

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего образования
«Московский государственный технический университет имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)



УТВЕРЖДАЮ
Первый проректор –
проректор по учебной работе
МГТУ им. Н.Э. Баумана
_____ Б.В. Падалкин
«11» мая 2024 г.

Дополнительное профессиональное образование

ДОПОЛНИТЕЛЬНАЯ ПРОФЕССИОНАЛЬНАЯ ПРОГРАММА
ПРОГРАММА ПОВЫШЕНИЯ КВАЛИФИКАЦИИ
«Программирование на Java для тестировщиков»

Регистрац. № 06.05-11/11

Москва, 2024

АВТОРЫ ПРОГРАММЫ:

Преподаватель



А.С. Банков

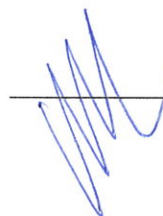
СОГЛАСОВАНО:

Начальник УСП



Т.А. Гузева

Директор
Центра дополнительного образования



М.В. Стоянова

#428 by K

Оглавление

1. ОБЩАЯ ХАРАКТЕРИСТИКА ДПП.....	4
1.1. Цель ДПП.....	4
1.2. Планируемые результаты обучения.....	4
1.3. Дополнительные характеристики ДПП.....	4
1.4. Перечень профессиональных компетенций в рамках имеющейся квалификации, качественное изменение которых осуществляется в результате обучения.....	4
1.5. Соответствие видов деятельности профессиональным компетенциям и их составляющих.....	5
2. УЧЕБНЫЙ ПЛАН ДПП.....	6
2.1. Категория слушателей ДПП.....	6
2.2. Общая трудоёмкость программы, аудиторная и самостоятельная работа.....	6
2.3. Форма обучения.....	6
2.4. Учебный план.....	6
3. КАЛЕНДАРНЫЙ УЧЕБНЫЙ ГРАФИК.....	6
4. РАБОЧАЯ ПРОГРАММА ДПП.....	8
5. УСЛОВИЯ РЕАЛИЗАЦИИ ДПП.....	15
5.1. Организационные условия реализации ДПП.....	15
5.2. Педагогические условия реализации ДПП.....	15
5.3. Учебно-методическое обеспечение ДПП.....	15
5.4. Методические рекомендации.....	15
6. ФОРМЫ ИТОГОВОЙ АТТЕСТАЦИИ ДПП.....	17
7. ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ИТОГОВОЙ АТТЕСТАЦИИ.....	18
7.1. Паспорт комплекта оценочных средств.....	18
7.2. Комплект оценочных средств.....	18

1. ОБЩАЯ ХАРАКТЕРИСТИКА ДПП

Программа подготовлена на основе:

- Федерального закона от 29 декабря 2012 года № 273-ФЗ «Об образовании в Российской Федерации»;
- требований Приказа Минобрнауки России от 01.07.2013 года № 499 «Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным профессиональным программам»;
- методических рекомендаций-разъяснений Минобрнауки России по разработке дополнительных профессиональных программ на основе профессиональных стандартов от 22 апреля 2015 года № ВК-1030/06.

Реализация программы ДПП направлена на получение новой компетенции, необходимой для профессиональной деятельности.

1.1. Цель ДПП

Сформировать у обучающихся знания, навыки и умения в области оценки качества разрабатываемого программного обеспечения (далее - ПО) путем проверки соответствия программного продукта заявленным требованиям.

1.2. Планируемые результаты обучения

Планируемые результаты обучения по ДПП:

- освоение профессиональных компетенций в процессе изучения перечисленных тем в учебном плане;
- успешное освоение программы повышения квалификации;
- успешное прохождение итоговой аттестации (зачет).

Обучающимся, успешно прошедшим обучение, выполнившим текущие контрольные задания и выдержавшим предусмотренное учебным планом зачет, выдается удостоверение о повышении квалификации по ДПП «Программирование на Java для тестировщиков».

1.3. Дополнительные характеристики ДПП

Характеристики новой квалификации определены в приказе Минтруда России от 02.08.2021 №531н «Об утверждении профессионального стандарта «Специалист по тестированию в области информационных технологий».

Вид профессиональной деятельности:

- Верификация и тестирование программного обеспечения (Код 06.004).

Трудовые функции:

- Верификация требований исходной документации на ПО (С/01.6).

1.4. Перечень профессиональных компетенций в рамках имеющейся квалификации, качественное изменение которых осуществляется в результате обучения

Получаемые компетенции базируются на основании Приказа Минобрнауки России от 19 сентября 2017 г. № 929 «Об утверждении федерального государственного образовательного стандарта высшего образования - бакалавриат по направлению подготовки 09.03.01 Информатика и вычислительная техника».

Перечень компетенций:

ОПК-8. Способен разрабатывать алгоритмы и программы, пригодные для практического применения.

1.5. Соответствие видов деятельности профессиональным компетенциям и их составляющих

Профессиональные компетенции	Практический опыт	Умения	Знания
Верификация требований исходной документации на ПО (С/01.6)			
ОПК-8. Способен разрабатывать алгоритмы и программы, пригодные для практического применения	Проведение анализа требований с точки зрения пригодности к тестированию	Анализировать взаимосвязи, выявлять пропущенную информацию	Теория тестирования ПО: модели тестирования ПО, планирование тестирования ПО, тест-дизайн, проектирование тестов

2. УЧЕБНЫЙ ПЛАН ДПП

2.1. Категория слушателей ДПП

Имеющаяся квалификация (требования к слушателям) – к освоению ДПП допускаются лица, имеющие среднее профессиональное и/или высшее образование.

2.2. Общая трудоёмкость программы, аудиторная и самостоятельная работа

Общая трудоёмкость программы 56 академических часов, из них 38 академических часов аудиторной работы, 16 академических часов самостоятельной работы и 2 академических часа итоговой аттестации.

2.3. Форма обучения

Форма обучения по ДПП – очная с применением дистанционных образовательных технологий.

2.4. Учебный план

ДПП «Программирование на Java для тестировщиков» реализуется одним модулем.

№ п/п	Наименование темы, модуля	Форма контроля	Всего, час	В том числе			
				Лекции	Практ. занятия	Самост. работа	Итоговая аттестация
1.	Вступление	Практ. задание	4	2	-	2	-
2.	Основы Java	Практ. задание	4	2	-	2	-
3.	Основы объектно-ориентированного программирования	Практ. задание	12	4	6	2	-
4.	Более сложные вопросы объектно-ориентированного программирования	Практ. задание	12	4	4	4	-
5.	Работа со стандартными библиотеками Java	Практ. задание	14	2	10	2	-
6.	Сложные вопросы программирования	Практ. задание	8	2	2	4	-
7.	Итоговая аттестация	Зачет	2	-	-	-	2
	ИТОГО	-	56	16	22	16	2

3. КАЛЕНДАРНЫЙ УЧЕБНЫЙ ГРАФИК

№ п/п	Наименование темы, модуля	1 день	2 день	3 день	4 день	5 день	6 день	7 день
1	Вступление							
2	Основы Java							
3	Основы объектно-ориентированного программирования							
4	Более сложные вопросы объектно-ориентированного программирования							
5	Работа со стандартными библиотеками Java							
6	Сложные вопросы программирования							
7	Итоговая аттестация							Зачет

Минимальный срок освоения ДПП – 7 дней.

4. РАБОЧАЯ ПРОГРАММА ДПП

4.1. Рабочая программа модуля «Программирование на Java для тестировщиков»

4.1.1. Цель изучения модуля: сформировать у обучающихся знания, навыки и умения в области оценки качества разрабатываемого программного обеспечения (далее - ПО) путем проверки соответствия программного продукта заявленным требованиям.

4.1.2. Задачи изучения модуля:

1. Знакомство с синтаксисом, семантикой и паттернами программирования на языке Java.
2. Получение практических навыков разработки простых программ на языке Java.

4.1.3. Планируемые результаты обучения

Процесс изучения раздела направлен на формирование следующих компетенций

Код компетенции	Перечень планируемых результатов обучения по модулю	Формы и методы обучения, способствующие формированию и развитию компетенции
ОПК-8	Знать: Теория тестирования ПО: модели тестирования ПО, планирование тестирования ПО, тест-дизайн, проектирование тестов. Уметь: Анализировать взаимосвязи, выявлять пропущенную информацию. Владеть: Проведение анализа требований с точки зрения пригодности к тестированию.	Формы обучения: Фронтальная. Методы обучения: Лекция; Практическая работа; Самостоятельная работа.

4.1.4 Содержание курса

Тема 1. Вступление (4 часа)

Лекции (2 часа). Введение. Литература. История и причины возникновения языков программирования. Типы языков программирования (компиляторы, интерпретаторы). Преимущества и недостатки языков assembler, basic, fortran, ada, pascal, algol, list, prolog, c/c++ и др. Причина появления Java. Существующие версии, установка JVM. Среды разработки, их установка. Точки входа в программу. Первая простейшая программа. Написание, сборка, запуск.

Самостоятельная работа (2 часа).

Наименование темы	Дидактические единицы, вынесенные на самостоятельное изучение	Формы самостоятельной работы	Учебно-методическое обеспечение	Форма контроля
Вступление	Программирование на языке Java	Проработка дополнительной литературы	Курбатова, И.В. Основы программирования на языке Java: учебное пособие для вузов / И.В. Курбатова, А.В. Печкуров. – Санкт-Петербург: Лань, 2024. – 348 с.	Практ. задание

Тема 2. Основы Java (4 часа)

Лекции (2 часа). Синтаксис Java. Примитивные типы. Типы с плавающей точкой. Логический тип. Литералы. Приведение типов. Переменные, объявление. Динамическая инициализация. Область видимости. Массивы. Одномерные массивы, многомерные массивы. Альтернативный способ объявления массивов. Выведение типов локальных переменных (var). Некоторые ограничения var. Строки, тип String (кратко). Приведение типов, автоматическое повышение типов. Операции. Арифметические операции. Деление по модулю. Инкремент, декремент. Побитовые операции. Логические операции. Тернарная операция. Присваивание. Приоритет операций, использование скобок. Операторы выбора if, switch. Циклы (while, do-while, for), вложенные циклы. Операторы перехода break, continue, return, goto. Почему не стоит использовать goto (Спагетти-код)?

Самостоятельная работа (2 часа).

Наименование темы	Дидактические единицы, вынесенные на самостоятельное изучение	Формы самостоятельной работы	Учебно-методическое обеспечение	Форма контроля
Основы Java	Переменные, примитивы, операции, циклы	Проработка дополнительной литературы	Курбатова, И.В. Основы программирования на языке Java: учебное пособие для вузов / И.В. Курбатова, А.В. Печкуров. – Санкт-Петербург: Лань, 2024. – 348 с.	Практ. задание

Тема 3. Основы объектно-ориентированного программирования (12 часов)

Лекции (4 часа). Объектно-ориентированный подход. Наследование. Инкапсуляция. Полиморфизм. Классы в Java. Объявление классов. Перегрузка конструкторов, параметризованные конструкторы. Принцип первичного конструктора. Перегрузка методов. Наследование. Доступ к членам класса при наследовании. Модификаторы доступа public, default, private, protected. Класс Object. Методы, предоставляемые классом. Хэш, сравнение по значению и по ссылке. Классы для примитивов. Перечисления. Перечисления являются классами. Базовый класс Enum. Классы-оболочки примитивов.

Практические занятия (6 часов). Создание объектов, оператор new. Поля, объявление. Методы, объявление. Ключевое слово this. Конструкторы. Последовательность создания объекта в памяти. Деструкторы. Почему их нет. Ключевое слово finalize. Сборщик мусора. Виды ссылок. Статические методы, переменные. Ключевое слово static, final. Отличие super от this. Ключевое слово super. Абстрактные классы, интерфейсы. Финальные классы. Реализация абстрактных классов, интерфейсов. Автоупаковка, автораспаковка. Предотвращение ошибок, предостережение. Аннотации. Объявление, политика хранения. Получение аннотаций, рефлексия. Встроенные аннотации.

Самостоятельная работа (2 часа).

Наименование темы	Дидактические единицы, вынесенные на самостоятельное изучение	Формы самостоятельной работы	Учебно-методическое обеспечение	Форма контроля
Основы объектно-ориентированного программирования	Объектно-ориентированное программирование	Проработка дополнительной литературы	Мейер Б. Объектно-ориентированное программирование и программная инженерия: учебное пособие / Мейер Б. – Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2024.	Практ. задание

Тема 4. Более сложные вопросы объектно-ориентированного программирования (12 часов)

Лекции (4 часа). Пакеты. Импорт пакетов. Соглашение об именовании классов, методов, переменных. Snake notation, camel notation. Магические константы. Польская нотация. Множественное наследование, проблемы множественного наследования. Интерфейсы. Отличие от абстрактных классов. Ошибки в программах и их обработка. Исключения. Типы исключения, иерархия исключений. Строки. Класс String. Иммутабельность. Сравнение строк. Модификация. Класс StringBuilder. Класс StringBuffer. Обобщения. Простые обобщения. Общая форма обобщенного класса. Обобщенные интерфейсы. Ссылки. Куда дели указатели? Связь с Objective C, Delphi. Сборщик мусора и типы ссылок (4 типа ссылок).

Практические занятия (4 часа). Создание своих пакетов. JAR-файлы. Переменные в интерфейсах. Расширение интерфейсов. Реализация по умолчанию. Закрытые (приватные) методы. Принцип Барбары Лисков. Перехват исключений try-catch-finally. Множественный перехват. Неперехваченные исключения. Генерация исключений throw. Конструкция throws. Встроенные исключения. Создание собственных исключений. Сцепленные исключения (цепочки исключений). Наложение ограничений на обобщения сверху и снизу. Ограничения обобщений.

Самостоятельная работа (4 часа).

Наименование темы	Дидактические единицы, вынесенные на самостоятельное изучение	Формы самостоятельной работы	Учебно-методическое обеспечение	Форма контроля
Более сложные вопросы объектно-ориентированного программирования	Классы и наследование	Проработка дополнительной литературы	Мейер Б. Объектно-ориентированное программирование и программная инженерия: учебное пособие / Мейер Б. – Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2024.	Практ. задание

Тема 5. Работа со стандартными библиотеками Java (14 часов)

Лекции (2 часа). Коллекции. Интерфейс Collection, List, Set, SortedSet, NavigableSet, Queue, Deque. Массивы, связанные списки. Классы коллекций: ArrayList, LinkedList. Наборы Set: HashSet, LinkedHashSet, TreeSet, EnumSet. Очереди. Векторы. Стек.

Практические занятия (10 часов). Обход коллекций в цикле, итератор, Spliterator. Устаревшие коллекции и итераторы: Enumeration, Dictionary, Hashtable, Properties. Класс Map и его наследники. Порядок хранения данных в Map. Коллизии в HashMap. Компараторы. Проблема сборщика мусора при работе с Map. Еще раз о типах ссылок. Лямбда-выражения. Функциональные интерфейсы. Примеры. Обобщенные функциональные интерфейсы. Ссылки на методы, конструкторы. Предопределенные функциональные интерфейсы. Ввод-вывод. Классы, интерфейсы. Класс File. Поток ввода-вывода. Буферизация. Интерфейсы AutoClosable, Closable. Обработка исключений при вводе-выводе. Принцип Барбары Лисков при работе с потоками. Регулярные выражения. Синтаксис, применение. Сайты-хелперы. Серезализация.

Самостоятельная работа (2 часа).

Наименование темы	Дидактические единицы, вынесенные на самостоятельное изучение	Формы самостоятельной работы	Учебно-методическое обеспечение	Форма контроля
Работа со стандартными библиотеками Java	Библиотеки Java	Проработка дополнительной литературы	Курбатова, И.В. Основы программирования на языке Java: учебное пособие для вузов / И.В. Курбатова, А.В. Печкуров. – Санкт-Петербург: Лань, 2024. – 348 с.	Практ. задание

Тема 6. Сложные вопросы программирования (8 часов)

Лекции (2 часа). Введение в паттерны программирования. Зачем нужны паттерны? Синглтоны. Ленивая инициализация. Слушатель/издатель. Машина состояний. Сервис-локаторы. Антипаттерны.

Практические занятия (2 часа). Многопоточное программирование. Многозадачность, виды. Поток. Преимущества многопоточности перед многозадачностью. Создание потоков. Приоритеты потоков. Состояние гонки, преодоление. Взаимная блокировка. Синхронизация потоков. Синхронизированные методы. Атомарные операции. Атомарные

примитивы. Главный поток. Взаимодействие между потоками. Приостановка, возобновление, остановка потоков. Ключевое слово `volatile`. Сопрограммы. Корутины.

Самостоятельная работа (4 часа).

Наименование темы	Дидактические единицы, вынесенные на самостоятельное изучение	Формы самостоятельной работы	Учебно-методическое обеспечение	Форма контроля
Сложные вопросы программирования	Многопоточное программирование	Проработка дополнительной литературы	Курбатова, И.В. Основы программирования на языке Java: учебное пособие для вузов / И.В. Курбатова, А.В. Печкуров. – Санкт-Петербург: Лань, 2024. – 348 с.	Практ. задание

4.1.5. Оценочное средство для текущего контроля (формулировка практических заданий):

Тема 1. Формулировка практического задания:

Скачать и установить JVM, среду разработки. Повторить простейшую программу из лекции. Собрать, запустить.

Тема 2. Формулировка практического задания:

Объявить переменные примитивов. Попробовать вычисления, посмотреть приоритет операций. Сделать циклы всех типов. Например, цикл, ожидающий нажатия клавиши.

Тема 3. Формулировка практического задания:

Создать интерфейс, сделать к нему реализацию. Перекрыть методы `equals`, `hashCode` у класса `Object`, сделать свое сравнение классов.

Тема 4. Формулировка практического задания:

Сделать два класса и наследовать от них один класс наследник. Увидеть ошибку, исправить. Собрать на основе `StringBuilder` строку "Hello world" по словам. С помощью обобщений позволить работать с данной строкой классу, наследованному из двух интерфейсов. Собрать свою `jar`-библиотеку. На основе библиотеки создать программу, упаковать ее в `jar`-файл и запустить.

Тема 5. Формулировка практического задания:

Создать свой класс с данными. Создать коллекцию с данным классом. Отсортировать коллекцию. Добиться коллизии, посмотреть, что в этом случае вернет метод `get` у `HashMap`.

Тема 6. Формулировка практического задания:

Создать синглтон с счетчиком. Сделать несколько процессов, которые будут случайно инкрементировать или декрментировать счетчик. Дать поработать счетчику, вывести его значение. Параллельно протоколировать все обращения к счетчику. Какой поток, в какой момент времени, какую операцию совершил. Сверить, что бы конечное состояние счетчика соответствовало тому, что делали потоки в процессе работы.

5. УСЛОВИЯ РЕАЛИЗАЦИИ ДПП

5.1. Организационные условия реализации ДПП

Наименование аудитории	Вид занятия	Наименование оборудования, программного обеспечения
Компьютерный класс/вебинар	Лекции	ПК с ОС GNU/Linux, IntelliJIDEA
Компьютерный класс/вебинар	Практические занятия	ПК с ОС GNU/Linux, IntelliJIDEA
Компьютерный класс/вебинар	Самостоятельная работа	ПК с ОС GNU/Linux, IntelliJIDEA
Компьютерный класс/вебинар	Итоговая аттестация	ПК с ОС GNU/Linux, IntelliJIDEA

5.2. Педагогические условия реализации ДПП

Реализация программы обеспечивается преподавательским составом, удовлетворяющим следующим условиям:

- наличие высшего профессионального образования, соответствующее профилю программы, из числа штатных преподавателей, или привлеченных на условиях почасовой оплаты труда;
- значительный опыт практической деятельности в соответствующей сфере из числа штатных преподавателей или привлеченных на условиях почасовой оплаты труда

5.3. Учебно-методическое обеспечение ДПП

Основная литература:

1. Курбатова, И.В. Основы программирования на языке Java: учебное пособие для вузов / И.В. Курбатова, А.В. Печкуров. – Санкт-Петербург: Лань, 2024. – 348 с.
2. Мейер Б. Объектно-ориентированное программирование и программная инженерия: учебное пособие / Мейер Б. – Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2024.

Дополнительная литература:

1. Герберт Шилдт. Java полное руководство 12-е издание. ISBN 978-5-907458-86-4.
2. Э. Гамма, Р. Хельм, Р. Джонсон, Дж. Влссидес. Паттерны объектно-ориентированного проектирования. ISBN 978-5-4461-1595-2/
3. С. А. Чернышев. Принципы, паттерны и методологии разработки программного обеспечения. ISBN 978-5-534-14383-6.

5.4. Методические рекомендации

ДПП построена по тематическому принципу, каждый раздел представляет собой логически завершенный материал.

Преподавание программы основано на личностно-ориентированной технологии образования, сочетающей два равноправных аспекта этого процесса: обучение и учение. Личностно-ориентированный подход развивается при участии слушателей в активной работе на практических занятиях. Личностно-ориентированный подход направлен, в первую очередь, на развитие индивидуальных способностей обучающихся, создание условий для развития творческой активности слушателя и разработке инновационных идей, а также на развитие самостоятельности мышления при решении учебных задач разными способами, нахождение рационального варианта решения, сравнения и оценки нескольких вариантов их решения и т.п. Это способствует формированию приемов умственной деятельности по восприятию новой информации, ее запоминанию и осознанию, созданию образов для сложных понятий и процессов, приобретению навыков поиска решений в условиях неопределенности.

Практические занятия проводятся для приобретения навыков решения практических задач в предметной области модуля. Задания, выполняемые на практических занятиях, выполняются с использованием активных и интерактивных методов обучения.

Самостоятельная работа слушателей предназначена для проработки дополнительной литературы. Результаты практических заданий слушателей учитываются на итоговой аттестации.

При изучении курса предусмотрены следующие методы организации и осуществления учебно-познавательной деятельности:

- объяснительно-иллюстративный метод;
- репродуктивный метод;
- частично-поисковый метод.

6. ФОРМЫ ИТОГОВОЙ АТТЕСТАЦИИ ДПП

Итоговая аттестация проводится в форме зачета для проверки сформированности компетенций, полученных в рамках ДПП.

Зачет проводится в формате ответов на вопросы билета. Результатом зачета служат правильные ответы на вопросы билета.

По результатам итоговой аттестации обучающемуся выставляется оценка «ЗАЧТЕНО/НЕ ЗАЧТЕНО»:

Оценка «ЗАЧТЕНО» выставляется обучающемуся, который:

- правильно ответил на 3 из 5 вопросов билета;
- продемонстрировал необходимые систематизированные знания и достаточную степень владения принципами предметной области программы, понимание их особенностей и взаимосвязь между ними в течение всего срока обучения по ДПП.

Оценка «НЕ ЗАЧТЕНО» ставится обучающемуся, который:

- неправильно ответил на 3 и более вопросов билета;
- имеет крайне слабые теоретические и практические знания, обнаруживает неспособность к построению самостоятельных заключений.

7. ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ИТОГОВОЙ АТТЕСТАЦИИ

7.1. Паспорт комплекта оценочных средств

Предметы оценивания	Объекты оценивания	Показатели оценки
ОПК-8. Способен разрабатывать алгоритмы и программы, пригодные для практического применения	Ответы на вопросы	Правильные ответы на вопросы теста

7.2. Комплект оценочных средств

7.2.1. Темы для подготовки к зачету:

1. Типы языков программирования.
2. Синтаксис Java.
3. Переменные.
4. Массивы.
5. Операции.
6. Объектно-ориентированный подход.
7. Конструкторы.
8. Наследование.
9. Ссылки.
10. Коллекции.

7.2.2. Вопросы для проведения зачета:

1. Назовите примитивные типы Java. Для целочисленных типов назовите их разрядность (сколько байт/бит отводится на хранение значения).
2. Является ли строка простым типом? Если да, то почему?
3. Чем отличается автоупаковка от автоматического повышения типов?
4. Как объявить массив, многомерный массив?
5. Назовите поддерживаемые арифметические операции.
6. Расскажите о приоритете арифметических операций.
7. Что такое тернарный оператор?
8. Опишите все виды циклов. Можно ли с помощью for заменить цикл while?
9. Что такое класс, чем он отличается от объекта?
10. Как объявить класс, члены класса, методы?
11. От какого класса наследуются все классы?
12. Назовите методы класса Object и что они делают, где применяются.

13. Что такое перегрузка методов, конструкторов?
14. В чем заключается принцип первичного конструктора?
15. Что такое статические методы и переменные, как они объявляются?
16. Что означают ключевые слова `this` и `super`?
17. Классы в Java по умолчанию являются открытыми или финализированными?
18. Что такое финализированный класс, как его объявить?
19. Аннотации. Для чего нужны, как объявить?
20. Как в Java реализуется множественное наследование?
21. Что такое абстрактный класс? Как создать экземпляр абстрактного класса?
22. Что такое интерфейс? Можно ли в интерфейсе объявить переменную?
23. Исключения. Какие основные два вида исключений в Java?
24. Что такое обобщения (дженерики)?
25. Коллекции. Какие коллекции реализованы в Java?
26. Что такое интератор, как его использовать?
27. Когда удобнее использовать `LinkedList`, а когда `ArrayList`?
28. Функциональные интерфейсы, лямбда выражения. Что это такое, как объявить?
29. Регулярные выражения.
30. Паттерны программирования. Приведите примеры.
31. Паттерн `singleton`, `lazy`, `observer`, `service locator`, `state machine`.
32. Антипаттерны. Какие знаете, расскажите.
33. Поток ввода-вывода. Что такое интерфейс `Closable`?
34. Какие правила именований переменных, методов и функций Вы знаете?
35. Какие виды многозадачности Вы знаете?
36. Какие классы реализуют многопоточную работу?
37. Что такое атомарная операция?
38. Что такое синхронизация? Как делается синхронизация по монитору?