполнения `Hello` и `World`, в указанном порядке.
- Программа скомпилируется без ошибок и во время выполнения просто завершится без какого-либо вывода.

Какое утверждение в отношении следующей программы с гарантией будет верным?

```
public class ThreadedPrint {
    static Thread makeThread(final String id, boolean daemon) {
        Thread t = new Thread(id) {
            public void run() {
                System.out.println(id);
            }
        };
        t.setDaemon(daemon);
        t.start();
        return t;
    }

    public static void main(String[] args) {
        Thread a = makeThread("A", false);
        Thread b = makeThread("B", true);
        System.out.print("End\n");
    }
}
```

Выберите два правильных ответа.

- ☐ Всегда напечатается буква `B`.
- ☐ Буква `A` никогда не будет выведена после `End`.
- ☐ Буква `B` никогда не будет выведена после `End`.
- ☐ Программа может напечатать `B`, `End` и `A` в указанном порядке.
- ☐ Всегда напечатается буква `A`.

Какое утверждение справедливо? Выберите правильный ответ.

- ○ Предполагается, что если поток находится внутри синхронизированного метода, то никакие другие потоки не выполняют в это время никакие другие методы того же класса.
- ○ Синхронизированные методы могут непосредственно вызывать только другие синхронизированные методы.
- ○ Никакие два потока не могут одновременно выполнять синхронизированные методы одного и того же объекта.
- ○ Методы, объявленные как `synchronized`, не должны быть рекурсивными, поскольку блокировка объекта не разрешает новых вызовов методов.

Какое утверждение относительно следующей программы верно?

```
public class MyClass extends Thread {
    static Object lock1 = new Object();
    static Object lock2 = new Object();

    static volatile int i1, i2, j1, j2, k1, k2;

    public void run() { while (true) { doit(); check(); } }

    void doit() {
        synchronized(lock1) { i1++; }
        j1++;
```

```
            synchronized(lock2) { k1++; k2++; }
            j2++;
            synchronized(lock1) { i2++; }
        }

        void check() {
            if (i1 != i2) System.out.println("i");
            if (j1 != j2) System.out.println("j");
            if (k1 != k2) System.out.println("k");
        }

        public static void main(String[] args) {
            new MyClass().start();
            new MyClass().start();
        }
    }
```

Выберите правильный ответ.

- Программа не скомпилируется успешно.
- Нельзя сказать определенно, будут ли напечатаны во время выполнения буквы `i`, `j` и `k`.
- Определенно можно сказать, что ни одна из букв `i`, `j` и `k` не будет напечатана во время выполнения.
- Определенно можно сказать, что буква `k` никогда не будет напечатана во время выполнения.
- Определенно можно сказать, что буквы `i` и `k` никогда не будут напечатаны во время выполнения.

Какое из следующих событий послужит причиной смерти потока? Выберите правильный ответ.

- Заканчивается выполнение метода `run()`.
- Заканчивается выполнение конструктора потока.
- Вызывается метод `wait()`.
- Вызывается метод `sleep()`.
- Заканчивается выполнение метода `start()`.

Какое утверждение относительно следующего кода верно?

```
public class Joining {
    static Thread createThread(final int i, final Thread t1) {
        Thread t2 = new Thread() {
            public void run() {
                System.out.println(i + 1);
                try {
                    t1.join();
                } catch (InterruptedException e) {
                }
                System.out.println(i + 2);
            }
        };
        System.out.println(i + 3);
        t2.start();
        System.out.println(i + 4);
        return t2;
    }

    public static void main(String[] args) {
        createThread(10, createThread(20, Thread.currentThread()));
    }
}
```

Выберите два правильных ответа.

- ☐ Последнее число, которое будет напечатано, — это `12`.
- ☐ Число `24` будет напечатано прежде, чем число `21`.
- ☐ Первое число, которое будет напечатано, — это `13`.
- ☐ Число `14` будет напечатано прежде, чем число `22`.
- ☐ Число `11` будет напечатано прежде, чем число `23`.

Что можно гарантировать, когда вызывается метод `yield()`? Выберите правильный ответ.

- ○ Всем потокам с низшим приоритетом будет предоставлено процессорное время.
- ○ Текущий поток уснет на некоторое время, пока другие потоки работают.
- ○ Текущий поток не продолжится, пока другие потоки не завершатся.
- ○ Поток будет ждать, пока не получит уведомления.
- ○ Ничего из вышеперечисленного.

Где определен метод `notify()`? Выберите правильный ответ.

- ○ `Applet`.
- ○ `Thread`.
- ○ `Runnable`.
- ○ `Object`.

Как установить приоритет потока? Выберите правильный ответ.

- ○ Передавая приоритет в качестве параметра в конструктор потока.
- ○ Ни один из выше перечисленных.
- ○ Используя метод `setPriority()` класса `Thread`.
- ○ Оба из вышеперечисленных.

Какое утверждение верно касательно блокировки? Выберите два правильных ответа.

- ☐ Поток может обладать более чем одной блокировкой одновременно.
- ☐ Вызов `wait()` у объекта `Thread` освободит все блокировки, удерживаемые потоком.
- ☐ Вызов `notify()` у объекта, блокировка которого удерживается текущим потоком, освободит блокировку.
- ☐ Много потоков могут владеть одной и той же блокировкой в одно и то же время.
- ☐ Вызов `wait()` у объекта, блокировка которого удерживается текущим потоком, освободит блокировку.

Что получится в результате вызова метода `wait()` у объекта в отсутствие уверенности, что текущий поток владеет блокировкой объекта? Выберите правильный ответ.

- ○ Поток будет заблокирован, пока не получит блокировку объекта.
- ○ Будет выброшено `IllegalMonitorStateException`, если будет вызван метод `wait()`, в то время как текущий поток не владеет блокировкой на объект.
- ○ Код скомпилируется неуспешно.
- ○ Ничего особенного не произойдет.

Что из нижеперечисленного является правдоподобной причиной того, что поток может быть жив, но до сих пор не работает? Выберите четыре правильных ответа.

- ☐ Поток ожидает некоторого условия в результате вызова метода `wait()` .
- ☐ Выполнение достигло конца метода `run()` .
- ☐ Поток спит после вызова метода `sleep()` .
- ☐ Поток не имеет наивысшего приоритета и в текущий момент не выполняется.
- ☐ Поток ожидает получения блокировки объекта, чтобы выполнить определенный метод этого объекта.

Урок 10. Основные классы

Основные классы

Общая группа

Какой тип у возвращаемого значения метода `hashCode()` класса `Object`?
Выберите правильный ответ.

- `Class`
- `String`
- `long`
- `int`
- `Object`

Какое исключение может быть выброшено методом `clone()` класса `Object`?
Выберите правильный ответ.

- `NotCloneableException`
- `NoClonesAllowedException`
- `IllegalCloneException`
- `CloneNotSupportedException`

Что из нижеследующего есть класс-оболочка? Выберите три правильных ответа.

- ☐ `java.lang.Void`
- ☐ `java.lang.Long`
- ☐ `java.lang.String`
- ☐ `java.lang.Boolean`
- ☐ `java.lang.Int`

Какой класс из следующих ниже не расширяет класс `java.lang.Number` ?
Выберите два правильных ответа.

- ☐ `java.lang.Boolean`
- ☐ `java.lang.Character`
- ☐ `java.lang.Short`
- ☐ `java.lang.Float`
- ☐ `java.lang.Byte`

Какие из этих классов определяют постоянные (`immutable`) объекты? Выберите три правильных ответа.

- ☐ Short
- ☐ Byte
- ☐ Object
- ☐ Thread
- ☐ Character

Какой из этих классов имеет конструктор с одним параметром типа `String` .
Выберите два правильных ответа.

- ☐ `Void`
- ☐ `Object`
- ☐ `Character`
- ☐ `Boolean`
- ☐ `Integer`

Какой из классов-оболочек содержит метод `booleanValue()` ? Выберите правильный ответ.

- ○ Все классы-оболочки, расширяющие класс `Number` .
- ○ Все классы-оболочки.
- ○ Все классы-оболочки, которые также реализуют метод `compareTo()` .
- ○ Все классы-оболочки, за исключением `Void` .
- ○ Только класс `Boolean` .

Какое утверждение справедливо в отношении классов-оболочек? Выберите два правильных ответа.

- ☐ `Double` имеет метод `compareTo()` .
- ☐ `Character` имеет метод `intValue()` .
- ☐ `String` — это класс-оболочка.
- ☐ `Byte` расширяет `Number` .

Какая строка следующего кода выведет в точности `11` ? Выберите два правильных ответа.

```
class MyClass {
    public static void main(String[] args) {
```

```
        double v = 10.5;

        System.out.println(Math.ceil(v));         // (1)
        System.out.println(Math.round(v));        // (2)
        System.out.println(Math.floor(v));        // (3)
        System.out.println((int) Math.ceil(v));   // (4)
        System.out.println((int) Math.floor(v)); // (5)
    }
}
```

- ☐ Строка с маркером (1).
- ☐ Строка с маркером (2).
- ☐ Строка с маркером (3).
- ☐ Строка с маркером (4).
- ☐ Строка с маркером (5).

Какой из методов не определен в классе `Math` ? Выберите правильный ответ.

- ○ `double tan2(double)`
- ○ `double ceil(double)`
- ○ `float max(float, float)`
- ○ `double cos(double)`
- ○ `int abs(int a)`

Какой тип имеет возвращаемое значение в методе `round(float)` класса `Math` ? Выберите правильный ответ.

- `float`
- `double`
- `Float`
- `Integer`
- `int`

Какой тип имеет возвращаемое значение в методе `ceil(double)` класса `Math`? Выберите правильный ответ.

- `double`
- `Integer`
- `Double`
- `int`
- `float`

Что программа выведет на печать во время выполнения? Выберите правильный ответ.

```
public class Round {
    public static void main(String[] args) {
        System.out.println(Math.round(-0.5) + " " + Math.round(0.5));
    }
};
```

- `0 0`
- `-1 0`
- `0 1`
- Ничего из вышеперечисленного.
- `-1 1`

Какое утверждение относительно выражения `((int)(Math.random() * 4))` справедливо? Выберите три правильных ответа.

- ☐ Вероятность получения числа `1` или `2` равна.
- ☐ Может быть получено число `4`.
- ☐ Может быть получено число `0`.
- ☐ Может быть получено отрицательное значение.
- ☐ Может быть получено число `3`.

Какой из следующих операторов не может использоваться в сочетании с объектом `String`? Выберите два правильных ответа.

- ☐ `+=`
- ☐ `&`
- ☐ `+`
- ☐ `.`
- ☐ `-`

Какое из выражений извлечет подстроку `"kap"` из строки, задаваемой как `String str = "kakapo"`? Выберите правильный ответ.

- `str.substring(2, 3)`
- `str.substring(3, 3)`
- `str.substring(2, 2)`
- `str.substring(2, 5)`
- `str.substring(2, 4)`

Что получится, если попытаться скомпилировать и выполнить следующий код?

```
class MyClass {
    public static void main(String[] args) {
        String str1 = "str1";
        String str2 = "str2";
        String str3 = "str3";

        str1.concat(str2);
        System.out.println(str3.concat(str1));
    }
}
```

Выберите правильный ответ.

- Во время выполнения программа выведет `str3str1`.
- Во время выполнения программа выведет `str3`.
- Код не скомпилируется, поскольку результатом выражения `str3.concat(str1)` будет недопустимый аргумент для метода `println()`.
- Во время выполнения программа выведет `str3str1str2`.
- Во время выполнения программа выведет `str3str2`.

Какую функцию выполняет метод `trim()` класса `String`? Выберите правильный ответ.

- ○ Возвращает строку, в которой удалены из первоначальной строки и лидирующие, и замыкающие пробелы.
- ○ Ничто из вышеперечисленного.
- ○ Возвращает строку, в которой удалены из первоначальной строки все пробелы.
- ○ Возвращает строку, в которой удалены из первоначальной строки замыкающие пробелы.
- ○ Возвращает строку, в которой удалены из первоначальной строки лидирующие пробелы.

Какое утверждение верно? Выберите два правильных ответа.

- ☐ Объекты `String` неизменны.
- ☐ Подклассы класса `String` могут не быть неизменны.
- ☐ Все объекты имеют публичный метод `clone()` .
- ☐ Выражение `((new StringBuffer()) instanceof String)` всегда истинно.
- ☐ Все классы оболочки объявлены как `final` .

Какое из этих выражений недопустимо? Выберите правильный ответ.

- ○ `"co".concat("ol")`
- ○ `("co" + new String('o' + 'l'))`
- ○ `("co" + new String("co"))`
- ○ `("co" + "ol")`
- ○ `('c' + 'o' + 'o' + 'l')`

Что получится, если попытаться скомпилировать и выполнить следующий код?

```
public class RefEq {
    public static void main(String[] args) {
        String s = "ab" + "12";
        String t = "ab" + 12;
        String u = new String("ab12");
        System.out.println((s == t) + " " + (s == u));
    }
}
```

Выберите правильный ответ.

- Во время выполнения программа выведет `false false`.
- Код не скомпилируется.
- Во время выполнения программа выведет `true true`.
- Во время выполнения программа выведет `false true`.
- Во время выполнения программа выведет `true false`.

Для какого из списков параметров имеются соответствующие конструкторы в классе `String`? Выберите три правильных ответа.

- `(String str)`
- `()`
- `(char[] data)`
- `(int capacity)`

Какой метод не определен в классе `String`? Выберите правильный ответ.

- `trim()`
- `reverse()`
- `length()`
- `hashCode()`
- `concat(String)`

Какое утверждение относительно метода `charAt()` класса `String` истинно? Выберите правильный ответ.

- Метод `charAt()` получает в качестве аргумента значение типа `char`.
- Выражение `("abcdef").charAt(3)` допустимо.
- Метод `charAt()` возвращает объект `Character`.
- Индекс первого символа 1.
- Результатом вычисления `"abcdef".charAt(3)` будет символ `'d'`.

Какое выражение даст в результате `true`? Выберите правильный ответ.

- `("hello".concat("there")).equals("hello there")`
- `"Hello there".toLowerCase().equals("hello there")`
- `"hello: there!".equals("hello there")`
- `"Hello There".compareTo("hello there") == 0`
- `"HELLO THERE".equals("hello there")`

Что следующая программа выведет во время выполнения? Выберите правильный ответ.

```
public class Search {
```

```
    public static void main(String[] args) {
        String s = "Contentment!";
        int middle = s.length() / 2;
        String nt = s.substring(middle - 1, middle + 1);
        System.out.println(s.lastIndexOf(nt, middle));
    }
};
```

- 5
- 11
- 2
- 4
- 9
- 7

Что получится в результате попытки скомпилировать и выполнить следующую программу? Выберите правильный ответ.

```
public class MyClass {
    public static void main(String[] args) {
        String s = "hello";
        StringBuffer sb = new StringBuffer(s);
        sb.reverse();
        if (s == sb) System.out.println("a");
        if (s.equals(sb)) System.out.println("b");
        if (sb.equals(s)) System.out.println("c");
    }
}
```

- Код не будет скомпилирован, так как выражение `(s.equals(sb))` недопустимо.
- Во время выполнения программа выведет `c`.
- Код не будет скомпилирован, так как конструктор класса `String` вызывается неправильно.
- Во время выполнения будет выброшено исключение `ClassCastException`.
- Код не будет скомпилирован, так как выражение `(s == sb)` недопустимо.

Что получится в результате попытки скомпилировать и выполнить следующую программу?

```
public class MyClass {
    public static void main(String[] args) {
        StringBuffer sb = new StringBuffer("have a nice day");
        sb.setLength(6);
        System.out.println(sb);
    }
}
```

Выберите правильный ответ.

- Во время выполнения программа выведет `have a`.
- Во время выполнения программа выведет `ce day`.
- Код не будет скомпилирован, так как в классе `StringBuffer` не существует метода с именем `setLength`.
- Во время выполнения будет выброшено исключение `StringIndexOutOfBoundsException`.
- Во время выполнения программа выведет `have a nice day`.
- Код не будет скомпилирован, так как ссылка `sb` не является допустимым аргументом для метода `println()`.

Для какого из этих списков параметров существует соответствующий конструктор в классе `StringBuffer`? Выберите три правильных ответа.

- [ ] `(String str)`
- [ ] `(int capacity)`
- [ ] `()`
- [ ] `(char[] data)`

Урок 11. Коллекции и карты

Коллекции и карты

Общая группа

Какие интерфейсы являются основными в структуре коллекций? Выберите три правильных ответа.

- ☐ `Set`
- ☐ `LinkedList`
- ☐ `Map`
- ☐ `Bag`
- ☐ `Collection`

Какие реализации из нижеперечисленных представлены в пакете `java.util`? Выберите два правильных ответа.

- ☐ `HashMap`
- ☐ `TreeMap`
- ☐ `HashList`
- ☐ `ArrayMap`
- ☐ `ArraySet`

Какой интерфейс используются для представления коллекций, которые содержат упорядоченно не уникальные элементы? Выберите правильный ответ.

- `SortedSet`
- `List`
- `Set`
- `Collection`
- `Sequence`

Какое утверждение касательно коллекций справедливо? Выберите два правильных ответа.

- ☐ Некоторые действия в коллекции могут выбрасывать `UnsupportedOperationException`.
- ☐ Интерфейс `Collection` содержит метод с именем `get`.
- ☐ Методы, вызывающие необязательные операции коллекции, должны или перехватывать исключение `UnsupportedOperationException`, или объявлять его в своем выражении `throws`.
- ☐ В списке `List` могут содержаться элементы-дубликаты.
- ☐ `ArrayList` может содержать только неизменное количество элементов.

Что получится, если попробовать скомпилировать и выполнить следующую программу?

```
import java.util.*;

public class Sets {
    public static void main(String[] args) {
        HashSet set1 = new HashSet();
        addRange(set1, 1);
        ArrayList list1 = new ArrayList();
        addRange(list1, 2);
        TreeSet set2 = new TreeSet();
        addRange(set2, 3);
```

```
        LinkedList list2 = new LinkedList();
        addRange(list2, 5);

        set1.removeAll(list1);
        list1.addAll(set2);
        list2.addAll(list1);
        set1.removeAll(list2);

        System.out.println(set1);
    }

    static void addRange(Collection col, int step) {
        for (int i = step * 2; i <= 25; i += step)
            col.add(new Integer(i));
    }
}
```

Выберите правильный ответ.

- Программа скомпилируется без ошибок и при выполнении выведет некую последовательность чисел.
- Программа не скомпилируется, поскольку множеству `TreeSet`, обозначенному `set2`, не был предоставлен объект `Comparator` для сравнения элементов.
- Программа скомпилируется без ошибок, но во время выполнения будет выброшено исключение `UnsupportedOperationException`.
- Программа скомпилируется без ошибок и напечатает все простые числа меньше `25`.
- Программа не скомпилируется, поскольку выполняются операции, несовместимые с реализацией коллекции.

Какие из этих методов определены в интерфейсе `Collection`?

- ☐ `get(int index)`
- ☐ `iterator()`
- ☐ `add(Object o)`
- ☐ `retainAll(Collection c)`
- ☐ `indexOf(Object o)`

Какой вывод сформирует следующая программа?

```
import java.util.*;
public class Iterate {
    public static void main(String[] args) {
        List l = new ArrayList();
        l.add("A"); l.add("B"); l.add("C"); l.add("D"); l.add("E");
        ListIterator i = l.listIterator();
        i.next(); i.next(); i.next(); i.next();
        i.remove();
        i.previous(); i.previous();
        i.remove();
        System.out.println(l);
    };
};
```

Выберите правильный ответ.

- ○ Будет выведено `[A, B, C, D, E]`.
- ○ Будет выведено `[A, C, E]`.
- ○ Будет выведено `[B, C, E]`.
- ○ Будет выведено `[A, B, D]`.
- ○ Будет выведена информация о выбрасываемом исключении `NoSuchElementException`.
- ○ Будет выведено `[B, D, E]`.

Какой из этих методов из интерфейса `Collection` вернет значение `true`, если коллекция была изменена во время выполнения операции? Выберите два правильных ответа.

- ☐ `containsAll()`
- ☐ `clear()`
- ☐ `add()`
- ☐ `contains()`
- ☐ `retainAll()`

Какой из методов можно вызвать у объектов, реализующих интерфейс `Map`? Выберите два правильных ответа.

- ☐ `remove(Object o)`
- ☐ `toArray()`
- ☐ `addAll(Collection c)`
- ☐ `contains(Object o)`
- ☐ `values()`

Какое из утверждений справедливо касательно карт? Выберите два правильных ответа.

- ☐ Изменения, производимые в множестве-представлении, полученном с помощью метода `keySet()`, будут отражены в исходной карте.
- ☐ Все ключи в карте уникальны.
- ☐ Возвращаемый тип метода `values()` это `Set`.
- ☐ Все реализации `Map` поддерживают ключи упорядоченно.
- ☐ Интерфейс `Map` расширяет интерфейс `Collection`.

Какую последовательность символов выведет следующая программа?

```
import java.util.*;
public class Lists {
    public static void main(String[] args) {
        List list = new ArrayList();
        list.add("1");
        list.add("2");
        list.add(1, "3");
        List list2 = new LinkedList(list);
        list.addAll(list2);
        list2 = list.subList(2, 5);
        list2.clear();
        System.out.println(list);
    }
}
```

Выберите правильный ответ.

- ○ `[3, 1, 2]`
- ○ Ничто из вышеперечисленного.
- ○ `[3, 1, 1, 2]`
- ○ `[1, 3, 2, 1, 3, 2]`
- ○ `[1, 3, 3, 2]`
- ○ `[1, 3, 2]`

Какой из следующих классов имеет метод `comparator()` ? Выберите два правильных ответа.

- ☐ `HashMap`
- ☐ `TreeMap`
- ☐ `TreeSet`
- ☐ `ArrayList`
- ☐ `HashSet`

Какие прототипы методов определены в интерфейсе `java.util.Map.Entry` ? Выберите два правильных ответа.

- ☐ `Object getKey()`
- ☐ `Object getValue()`
- ☐ `Object setKey(Object value)`
- ☐ `void setValue(Object value)`
- ☐ `void remove()`

Задано, что объекты, обозначаемые параметрами, переопределяют методы `equals()` и `hashCode()` соответственно. Какое значение вероятно будет возвращено следующим кодом?

```
String func(Object x, Object y) {
    return (x == y) + " " + x.equals(y) + " " + (x.hashCode() ==
y.hashCode());
}
```

Выберите два правильных ответа.

- ☐ `true false true`
- ☐ `false false true`
- ☐ `false true true`
- ☐ `true false false`
- ☐ `false true false`

Вставьте код в метод `equalsImpl()`, для того чтобы предоставить правильную реализацию метода `equals()`.

```
public class Pair {
    int a, b;
    public Pair(int a, int b) {
        this.a = a;
        this.b = b;
    }

    public boolean equals(Object o) {
        return (this == o) || (o instanceof Pair) &&
equalsImpl((Pair) o);
    }

    private boolean equalsImpl(Pair o) {
        // ... Добавьте код здесь ...
    }
}
```

Выберите три правильных ответа.

- ☐ `return false;`
- ☐ `return a == o.a && b == o.b;`
- ☐ `return a == o.a || b == o.b;`
- ☐ `return a >= o.a;`
- ☐ `return a == o.a;`