

## ПРИКАЗ № Java/2025

об утверждении дополнительной профессиональной программы

01.09.2025 г., г. Санкт-Петербург

В соответствии с Федеральным законом № 273-ФЗ от 29.12.2012 «Об образовании в Российской Федерации» (ст. 12, 75, 101), Приказом Министерства образования и науки РФ от 1 июля 2013 г. № 499 «Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным профессиональным программам» приказываю:

1. Утвердить дополнительную профессиональную программу – программу профессиональной переподготовки «Профессия: Java-разработчик» общей трудоемкостью 272,7 академ. ч.
2. Контроль исполнения приказа оставляю за собой.



/ Зинин Е.Ю.

Утверждено  
Приказом № Java/2024 от 01.09.2025 г.  
Генеральный директор  
Зинин Е.Ю.  
«01» сентября 2025 г.



**ДОПОЛНИТЕЛЬНАЯ ПРОФЕССИОНАЛЬНАЯ ПРОГРАММА  
– ПРОГРАММА ПРОФЕССИОНАЛЬНОЙ ПЕРЕПОДГОТОВКИ  
«ПРОФЕССИЯ: JAVA-РАЗРАБОТЧИК»**

**Срок реализации:** 9 месяцев  
**Количество часов:** 272,9 акад. ч.  
**Форма обучения:** заочная форма  
**Формат обучения:** с применением  
исключительно дистанционных технологий  
**Возраст обучающихся:** для лиц старше 18  
лет, имеющих или получающих среднее  
профессиональное и (или) высшее  
образование  
**Автор:** Микрюкова Е.Р.

г. Санкт-Петербург  
2025

## 1. ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Настоящая дополнительная профессиональная программа – программа профессиональной переподготовки «Профессия: Java-разработчик» (далее – Программа) разработана в соответствии с:

- Федеральным законом от 29 декабря 2012 г. № 273-ФЗ «Об образовании в Российской Федерации»;
- Приказом Министерства образования и науки РФ от 1 июля 2013 г. № 499 «Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным профессиональным программам»;
- Постановлением Правительства РФ от 11.10.2023 № 1678 «Об утверждении Правил применения организациями, осуществляющими образовательную деятельность, электронного обучения, дистанционных образовательных технологий при реализации образовательных программ»;
- Постановлением Главного государственного санитарного врача РФ от 28 сентября 2020 г. № 28 «Об утверждении санитарных правил СП 2.4.3648-20 "Санитарно-эпидемиологические требования к организациям воспитания и обучения, отдыха и оздоровления детей и молодежи"»;
- Профессиональным стандартом 06.001 «Программист», утвержден приказом Министерства труда и социальной защиты Российской Федерации от 20 июля 2022 г. № 424н;
- ФГОС высшего образования - магистратура по направлению подготовки 09.04.01 Информатика и вычислительная техника, утв. приказом Минобрнауки России от 19.09.2017 г. № 918.

В данной программе учтены основные идеи формирования универсальных учебных действий учащихся и соблюдена преемственность с программами высшего и/или среднего профессионального образования.

### **Направленность программы:**

Программа имеет техническую направленность.

### **Адресат:**

Программа рассчитана на обучающихся старше 18 лет, имеющих или получающих среднее профессиональное и (или) высшее образование.

### **Актуальность реализации:**

Актуальность программы профессиональной переподготовки «Профессия: Java-разработчик» обусловлена устойчивым спросом на квалифицированных разработчиков на языке Java как в России, так и за рубежом. Java остаётся одним из самых востребованных языков программирования в корпоративной разработке, особенно в сфере финансов, телекоммуникаций и государственного сектора. Развитие цифровой экономики требует постоянного расширения штата ИТ-специалистов, способных проектировать, разрабатывать и сопровождать надёжные программные решения. Компании нуждаются в специалистах, владеющих современными фреймворками (Spring, Hibernate), методами тестирования и принципами построения микросервисной архитектуры. Программа позволяет освоить прикладную разработку на Java с нуля или систематизировать и углубить уже имеющиеся знания. Обучение построено на реальных кейсах и ориентировано на требования рынка труда. Полученные навыки позволяют выпускникам успешно трудоустроиться или переходить на более высокие должности в сфере ИТ.

**Отличительные особенности программы:**

Программа ориентирована на практическое освоение Java-разработки с нуля до уровня junior/middle-специалиста. Обучение построено на реальных кейсах и задачах, максимально приближенных к требованиям индустрии. Упор сделан на освоение востребованных технологий: Java Core, Spring, Hibernate, SQL, REST API, работа с Git и Docker. Все занятия проходят в дистанционном формате на образовательной онлайн-платформе с доступом к материалам после окончания обучения.

**Объем и срок освоения программы:** 272,9 академ. ч. в течение 9 месяцев.

Доступ к материалам Программы обучающимся предоставляется на 3 года. Это позволяет повторять изученный материал в удобное время, восполнять пробелы в знаниях, а также возвращаться к практическим заданиям при решении рабочих задач. Такой формат способствует более глубокому закреплению навыков и поддерживает профессиональное развитие выпускников даже после завершения обучения.

**Выдаваемый документ о квалификации:**

Обучающиеся успешно прошедшие Программу и сдавшие итоговую аттестацию вправе получить диплом о профессиональной переподготовке установленного ООО «Тривиум» образца.

**Цели и задачи программы:**

Цель дополнительной профессиональной программы «Профессия: Java-разработчик» – сформировать у слушателей практические компетенции в области разработки программного обеспечения на языке Java с использованием современных технологий и инструментов.

**Программа направлена на решение следующих основных задач:**

Обучающие:

- Изучение синтаксиса и базовых конструкций языка Java.
- Освоение принципов объектно-ориентированного программирования и архитектуры приложений.
- Получение навыков работы с базами данных и реализации REST API.
- Освоение инструментов и технологий для сборки, тестирования и деплоя Java-приложений.

Развивающие:

- Формирование алгоритмического и системного мышления.
- Развитие навыков поиска и самостоятельного освоения технической информации.
- Совершенствование аналитических способностей при решении прикладных задач.
- Развитие навыков командной работы через выполнение проектных заданий.
- Повышение цифровой грамотности и профессиональной мобильности.

Воспитательные:

- Формирование ответственности за качество и результат своей работы.
- Воспитание устойчивого интереса к профессиональному развитию в сфере ИТ.
- Развитие целеустремленности и настойчивости в достижении поставленных целей.

**Планируемые результаты:**

Планируемыми результатами обучения по программе является приобретение следующих знаний, умений, навыков, участвующих в качественном изменении компетенций:

**Знания:**

- Основные парадигмы и принципы программирования, включая объектно-ориентированное программирование.
- Архитектура и жизненный цикл Java-приложений.
- Принципы построения клиент-серверных и многослойных приложений.
- Основы проектирования и моделирования программных систем.
- Основы реляционных баз данных и принципы нормализации.
- Теоретические основы тестирования программного обеспечения.
- Подходы к организации командной разработки и сопровождению программных продуктов.

**Умения:**

- Разрабатывать и отлаживать Java-приложения с использованием ООП.
- Создавать и взаимодействовать с базами данных через JDBC и ORM.
- Реализовывать REST API и обеспечивать интеграцию компонентов системы.
- Применять фреймворки Spring и Hibernate для создания корпоративных приложений.
- Использовать системы контроля версий и инструменты автоматизации (Git, Maven, Docker).

**Навыки:**

- Работа в среде разработки (IDE) и с системой контроля версий Git.
- Построение клиент-серверных приложений на Java.
- Использование шаблонов проектирования при создании программных решений.
- Настройка и использование фреймворков Spring Boot и Hibernate.
- Разработка и тестирование REST-сервисов.
- Работа с базами данных: написание SQL-запросов, миграции, подключение к БД.
- Подготовка, сборка и развёртывание приложений с использованием Maven и Docker.

**Перечень профессиональных компетенций, на получение которых направлено обучение:**

На основе профстандарта «Программист»:

Обобщенные трудовые функции			Трудовые функции		
код	наименование	уровень квалификации	наименование	код	уровень (подуровень) квалификации

А	Разработка и отладка программного кода	3	Формализация и алгоритмизация поставленных задач для разработки программного кода	A/01.3	3
			Написание программного кода с использованием языков программирования, определения и манипулирования данными в базах данных	A/02.3	3
			Оформление программного кода в соответствии с установленными требованиями	A/03.3	3
			Работа с системой управления версиями программного кода	A/04.3	3
			Проверка и отладка программного кода	A/05.3	3
В	Проверка работоспособности и рефакторинг кода программного обеспечения	4	Разработка процедур проверки работоспособности и измерения характеристик компьютерного программного обеспечения	B/01.4	4
			Проверка работоспособности компьютерного программного обеспечения	B/03.4	4
			Рефакторинг, оптимизация и инспекция программного кода	B/04.4	4
			Исправление дефектов программного кода,	B/05.4	4

			зафиксированных в базе данных дефектов		
			Осуществление сборки однородных программных модулей в программный проект	В/06.4	4
С	Интеграция программных модулей и компонентов и проверка работоспособности и выпусков программного продукта	5	Разработка процедур интеграции программных модулей	С/01.5	5

Таким образом, в результате освоения программы у обучающихся формируются следующие профессиональные компетенции:

- Способность формализовывать прикладные задачи и разрабатывать алгоритмы их решения.
- Умение писать, оформлять и отлаживать программный код на языке Java в соответствии с техническим заданием.
- Владение принципами объектно-ориентированного программирования и архитектуры программных систем.
- Навыки взаимодействия с базами данных с использованием SQL и ORM.
- Умение реализовывать REST API и обеспечивать клиент-серверное взаимодействие.
- Владение инструментами контроля версий, сборки и деплоя программных продуктов.
- Умение применять методы тестирования и обеспечивать стабильность программного обеспечения.
- Способность к оптимизации существующего кода.

**Организационно-педагогические условия реализации программы дополнительного профессионального образования:**

**Язык реализации образовательной программы:** обучение проводится на русском языке.

**Форма обучения:** заочная форма.

**Особенности реализации программы:** программа реализуется с использованием электронного обучения и исключительно дистанционных образовательных технологий.

**Условия набора:** на обучение принимаются все желающие лица, оплатившие обучение и заключившие договор об образовании. Обучение проходит в индивидуальном формате без

формирования учебных групп. Обучающийся самостоятельно определяет время освоения Программы.

#### **Формы проведения занятий:**

- занятия в видео-формате;
- занятия в текстовом формате;
- практическая работа;
- самостоятельная работа;
- индивидуальные вопросы.

#### **Материально-техническое оснащение:**

##### **Материальное обеспечение программы**

Занятия проводятся на образовательной онлайн-платформе «Productstar». Каждый обучающийся и педагог оснащены доступом к образовательной онлайн-платформе: <https://platform.productstar.ru/login>.

У педагога дополнительного профессионального образования имеется необходимое оборудование средства для реализации программы: ноутбук с подключением к интернету, программное обеспечение.

##### **Методическое обеспечение программы**

Программа обеспечена:

- учебно-методическими материалами;
- практическими заданиями и тренажерами;
- теоретическими и практическими видео-занятиями.

##### **Кадровое обеспечение:**

К реализации программы в качестве педагогов дополнительного образования допускаются лица:

1) отвечающее одному из требований:

а) имеющее высшее образование или среднее профессиональное образование в рамках укрупненных групп специальностей и направлений подготовки высшего образования и специальностей среднего профессионального образования «Образование и педагогические науки»;

б) имеющее высшее образование либо среднее профессиональное образование в рамках иных укрупненных групп специальностей и направлений подготовки высшего образования и специальностей среднего профессионального образования при условии его соответствия дополнительной общеобразовательной общеразвивающей программе, реализуемой ООО «Тривиум», и получение при необходимости дополнительного профессионального образования педагогической направленности;

в) успешно прошедшее промежуточной аттестации не менее чем за два года обучения по образовательным программам высшего образования по специальностям и направлениям подготовки, соответствующей направленности программы дополнительного профессионального образования;

2) не имеющее ограничений на занятие педагогической деятельностью, установленных законодательством Российской Федерации;

3) прошедшее обязательный предварительный (при поступлении на работу) и периодические медицинские осмотры (обследования), а также внеочередные медицинские

осмотры (обследования) в порядке, установленном законодательством Российской Федерации.

Реализация Программы также возможна лицами, привлекаемыми на условиях гражданско-правового договора в соответствии с действующим законодательством РФ.

## 2. УЧЕБНЫЙ ПЛАН

№ п/п	Наименование блока	Количество часов			Формы контроля / аттестация
		Всего	Теория	Практика	
1.	Основы Java. Git	29,1	12,7	16,4	Текущий контроль, промежуточная аттестация
2.	Java. Основные концепции	19,7	9	10,7	Текущий контроль
3.	Java Core. Коллекции	26,2	9,2	17	Текущий контроль, промежуточная аттестация
4.	Java Core. Файлы и сеть. Сборка	27,4	13,7	13,7	Текущий контроль, промежуточная аттестация
5.	Базы данных. SQL. PostgreSQL	23,7	7,8	15,9	Текущий контроль
6.	Объектно- ориентированное программирование в Java	19,3	6,2	13,1	Текущий контроль, промежуточная аттестация
7.	Основы тестирования на Java	32,9	8,3	24,6	Текущий контроль, промежуточная аттестация
8.	Java Spring Framework	34,6	13,2	21,4	Текущий контроль, промежуточная аттестация
9.	Основы Backend на Java	30,6	9,1	21,5	Текущий контроль, промежуточная аттестация
10.	Инструменты DevOps	14,4	6,3	8,1	Текущий контроль, промежуточная аттестация
11.	Дипломная работа	15	3,2	11,8	Итоговая аттестация
	<b>Итого:</b>	272,9	98,7	174,2	

### 3. КАЛЕНДАРНЫЙ УЧЕБНЫЙ ГРАФИК

№ п/п	Наименование блоков	1 месяц	2 месяц	3 месяц	4 месяц	5 месяц	6 месяц	7 месяц	8 месяц	9 месяц
1.	Основы Java. Git	X								
2.	Java. Основные концепции		X							
3.	Java Core. Коллекции			X						
4.	Java Core. Файлы и сеть. Сборка				X					
5.	Базы данных. SQL. Postgres					X				
6.	Объектно-ориентированное программирование в Java					X				
7.	Основы тестирования на Java						X			
8.	Java Spring Framework							X		
9.	Основы Backend на Java								X	
10	Инструменты DevOps									X
11.	Дипломная работа									X

Календарный учебный график устанавливает предельный срок освоения Программы — 9 месяцев. Вместе с тем, в связи с заочной формой и дистанционным форматом обучения, слушатели могут проходить Программу в индивидуальном темпе и завершить обучение раньше установленного срока.

## 4. РАБОЧАЯ ПРОГРАММА

### Блок 1. Основы Java. Git

Теория 12,7 академ. ч. Практика 16,4 академ. ч.

**В рамках блока ставятся следующие задачи:**

- Ознакомить слушателей с историей и особенностями языка Java, инструментами разработки и процессом запуска программ.
- Сформировать представление о базовом синтаксисе Java, переменных, типах данных, операторах и циклах.
- Научить использовать структуры данных (массивы, списки, словари, деревья) при решении практических задач.
- Дать практические навыки работы с системой контроля версий Git и платформой GitHub.
- Организовать выполнение практических заданий и мини-проекта для закрепления изученного материала.

**Планируемые результаты обучения:**

По завершении блока слушатели:

- Знают синтаксис Java и базовые конструкции языка.
- Умеют создавать и запускать простые программы в среде IntelliJ IDEA.
- Владеют навыками работы с массивами и структурами данных.
- Умеют использовать Git для управления версиями кода, работать с ветками, коммитами и GitHub.
- Способны выполнить небольшой учебный проект (кредитный калькулятор), применяя знания по Java и Git.

Блок состоит из следующих тем:

Тема 1: Знакомство с языком Java и инструментами

Тема 2: Базовый синтаксис Java

Тема 3: Переменные и типы данных в Java

Тема 4: Практическая работа № 1: основы Java. Git

Тема 5: Циклы и языковые конструкции в Java

Тема 6: Практическая работа № 2: основы Java. Git

Тема 7: Структуры данных

Тема 8: Практическая работа № 3: основы Java. Git

Тема 9: Системы контроля версий

Тема 10: Git и GitHub

Тема 11: Продвинутая работа с Git

Тема 12: Проект: Разработка кредитного калькулятора

Форма контроля: текущий контроль, промежуточная аттестация

Текущий контроль осуществляется через выполнение практических работ в рамках тем. Содержание и критерии практических заданий указываются в личном кабинете на образовательной онлайн-платформе. Работы загружаются в личный кабинет и проверяются педагогами.

Оценивание осуществляется по системе:

- Зачёт — работа выполнена полностью, все основные критерии соблюдены, решение корректно и соответствует заданию.
- На доработку — работа в целом выполнена, но содержит отдельные ошибки или недочёты, которые могут быть исправлены; допускается повторное представление после корректировок.
- Незачёт — задание выполнено с критическими ошибками, не соответствует требованиям или не демонстрирует освоение материала. После двух попыток доработки педагог вправе выставить итоговую оценку «незачёт».

Промежуточная аттестация проводится по итоговому проекту блока — разработке кредитного калькулятора. Оценивается правильность реализации, качество кода и работа с Git. Решение загружается в личный кабинет, проверяется педагогами. Оценка выставляется в том же формате: «зачёт», «на доработку», «незачёт».

## **Блок 2. Java. Основные концепции**

Теория 9 академ. ч. Практика 10,7 академ. ч.

### **В рамках блока ставятся следующие задачи:**

- Научить создавать и использовать классы, объекты, поля и методы, включая модификаторы доступа и инкапсуляцию.
- Сформировать навыки обработки ошибок и исключений в приложениях.
- Научить работать с массивами, строками и регулярными выражениями для решения прикладных задач.
- Развить умение применять изученные концепции на практике через выполнение практических заданий и мини-проектов.

### **Планируемые результаты обучения:**

По завершении блока слушатели:

- Умеют разрабатывать собственные классы, конструкторы, методы и поля, использовать инкапсуляцию и модификаторы доступа.
- Осваивают работу с исключениями, умеют отлавливать, обрабатывать и генерировать ошибки.
- Владеют навыками работы с массивами и строками, включая использование регулярных выражений.
- Способны реализовать практические задачи и мини-проекты, применяя полученные знания в реальных сценариях.

Блок состоит из следующих тем:

Тема 1: Классы и объекты: начальный уровень

Тема 2: Классы и объекты: продвинутый уровень

Тема 3: Практическая работа № 1: Java. Основные концепции

Тема 4: Методы и поля классов

Тема 5: Практическая работа № 2: Java. Основные концепции

Тема 6: Обработка исключений

Тема 7: Работа с массивами

Тема 8: Строки и регулярные выражения

Тема 9: Практическая работа № 3: Java. Основные концепции

Форма контроля: текущий контроль

Текущий контроль осуществляется через выполнение практических работ в рамках тем. Содержание и критерии заданий указываются в личном кабинете на образовательной платформе. Работы загружаются в личный кабинет и проверяются педагогами.

Оценивание осуществляется по системе:

- Зачёт — работа выполнена полностью, все основные критерии соблюдены, решение корректно и соответствует заданию.
- На доработку — работа в целом выполнена, но содержит отдельные ошибки или недочёты, которые могут быть исправлены; допускается повторное представление после корректировок.
- Незачёт — задание выполнено с критическими ошибками, не соответствует требованиям или не демонстрирует освоение материала. После двух попыток доработки педагог вправе выставить итоговую оценку «незачёт».

### **Блок 3. Java Core. Коллекции**

Теория 9,2 академ. ч. Практика 17 академ. ч.

**В рамках блока ставятся следующие задачи:**

- Ознакомить слушателей с понятием алгоритмов, их сложностью и применением в Java-разработке.
- Научить использовать Java Collections Framework: основные виды коллекций, их структуру, возможности и ограничения.
- Развить навыки выбора оптимальной коллекции для решения конкретной задачи и работы с динамическими структурами данных.
- Освоить работу с современными инструментами обработки данных: Stream API, лямбда-выражениями и optional-объектами.
- Закрепить изученный материал через выполнение практических работ и разработку консольного приложения для управления данными.

**Планируемые результаты обучения:**

По завершении блока слушатели:

- Знают основные алгоритмы сортировки и поиска, понимают оценку сложности алгоритмов.
- Понимают устройство и принципы работы Java Collections Framework, владеют методами работы с основными коллекциями.
- Умеют применять различные типы коллекций (ArrayList, LinkedList, HashMap, HashSet, TreeMap и др.) в практических задачах.
- Владеют навыками обработки данных с использованием Stream API, лямбда-выражений и optional-объектов.
- Способны разработать консольное приложение для управления данными, включая добавление, обновление, удаление и анализ информации.

Блок состоит из следующих тем:

Тема 1: Знакомство с алгоритмами

Тема 2: Обзор коллекций в Java

Тема 3: Практическая работа № 1. Java Core. Коллекции

Тема 4: Коллекции: ArrayList и LinkedList  
Тема 5: Коллекции: HashMap, HashSet, LinkedHashMap, LinkedHashSet  
Тема 6: Коллекции: TreeMap, TreeSet  
Тема 7: Практическая работа № 2. Java Core. Коллекции  
Тема 8: Stream API, lambda, optional  
Тема 9: Проект. Консольное приложение для управления данными  
Тема 10: Generic-типы. Runtime, wildcards и PECS

Форма контроля: текущий контроль, промежуточная аттестация

Текущий контроль осуществляется через выполнение практических работ в рамках тем. Содержание и критерии практических заданий указываются в личном кабинете на образовательной платформе. Работы загружаются в личный кабинет и проверяются педагогами.

Оценивание осуществляется по системе:

- Зачёт — работа выполнена полностью, все основные критерии соблюдены, решение корректно и соответствует заданию.
- На доработку — работа в целом выполнена, но содержит отдельные ошибки или недочёты, которые могут быть исправлены; допускается повторное представление после корректировок.
- Незачёт — задание выполнено с критическими ошибками, не соответствует требованиям или не демонстрирует освоение материала. После двух попыток доработки педагог вправе выставить итоговую оценку «незачёт».

Промежуточная аттестация проводится по итоговому проекту блока — разработке консольного приложения для управления данными. Оценивается корректность реализации функционала, структура кода и применение изученных инструментов. Решение загружается в личный кабинет, проверяется педагогами и оценивается по той же системе: «зачёт», «на доработку», «незачёт».

#### **Блок 4. Java Core. Файлы и сеть. Сборка**

Теория 13,7 академ. ч. Практика 13,7 академ. ч.

**В рамках блока ставятся следующие задачи:**

- Сформировать понимание модели памяти JVM, принципов управления памятью и работы сборщика мусора.
- Научить сериализовать и десериализовать объекты, работать с различными форматами данных (JSON, XML, YAML).
- Ознакомить слушателей с ключевыми шаблонами проектирования и научить применять их в практике.
- Развить навыки работы с сетевыми протоколами, архитектурой OSI, TCP/IP, UDP и HTTP.
- Дать практические навыки работы с библиотеками IO и NIO, включая передачу данных и работу с потоками ввода-вывода.
- Освоить основы многопоточного программирования в Java, работу с синхронизацией, блокировками и асинхронными операциями.
- Научить использовать рефлексию и создавать собственные аннотации.
- Ознакомить с современными системами автоматической сборки Java-проектов и применением Maven, Gradle и Bazel.

## Планируемые результаты обучения:

По завершении блока слушатели:

- Понимают работу памяти JVM, сборщика мусора и механизмов оптимизации приложений.
- Умеют выполнять сериализацию и десериализацию объектов в различных форматах данных.
- Освоили основные шаблоны проектирования и применяют их для решения практических задач.
- Знают архитектуру сетей и протоколов, умеют реализовывать клиент-серверное взаимодействие с использованием Java.
- Владеют навыками работы с библиотеками IO и NIO для эффективной обработки данных.
- Могут разрабатывать многопоточные приложения, использовать синхронизацию и оптимизировать работу потоков.
- Умеют применять механизмы рефлексии для анализа и изменения структуры объектов, создавать собственные аннотации.
- Способны собирать Java-проекты с использованием современных инструментов автоматизации.

Блок состоит из следующих тем:

Тема 1: Работа с памятью JVM

Тема 2: Сериализация

Тема 3: Шаблоны проектирования

Тема 4: Практическая работа № 1. Java Core. Файлы и сеть. Сборка

Тема 5: Работа с сетью. OSI, TCP/IP, UDP, HTTP

Тема 6: Практическая работа № 2. Java Core. Файлы и сеть. Сборка

Тема 7: Java библиотека NIO. Отличия IO и NIO

Тема 8: Многопоточное программирование в Java

Тема 9: Практическая работа № 3. Java Core. Файлы и сеть. Сборка

Тема 10: Рефлексия в Java. Создание аннотаций

Тема 11: Системы автоматической сборки Java-проектов

Форма контроля: текущий контроль, промежуточная аттестация

Текущий контроль осуществляется через выполнение практических работ в рамках тем. Содержание и критерии практических заданий указываются в личном кабинете на образовательной платформе. Работы загружаются в личный кабинет и проверяются педагогами.

Оценивание осуществляется по системе:

- Зачёт — работа выполнена полностью, все основные критерии соблюдены, решение корректно и соответствует заданию.
- На доработку — работа в целом выполнена, но содержит отдельные ошибки или недочёты, которые могут быть исправлены; допускается повторное представление после корректировок.
- Незачёт — задание выполнено с критическими ошибками, не соответствует требованиям или не демонстрирует освоение материала. После двух попыток доработки педагог вправе выставить итоговую оценку «незачёт».

Промежуточная аттестация проводится по итоговому проекту блока, в котором слушатели демонстрируют умение работать с памятью, файлами, сетью, потоками и системами сборки. Оценивается корректность реализации функционала, структура кода и применение изученных инструментов. Решение загружается в личный кабинет, проверяется педагогами и оценивается по той же системе: «зачёт» / «на доработку» / «незачёт».

### **Блок 5. Базы данных. SQL. PostgreSQL**

Теория 7,8 академ. ч. Практика 15,9 академ. ч.

#### **В рамках блока ставятся следующие задачи:**

- Ознакомить слушателей с основами баз данных, их структурой, принципами хранения и обработки данных.
- Научить работать с СУБД PostgreSQL: создавать базы данных, таблицы, индексы, настраивать связи и выполнять CRUD-операции.
- Дать знания по работе с SQL-запросами: выборка, фильтрация, сортировка, группировка, агрегатные функции.
- Сформировать навыки работы с JDBC и интеграции приложений Java с базами данных.
- Отработать использование транзакций, уровней изоляции и принципов обеспечения целостности данных.
- Развить практические умения по работе с реальными проектами на PostgreSQL и Docker.

#### **Планируемые результаты обучения:**

По завершении блока слушатели:

- Понимают основные принципы реляционных баз данных и умеют проектировать структуры хранения данных.
- Владеют базовыми и продвинутыми возможностями SQL, включая фильтрацию, группировку и агрегатные функции.
- Умеют подключать Java-приложения к базам данных с помощью JDBC и выполнять CRUD-операции.
- Освоили работу с PostgreSQL, умеют настраивать схемы, индексы и выполнять оптимизацию запросов.
- Разбираются в транзакциях, уровнях изоляции и механизмах обеспечения целостности данных.
- Могут разрабатывать учебные проекты с применением СУБД PostgreSQL и Docker.

Блок состоит из следующих тем:

Тема 1: Введение в базы данных

Тема 2: Практическая работа № 1. Базы данных. SQL. PostgreSQL

Тема 3: Базы данных. JDBC

Тема 4: Развитие проекта

Тема 5: Практическая работа № 2. Базы данных. SQL. PostgreSQL

Тема 6: SQL-запросы

Тема 7: Практическая работа № 3. Базы данных. SQL. PostgreSQL

Тема 8: Воркшоп: реальный Postgres

Тема 9: Практическая работа № 4. Базы данных. SQL. PostgreSQL

Тема 10: Транзакции и уровни изоляции

Форма контроля: текущий контроль

Текущий контроль осуществляется через выполнение практических работ в рамках тем. Содержание и критерии практических заданий указываются в личном кабинете на образовательной платформе. Работы загружаются в личный кабинет и проверяются педагогами.

Оценивание осуществляется по системе:

- Зачёт — работа выполнена полностью, все основные критерии соблюдены, решение корректно и соответствует заданию.
- На доработку — работа в целом выполнена, но содержит отдельные ошибки или недочёты, которые могут быть исправлены; допускается повторное представление после корректировок.
- Незачёт — задание выполнено с критическими ошибками, не соответствует требованиям или не демонстрирует освоение материала. После двух попыток доработки педагог вправе выставить итоговую оценку «незачёт».

### **Блок 6. Объектно-ориентированное программирование в Java**

Теория 6,2 академ. ч. Практика 13,1 академ. ч.

**В рамках блока ставятся следующие задачи:**

- Познакомить слушателей с методологиями разработки ПО и ключевыми принципами объектно-ориентированного программирования (ООП).
- Научить применять основные концепции ООП: инкапсуляцию, наследование, полиморфизм и абстракцию при создании Java-приложений.
- Дать понимание принципов проектирования SOLID и их влияния на качество архитектуры.
- Освоить практические приёмы упрощения кода с использованием принципов YAGNI, DRY и KISS.
- Сформировать представление о свойствах хорошего кода, включая абстракцию, композицию, связанность (Coupling) и когезию (Cohesion).
- Научить применять полученные знания на практике при создании мини-приложений.

**Планируемые результаты обучения:**

По завершении блока слушатели:

- Знают методологии разработки и концепции ООП.
- Умеют разрабатывать собственные Java-приложения, используя инкапсуляцию, наследование, интерфейсы и полиморфизм.
- Понимают и применяют на практике принципы SOLID для создания гибкой и расширяемой архитектуры.
- Умеют анализировать и улучшать качество кода, применяя подходы YAGNI, DRY и KISS.
- Владеть навыками проектирования качественного кода с учётом абстракции, композиции, уровня связанности и когезии.
- Умеют создавать небольшие приложения.

Блок состоит из следующих тем:

Тема 1: ООП в Java

Тема 2: Принципы SOLID

Тема 3: Практическая работа № 1: объектно-ориентированное программирование в Java

Тема 4: Упрощаем код: YAGNI, DRY, KISS в Java-проектах

Тема 5: Проект от компании «Совкомбанк Технологии»: Telegram-бот для расчета графиков погашения кредитов

Форма контроля: текущий контроль, промежуточная аттестация

Текущий контроль осуществляется через выполнение практических работ в рамках тем блока. Содержание и критерии практических заданий доступны в личном кабинете на образовательной платформе. Работы загружаются в личный кабинет и проверяются педагогами.

Оценивание осуществляется по системе:

- Зачёт — работа выполнена полностью, все основные критерии соблюдены, решение корректно и соответствует заданию.
- На доработку — работа в целом выполнена, но содержит отдельные ошибки или недочёты, которые могут быть исправлены; допускается повторное представление после корректировок.
- Незачёт — задание выполнено с критическими ошибками, не соответствует требованиям или не демонстрирует освоение материала. После двух попыток доработки педагог вправе выставить итоговую оценку «незачёт».

Промежуточная аттестация проводится по итоговому проекту блока — разработке Telegram-бота на Java на основе кейса от «Совкомбанк Технологии». Оценивается правильность реализации функциональности бота, применение принципов ООП и SOLID, качество кода и архитектуры, работа с Git. Решение загружается в личный кабинет, проверяется педагогами и оценивается по системе: «зачёт», «на доработку», «незачёт».

## **Блок 7. Основы тестирования на Java**

Теория 8,3 академ. ч. Практика 24,6 академ. ч.

**В рамках блока ставятся следующие задачи:**

- Ознакомить слушателей с основными подходами к тестированию программного обеспечения.
- Научить применять инструменты для юнит-, интеграционного и нагрузочного тестирования Java-приложений.
- Освоить написание модульных тестов с использованием JUnit и TestNG.
- Познакомить с работой фреймворков для мокинга (Mockito) и тестирования REST API (REST Assured).
- Научить использовать инструмент JMeter для нагрузочного тестирования.
- Дать практические навыки применения принципов BDD (поведенческого тестирования) и написания тестовых сценариев с помощью Cucumber-JVM, JBehave и других инструментов.
- Сформировать навыки работы с комплексом инструментов тестировщика для автоматизации тестирования в Java.

**Планируемые результаты обучения:**

По завершении блока слушатели:

- Знают основные подходы к тестированию и умеют определять оптимальные виды тестов.
- Умеют писать модульные тесты на Java с использованием JUnit и TestNG.
- Владеют навыками работы с моками и стабами, используя библиотеку Mockito.
- Умеют тестировать REST API через REST Assured и выполнять нагрузочное тестирование с помощью JMeter.
- Осваивают подход BDD, умеют писать тестовые сценарии на языке Gherkin и запускать их в среде Cucumber и JBehave.
- Владеют современными инструментами тестировщика: Mockito, JMeter, REST Assured, Selenium, Allure и др.
- Способны проектировать, автоматизировать и анализировать комплексные тестовые сценарии.

Блок состоит из следующих тем:

Тема 1: Введение в тестирование

Тема 2: Тестирование исходящих взаимодействий: outcoming. Мокинг

Тема 3: Среда тестирования Mockito

Тема 4: Практическая работа № 1: основы тестирования на Java

Тема 5: Основы юнит-тестирования

Тема 6: Тест-раннеры юнит-тестирования для Java: JUnit, TestNG

Тема 7: Практическая работа № 2: основы тестирования на Java

Тема 8: Основы интеграционного тестирования

Тема 9: Нагрузочное тестирование: JMeter

Тема 10: Практическая работа № 3: основы тестирования на Java

Тема 11: Инструмент тестирования: REST Assured

Тема 12: Практическая работа № 4: основы тестирования на Java

Тема 13: Основы BDD

Тема 14: Инструменты BDD для Java: Cucumber-JVM, Cukes-rest, JBehave

Тема 15: Инструменты тестировщика Java

Форма контроля: текущий контроль, промежуточная аттестация

Текущий контроль осуществляется через выполнение практических работ в рамках тем блока. Содержание и критерии заданий указываются в личном кабинете на образовательной платформе. Работы загружаются в личный кабинет и проверяются педагогами.

Оценивание осуществляется по системе:

- Зачёт — работа выполнена полностью, все основные критерии соблюдены, решение корректно и соответствует заданию.
- На доработку — работа в целом выполнена, но содержит отдельные ошибки или недочёты, которые могут быть исправлены; допускается повторное представление после корректировок.
- Незачёт — задание выполнено с критическими ошибками, не соответствует требованиям или не демонстрирует освоение материала. После двух попыток доработки педагог вправе выставить итоговую оценку «незачёт».

Промежуточная аттестация проводится по итоговому проекту блока — разработке BDD-сценария для тестирования Java-приложений с использованием Cucumber. Оценивается корректность написания BDD-сценариев на языке Gherkin; умение создавать параметризованные тесты и работать с таблицами данных; корректный запуск тестов через TestRunner; генерация и оформление отчётов о тестировании (Allure); использование

дополнительных библиотек, включая как минимум одну Assert-библиотеку. Решение загружается в личный кабинет, проверяется педагогами и оценивается по системе: «зачёт», «на доработку», «незачёт».

## **Блок 8. Java Spring Framework**

Теория 13,2 академ. ч. Практика 21,4 академ. ч.

### **В рамках блока ставятся следующие задачи:**

- Ознакомить слушателей с архитектурой Spring Framework и его ключевыми модулями.
- Научить использовать принципы Inversion of Control (IoC) и Dependency Injection (DI) для построения гибких приложений.
- Освоить работу со Spring Boot для быстрой разработки и конфигурации проектов.
- Научить проектировать и реализовывать REST API с помощью Spring Web и контроллеров.
- Дать практические навыки работы с базами данных через Spring JDBC, Spring Data и Hibernate.
- Обучить созданию CRUD-приложений и интеграции с внешними сервисами.
- Познакомить с принципами логирования и ведения журналов в веб-приложениях.
- Подготовить к выполнению итогового проекта — разработке банковского сервиса на Spring Boot в рамках реального кейса компании Aston.

### **Планируемые результаты обучения**

По завершении блока слушатели:

- Знают архитектуру Spring Framework, его возможности и основные модули.
- Умеют разрабатывать приложения на базе Spring Boot, создавать REST API и использовать контроллеры.
- Осваивают работу с базами данных через Spring JDBC, умеют строить стандартные и нестандартные SQL-запросы.
- Владеют навыками применения Hibernate и Spring Data для автоматизации взаимодействия с БД.
- Умеют создавать CRUD-приложения, использовать шаблоны проектирования и архитектуру MVC.
- Осваивают настройку логирования, ведение журналов и подключение сторонних зависимостей.
- Готовы выполнять проекты промышленного уровня, включая интеграцию бэкенда, работу с транзакциями и оптимизацию производительности.

Блок состоит из следующих тем:

Тема 1: Inversion of control. Dependency Injection

Тема 2: Что такое Spring framework и зачем он нужен?

Тема 3: Spring Application Context – от XML до Groovy

Тема 4: Spring Web. REST-архитектура и контроллеры

Тема 5: Spring JDBC. Работа с базой данных и построение запросов

Тема 6: Spring JDBC. Нестандартные запросы и транзакции

Тема 7: Hibernate. ORM – нет времени писать запросы

Тема 8: Spring Data: тот же Hibernate, только лучше!

Тема 9: Практическая работа № 1: Java Spring Framework

Тема 10: Servlets  
Тема 11: Архитектура веб-сервиса, MVC, Spring Boot  
Тема 12: Ведение журналов  
Тема 13: Основы HTML. Язык разметки страницы  
Тема 14: CRUD-приложение. Create, Read, Update, Delete application  
Тема 15: Практическая работа № 2: Java Spring Framework  
Тема 16: Проект от компании «Aston»: RestaurantHub API

Форма контроля: текущий контроль, промежуточная аттестация

Текущий контроль осуществляется через выполнение практических работ, самостоятельных заданий и мини-проектов в рамках тем блока. Содержание и критерии заданий указываются в личном кабинете на образовательной платформе. Работы загружаются в личный кабинет и проверяются педагогами.

Система оценивания:

- Зачёт — работа выполнена полностью, все основные критерии соблюдены, решение корректно и соответствует заданию.
- На доработку — работа в целом выполнена, но содержит отдельные ошибки или недочёты, которые могут быть исправлены; допускается повторное представление после корректировок.
- Незачёт — задание выполнено с критическими ошибками, не соответствует требованиям или не демонстрирует освоение материала. После двух попыток доработки педагог вправе выставить итоговую оценку «незачёт».

Промежуточная аттестация проводится по итоговому проекту блока — разработке банковского сервиса на Spring Boot в рамках кейса компании Aston. Оценивается корректность проектирования архитектуры и применение принципов MVC; реализация REST API и корректность работы контроллеров; работа с базами данных через Spring JDBC, Spring Data и Hibernate; организация CRUD-функционала, настройка транзакций и работа с ошибками; оформление и структура кода, читаемость и поддерживаемость проекта; интеграция логирования и подключение сторонних зависимостей. Решение загружается в личный кабинет, проверяется педагогами и оценивается по системе: «зачёт», «на доработку», «незачёт».

## **Блок 9. Основы Backend на Java**

Теория 9,1 академ. ч. Практика 21,5 академ. ч.

**В рамках блока ставятся следующие задачи:**

- Ознакомить слушателей с принципами клиент-серверного взаимодействия и современными архитектурными подходами.
- Научить реализовывать регистрацию, аутентификацию и авторизацию пользователей с использованием Spring Security.
- Дать практические навыки работы с REST API, gRPC, Protocol Buffers, JSON-RPC и XML-RPC.
- Освоить инструменты тестирования API (Postman, cURL) и отработать навыки интеграции Java-приложений с внешними сервисами.
- Познакомить с принципами CI/CD и настройкой непрерывной интеграции и доставки.
- Изучить расширенные возможности многопоточности в Java и их применение в backend-разработке.

- Научить проектировать масштабируемые приложения и использовать современные подходы: load balancing, API Gateway, reverse proxy, container orchestration.
- Подготовить к выполнению итогового проекта — разработке веб-приложения для мониторинга финансовых показателей компании «Globus IT».

### Планируемые результаты обучения

По завершении блока слушатели:

- Знают принципы работы клиент-серверной архитектуры и умеют анализировать взаимодействие браузера с сервером.
- Умеют реализовывать регистрацию, аутентификацию и авторизацию с помощью Spring Security.
- Владеют навыками работы с REST API, gRPC, Protocol Buffers, JSON-RPC, XML-RPC и инструментами тестирования API.
- Осваивают интеграцию Java-приложений с внешними сервисами.
- Умеют настраивать CI/CD для Java-проектов и автоматизировать развертывание приложений.
- Понимают принципы многопоточности в Java и могут использовать их для оптимизации производительности backend-систем.
- Осваивают проектирование масштабируемых приложений и работу с инструментами: load balancer, API Gateway, reverse proxy, container orchestrator.
- Готовы разрабатывать сложные backend-приложения промышленного уровня и защищать итоговой проект.

Блок состоит из следующих тем:

Тема 1: Основы клиент-серверного взаимодействия

Тема 2: Регистрация, аутентификация и авторизация

Тема 3: Практическая работа № 1: основы Backend на Java

Тема 4: Java интеграция с backend

Тема 5: Проект от компании «Globus IT»: веб-приложение для мониторинга финансовых показателей

Тема 6: CI/CD

Тема 7: Многопоточность advanced

Тема 8: Практическая работа № 2: основы Backend на Java

Тема 9: Масштабируемость

Тема 10: Искусственный интеллект для разработчика

Форма контроля: текущий контроль, промежуточная аттестация

Текущий контроль осуществляется через выполнение практических работ в рамках тем блока. Содержание и критерии заданий указываются в личном кабинете на образовательной платформе. Работы загружаются в личный кабинет и проверяются педагогами.

Оценивание проводится по системе:

- Зачёт — работа выполнена полностью, все основные критерии соблюдены, решение корректно.
- На доработку — работа в целом верная, но содержит ошибки или недочёты; допускается повторная отправка.

- Незачёт — работа не соответствует требованиям, не демонстрирует освоение материала; после двух попыток доработки педагог может выставить итоговую оценку «незачёт».

Промежуточная аттестация проводится по итоговому проекту — разработке веб-приложения для мониторинга финансовых показателей. Оценивается корректность реализации клиент-серверного взаимодействия; умение строить REST API и интегрировать gRPC, JSON-RPC, XML-RPC; применение Spring Security для аутентификации и авторизации; настройка CI/CD и оптимизация многопоточной обработки; проектирование масштабируемого приложения и использование соответствующих инструментов. Проект загружается в личный кабинет, проверяется педагогами и оценивается по системе: «зачёт», «на доработку», «незачёт».

## **Блок 10. Инструменты DevOps**

Теория 6,3 академ. ч. Практика 8,1 академ. ч.

### **В рамках блока ставятся следующие задачи:**

- Ознакомить слушателей с основными концепциями работы в среде Linux и освоить базовые команды для работы в консоли.
- Научить принципам контейнеризации приложений и работе с Docker.
- Дать практические навыки создания и упаковки Docker-образов, автоматизации сборки и управления контейнерами.
- Познакомить с системой Ansible и ее основными возможностями для автоматизации развертывания приложений.
- Научить создавать роли и переменные Ansible для управления инфраструктурой.
- Освоить разработку собственных ролей Ansible и применение их на практике.

### **Планируемые результаты обучения:**

По завершении блока слушатели:

- Знают основные концепции Linux, принципы контейнеризации, работу с Docker и возможности Ansible.
- Умеют работать в консоли Linux, создавать Dockerfile, собирать образы и управлять контейнерами, автоматизировать развертывание приложений с помощью Ansible, создавать и использовать роли, переменные и факты.
- Владеют инструментами DevOps: Linux, Docker, Ansible, GitLab CI.

Блок состоит из следующих тем:

Тема 1: Linux: введение и основные концепты ОС

Тема 2: Контейнеризация. Знакомство с Docker

Тема 3: Упаковка приложений в Docker

Тема 4: Практическая работа № 1: инструменты DevOps

Тема 5: Ansible. Основные концепции и команды

Тема 6: Ansible. Роли и переменные

Тема 7: Ansible. Разработка ролей

Форма контроля: текущий контроль, промежуточная аттестация

Текущий контроль осуществляется через выполнение практических работ в рамках тем блока. Содержание и критерии заданий указываются в личном кабинете на образовательной платформе. Работы загружаются в личный кабинет и проверяются педагогами.

Оценивание осуществляется по системе:

- Зачёт — работа выполнена полностью, все основные критерии соблюдены, решение корректно и соответствует заданию.
- На доработку — работа в целом выполнена, но содержит отдельные ошибки или недочёты, которые могут быть исправлены; допускается повторное представление после корректировок.
- Незачёт — задание выполнено с критическими ошибками, не соответствует требованиям или не демонстрирует освоение материала. После двух попыток доработки педагог вправе выставить итоговую оценку «незачёт».

Промежуточная аттестация проводится по итоговому проекту блока — созданию и упаковке Docker-образа приложения и разработке Ansible-плейбука для автоматизации развертывания. Оценивается корректность работы с Docker, умение создавать образы и управлять контейнерами, а также умение разрабатывать роли и переменные Ansible. Решение загружается в личный кабинет, проверяется педагогами и оценивается по системе: «зачёт», «на доработку», «незачёт».

### **Блок 11. Дипломная работа**

Теория 3,2 академ. ч. Практика 11,8 академ. ч.

**В рамках блока ставятся следующие задачи:**

- Ознакомить слушателей с инструментами и методами эффективной работы в команде.
- Сформировать навыки организации взаимодействия в проектной группе.
- Подготовить к выполнению итоговой дипломной работы, направленной на комплексное применение знаний и навыков, полученных в ходе программы.
- Содействовать развитию управленческих компетенций и лидерских качеств для потенциального роста до роли тимлида.
- Научить презентовать результаты работы и защищать проект.

**Планируемые результаты обучения:**

По завершении блока слушатели:

- Знают современные подходы к командной работе, принципы эффективного взаимодействия и распределения ролей в проекте.
- Умеют применять инструменты коллективной разработки и вести коммуникацию в команде.
- Владеют навыками подготовки и реализации проектных решений в условиях командного взаимодействия.
- Готовы к самостоятельной защите итогового проекта и демонстрации практических навыков, полученных в ходе обучения.

Блок состоит из следующих тем:

Тема 1: Инструменты работы в команде

Тема 2: Рост тимлида  
Тема 3: Дипломная работа

Итоговая аттестация проводится в формате защиты дипломной работы. Финальное задание направлено на комплексную проверку освоения всех ключевых тем программы.

Оценивается:

- корректность реализации проекта;
- умение применять полученные знания на практике;
- качество кода и архитектурных решений;
- оформление документации;
- защита и презентация проекта.

Решение направляется на проверку через личный кабинет. Оценивание осуществляется по системе:

- Зачёт — все критерии выполнены, приложение работает стабильно, структура проекта оформлена корректно.
- Доработка — задание содержит несущественные ошибки или неточности, возможно повторное представление после исправлений.
- Незачёт — критические нарушения, задание не демонстрирует достаточный уровень освоения программы. После двух попыток доработки педагог вправе выставить итоговую оценку «незачёт».

В случае успешной сдачи итоговой аттестации обучающийся получает диплом о профессиональной переподготовке установленного образца.

## 5. МЕТОДИЧЕСКИЕ И ОЦЕНОЧНЫЕ МАТЕРИАЛЫ

Программа обеспечена образовательной онлайн-платформе:  
<https://platform.productstar.ru/login>.

Педагогические технологии:

- технология дифференцированного обучения;
- технология разноуровневого обучения;
- технология развивающего обучения;
- технология проблемного обучения;
- технология дистанционного обучения.

Методы обучения:

- словесный, наглядный практический;
- объяснительно – иллюстративный;
- частично-поисковый, исследовательский проблемный;
- игровой, дискуссионный.

**Дидактический материал:**

1. Герберт Шилдт, Java. Полное руководство. Издательство «Диалектика-Вильямс», 2018 год, 1488 с.

2. Кей Хорстманн, Java. Библиотека профессионала. Перевод с английского: Берштейн И. Издательство «Диалектика-Вильямс», 2020 год, 1-й том — 864 с., 2-й том — 1008 с.
3. Джошуа Блох, Java. Эффективное программирование. Перевод с английского: Красиков И. Издательство «Диалектика-Вильямс», 2019 год, 464 с.
4. Крейг Уоллс, Spring в действии. Перевод с английского: Киселёв А. Издательство «ДМК-Пресс», 2022 год, 544 с.

#### **Электронно-библиотечные ресурсы и системы, информационно-справочные системы:**

1. Научная электронная библиотека eLIBRARY.RU.
2. Собственные учебные материалы, размещённые в личном кабинете обучающегося.

#### **Оценочные материалы:**

Для отслеживания результатов освоения программы среди слушателей проводится текущий контроль, промежуточный контроль и итоговое оценивание.

#### **Текущий контроль**

Осуществление текущего контроля проводится после занятий в виде написания практических заданий или тестирований. Тематика и условия выполнения практических заданий расписаны в личном кабинете обучающегося на образовательной онлайн-платформе. Педагог проверяет решение и принимает решение о принятии решения (зачёт), о необходимости доработать решение или о незачёте. Если текущий контроль представлен в виде тестирования, подсчет верных ответов и выставление оценки «зачёт» и «незачёт» происходят в автоматическом режиме.

#### **Примеры тестовых вопросов**

1. Какой модификатор доступа в Java делает поле доступным только внутри текущего класса?
  - a) public
  - б) protected
  - в) private
  - г) default
2. Какой интерфейс реализует коллекция ArrayList?
  - a) Set
  - б) List
  - в) Map
  - г) Collection
3. Какой инструмент используется в Java для управления зависимостями и сборкой проектов?
  - a) Git
  - б) Maven
  - в) Postman
  - г) JUnit

4. Какой протокол чаще всего используется для реализации REST API?
- а) FTP
  - б) HTTP
  - в) SMTP
  - г) UDP
5. Какой оператор SQL используется для выборки данных из таблицы?
- а) INSERT
  - б) UPDATE
  - в) SELECT
  - г) DELETE

### Примеры домашних заданий

#### 1. Java Core (блок 2–3).

Реализовать класс «Библиотека» с возможностью:

- добавлять книги (название, автор, год),
- искать книгу по названию,
- выводить список всех книг.

Использовать коллекцию ArrayList для хранения данных.

#### 2. SQL + JDBC (блок 5).

Создать таблицу students с полями: id, name, age, grade.

Написать SQL-запросы:

- выбрать всех студентов старше 20 лет;
- посчитать средний балл;
- обновить оценку студента с конкретным id.

Подключить таблицу к Java-программе через JDBC.

#### 3. Spring Boot (блок 8).

Разработать простое CRUD-приложение для заметок:

- POST /notes — добавление заметки,
- GET /notes — получение всех заметок,
- PUT /notes/{id} — обновление заметки,
- DELETE /notes/{id} — удаление заметки.

Данные хранить в PostgreSQL.

### Промежуточный контроль

Промежуточный контроль проводится после изучения следующих блоков: 1, 3, 4, 6, 7, 8, 9, 10. Тематика и условия выполнения работ в рамках промежуточного контроля расписаны в личном кабинете обучающегося на образовательной онлайн-платформе. Педагог проверяет решение и принимает решение о принятии решения (зачет), о необходимости доработать решение или о незачете. Если промежуточный контроль представлен в виде тестирования, подсчет верных ответов и выставление оценки «зачёт» и «незачёт» происходят в автоматическом режиме на образовательной онлайн-платформе.

## Примеры проектов для промежуточной аттестации

### Проект 1. «Консольное приложение для управления задачами» (блоки 3–4)

Цель: Отработать навыки работы с коллекциями, файлами и потоками ввода-вывода.

Описание задания:

1. Реализовать консольное приложение «Менеджер задач» с функциями:
  - добавление, удаление и обновление задачи;
  - поиск задачи по ключевому слову;
  - сохранение задач в файл и загрузка при запуске.
2. Использовать коллекции (ArrayList, HashMap) для хранения.
3. Применить сериализацию для сохранения состояния.

Форма сдачи: проект на GitHub + инструкция по запуску.

### Проект 2. «Телеграм-бот на Java» (блок 6–7)

Цель: Закрепить знания ООП, SOLID и тестирования на практике.

Описание задания:

1. Разработать Telegram-бота, который умеет:
  - принимать команды от пользователя (например, «/help», «/addtask», «/list»);
  - сохранять задачи пользователя в базу данных (Postgres);
  - выводить список задач.
2. Реализовать архитектуру с использованием принципов ООП и SOLID.
3. Написать юнит-тесты на ключевые модули (JUnit + Mockito).

Форма сдачи: репозиторий на GitHub (код + README).

## Итоговое оценивание

В конце программы обучающиеся сдают итоговую аттестацию. Для успешного прохождения итогового контроля слушатель должен выполнить критерии, указанные в разделе 4.

Результаты текущего контроля, промежуточной аттестации и итогового оценивания отображаются в личном кабинете слушателя на образовательной онлайн-платформе: <https://platform.productstar.ru/login>.

По результатам сдачи текущего контроля, промежуточного контроля и итогового оценивания педагог даёт обратную связь слушателям, отмечает их сильные стороны и обращает внимание на зоны для развития. При необходимости педагог может повторить пройденные темы со слушателями, если установлен факт плохого закрепления и усвоения темы у слушателей.