

**ОБЩЕСТВО С ОГРАНИЧЕННОЙ ОТВЕТСТВЕННОСТЬЮ
«ИННОПРОГ»**

УТВЕРЖДАЮ



Генеральный директор ООО «Иннопрог»

Венедиктов Р.В.
подпись

/ Венедиктов Р.В.

инициалы, фамилия

30 декабря 2025 г.

Приказ об утверждении программы
от 30 декабря 2025 г. № ОБР-5

**ДОПОЛНИТЕЛЬНАЯ ОБЩЕОБРАЗОВАТЕЛЬНАЯ ПРОГРАММА
ДОПОЛНИТЕЛЬНАЯ ОБЩЕРАЗВИВАЮЩАЯ ПРОГРАММА**

«Java-разработчик»

Направленность программы:	техническая
Уровень освоения:	разноуровневый
Возраст обучающихся:	дети от 12 лет и взрослые
Трудоемкость:	800 академических часов
Форма обучения:	очная, с применением электронного обучения и дистанционных образовательных технологий
Режим реализации:	10 месяцев (40 учебных недель), рекомендуемая нагрузка – 20 академических часов в неделю
Язык обучения:	русский

Иннополис, 2026

Содержание

1	Пояснительная записка	3
1.1	Наименование программы	3
1.2	Нормативно-правовые основания разработки программы	3
1.3	Структура образовательной программы	4
1.4	Направленность и уровень программы	5
1.5	Актуальность программы	5
1.6	Цель реализации программы	5
1.7	Задачи программы	6
1.8	Адресат программы	6
1.9	Порядок приема, зачисления, перевода и прекращения обучения	7
1.10	Форма обучения, формы организации занятий и язык обучения	7
1.11	Объем, срок освоения и режим занятий	7
1.12	Документ об обучении	8
1.13	Профориентационная и практическая направленность	8
1.14	Планируемые результаты освоения программы	8
1.14.1	Перечень формируемых предметных результатов	8
1.14.2	Планируемые результаты обучения	9
1.14.3	Метапредметные и личностные результаты	12
2	Содержание программы	13
2.1	Учебный план	13
2.2	Календарный учебный график	16
2.3	Рабочие программы модулей	17
2.3.1	Модуль 1. Введение в профессию и организация обучения	19
2.3.2	Модуль 2. Java Core	21
2.3.3	Модуль 3. Объектно-ориентированное программирование на Java	22
2.3.4	Модуль 4. Коллекции, обобщения, исключения и Stream API	24
2.3.5	Модуль 5. Инструменты Java-разработчика: Git, Maven, Gradle и отладка	26

2.3.6	Модуль 6. Основы SQL	27
2.3.7	Модуль 7. Алгоритмы и структуры данных	29
2.3.8	Модуль 8. Web-основы, HTTP, JSON и клиентская часть	30
2.3.9	Модуль 9. Linux и среда исполнения Java-приложений	32
2.3.10	Модуль 10. PostgreSQL, JDBC, JPA и Hibernate . . .	34
2.3.11	Модуль 11. JavaFX и клиентские приложения на Java	35
2.3.12	Модуль 12. Spring Boot: REST API, безопасность, тестирование и деплой	37
2.3.13	Модуль 13. Проектная практика и консультации . . .	38
2.3.14	Модуль 14. Промежуточная аттестация по завершении Программы	40
2.4	Проектная практика и консультации	41
3	Организационно-педагогические условия реализации программы	43
3.1	Общие условия реализации	43
3.2	Материально-технические условия	43
3.3	Электронная информационно-образовательная среда	43
3.4	Кадровые условия	44
3.5	Условия для несовершеннолетних обучающихся	44
3.6	Условия для обучающихся с ограниченными возможностями здоровья	45
3.7	Учебно-методическое обеспечение	45
3.8	Применение электронного обучения и дистанционных образовательных технологий	45
3.9	Информационная безопасность, персональные данные и этика	46
3.10	Локальные процедуры обучения	46
3.11	Обновление программы	46
4	Оценка качества освоения программы	47
4.1	Общие положения	47
4.2	Текущий контроль успеваемости	47
4.3	Промежуточная аттестация	47
4.4	Пересдача, апелляция и подтверждение самостоятельности .	48
4.5	Допуск к промежуточной аттестации по завершении Программы	48

4.6	Промежуточная аттестация по завершении Программы . . .	48
4.6.1	Примерная тематика проектных работ	48
4.6.2	Обязательные требования к итоговому проекту . . .	50
4.6.3	Порядок проведения промежуточной аттестации по завершении Программы	51
4.6.4	Критерии оценивания итогового проекта	51
4.7	Внутренняя и внешняя оценка качества реализации программы	52
5	Методические материалы	53
5.1	Методические указания для обучающихся	53
5.2	Методические указания для преподавателей и проверяющих	53
5.3	Рекомендуемая литература и ресурсы	53
	Приложение 1. Матрица формирования компетенций	54
	Приложение 2. Оценочные материалы промежуточной аттестации .	56
	Приложение 3. Оценочные материалы промежуточной аттестации по завершении Программы	58
	Приложение 4. Перечень локальных нормативных актов, на которые должна опираться реализация программы	63
	Приложение 5. Лист актуализации программы	64
	Приложение 6. Требования к репозиторию, README и демонстрации	65
	Приложение 7. Дорожная карта подготовки итогового проекта	72
	Приложение 8. Контрольный лист соответствия требованиям программы	73

1. Пояснительная записка

1.1. Наименование программы

Дополнительная общеобразовательная общеразвивающая программа технической направленности «Java-разработчик» (далее – Программа) разработана для реализации обществом с ограниченной ответственностью «Иннопрог» в сфере дополнительного образования детей и взрослых.

Программа утверждена и введена в действие с 30 декабря 2025 года приказом ООО «ИННОПРОГ» от 30.12.2025 № ОБР-5 «О дополнении перечня образовательных программ».

1.2. Нормативно-правовые основания разработки программы

Программа разработана в соответствии с:

- Федеральным законом от 29.12.2012 № 273-ФЗ «Об образовании в Российской Федерации»;
- приказом Министерства просвещения Российской Федерации от 27.07.2022 № 629 «Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам»;
- постановлением Правительства Российской Федерации от 11.10.2023 № 1678 «Об утверждении Правил применения организациями, осуществляющими образовательную деятельность, электронного обучения, дистанционных образовательных технологий при реализации образовательных программ»;
- постановлением Правительства Российской Федерации от 18.09.2020 № 1490 «О лицензировании образовательной деятельности»;
- постановлением Правительства Российской Федерации от 15.09.2020 № 1441 «Об утверждении Правил оказания платных образовательных услуг»;
- Федеральным законом от 27.07.2006 № 152-ФЗ «О персональных данных»;
- постановлением Главного государственного санитарного врача Российской Федерации от 02.06.2026 № 19, которым утверждены

санитарные правила СП 2.4.2.4283-26 «Санитарно-эпидемиологические требования к организациям воспитания и обучения, отдыха и оздоровления детей и молодежи» (применяются с 01.09.2026);

- ГОСТ Р 7.0.97-2025 «Система стандартов по информации, библиотечному и издательскому делу. Организационно-распорядительная документация. Требования к оформлению документов» в части оформления организационно-распорядительных реквизитов;
- локальными нормативными актами ООО «Иннопрог», регуливающими разработку и утверждение дополнительных общеобразовательных программ, прием и зачисление, организацию образовательного процесса, обучение по индивидуальному учебному плану, применение электронного обучения и дистанционных образовательных технологий, текущий контроль и промежуточную аттестацию, выдачу документов об обучении, защиту персональных данных, охрану здоровья обучающихся и оказание платных образовательных услуг.

Программа реализуется за пределами федеральных государственных образовательных стандартов и федеральных государственных требований. Она не относится к дополнительным профессиональным программам, не является программой профессионального обучения и не предусматривает присвоение профессии, специальности, квалификации, квалификационного разряда или права на ведение нового вида профессиональной деятельности.

1.3. Структура образовательной программы

Программа представляет собой комплекс основных характеристик образования, организационно-педагогических условий, оценочных и методических материалов. В состав Программы включены:

- пояснительная записка, цель, задачи, адресат, условия приема, объем, срок и режим освоения;
- планируемые предметные, метапредметные и личностные результаты;
- учебный план, календарный учебный график и рабочие программы модулей;

- организационно-педагогические условия, формы контроля, оценочные и методические материалы;
- описание проектной практики и промежуточной аттестации по завершении Программы в форме защиты итогового проекта.

1.4. Направленность и уровень программы

Программа имеет техническую направленность и разноуровневый характер. Стартовый уровень обеспечивает освоение терминов и базовых операций; базовый – самостоятельное решение типовых задач; углубленный – интеграцию нескольких технологий в проекте. Индивидуальная траектория определяется входной диагностикой, темпом освоения, консультациями и уровнем проектных заданий.

1.5. Актуальность программы

Программа обусловлена устойчивым спросом на специалистов, владеющих современным стеком Java-разработки, способных писать надежный код на Java, проектировать объектные модели, использовать реляционные базы данных, разрабатывать REST API и backend-сервисы на Spring Boot, создавать клиентские JavaFX-приложения, выполнять тестирование, документирование, сборку и подготовку приложений к развертыванию в Linux-среде. Программа ориентирована на практико-ориентированное формирование компетенций, востребованных в коммерческой разработке, корпоративных информационных системах, продуктовых командах, сервисной разработке и цифровизации внутренних процессов организаций.

1.6. Цель реализации программы

Цель Программы – формирование у детей от 12 лет и взрослых устойчивого интереса, предметных результатов и опыта проектной деятельности в области разработки программ на Java, объектно-ориентированного проектирования, работы с базами данных, тестирования и создания серверных приложений, развитие цифровой грамотности, алгоритмического мышления, самостоятельности, культуры безопасной и ответственной работы с информационными технологиями.

1.7. Задачи программы

Задачами Программы являются:

- формирование устойчивой базы по синтаксису, стандартной библиотеке и идиомам Java;
- развитие алгоритмического мышления и навыков решения прикладных задач;
- освоение принципов объектно-ориентированного проектирования, декомпозиции и разделения ответственности;
- формирование навыков командной разработки с использованием Git, Maven и Gradle;
- освоение реляционных баз данных, языка SQL, PostgreSQL, JDBC, JPA и Hibernate;
- приобретение навыков разработки клиентских приложений на JavaFX;
- освоение backend-разработки на Spring Boot, проектирования REST API, валидации, обработки ошибок и документирования интерфейсов;
- освоение Spring Security, базовых моделей авторизации, разграничения ролей и безопасной обработки пользовательских данных;
- формирование навыков тестирования с использованием JUnit, Mockito, Spring Boot Test и интеграционных проверок;
- формирование навыков подготовки программного продукта к развертыванию и сопровождению с использованием Linux, Docker, Nginx, журналирования, переменных окружения и базовых CI/CD-практик;
- развитие культуры инженерной разработки: документирование, тестирование, читаемость, структурированность и сопровождение кода.

1.8. Адресат программы

Программа предназначена для детей от 12 лет и взрослых, заинтересованных в освоении направления «Java-разработка». Наличие среднего профессионального или высшего образования не требуется. При приеме несовершеннолетнего договор заключается с его законным

представителем либо с участием законного представителя в порядке, установленном законодательством и локальными актами организации.

Рекомендуемые начальные условия: уверенная работа с компьютером и браузером, умение создавать и сохранять файлы, базовая математическая и информационная грамотность, готовность регулярно выполнять практические задания. Входная диагностика используется для подбора стартового уровня и не является конкурсным испытанием.

1.9. Порядок приема, зачисления, перевода и прекращения обучения

Прием осуществляется на основании заявления, договора об образовании и документов, предусмотренных локальными актами организации. Зачисление оформляется приказом. Перевод на индивидуальный учебный план, изменение срока, восстановление и прекращение образовательных отношений осуществляются в порядке, установленном законодательством, договором и локальными нормативными актами ООО «Иннопрог».

1.10. Форма обучения, формы организации занятий и язык обучения

Форма обучения – очная, с применением электронного обучения и дистанционных образовательных технологий. Занятия проводятся в синхронном и асинхронном форматах и включают объяснение нового материала, демонстрации, практикумы, самостоятельную работу, консультации, текущий контроль, промежуточную аттестацию по модулям и проектную практику. Язык обучения – русский. Академический час равен 45 минутам.

1.11. Объем, срок освоения и режим занятий

Трудоемкость Программы составляет 800 академических часов, включая теоретические и практические занятия, самостоятельную работу, текущий контроль, промежуточную аттестацию по модулям, проектную практику и защиту итогового проекта по завершении Программы.

Рекомендуемый срок освоения составляет 10 месяцев (40 учебных недель) при средней нагрузке 20 академических часов в неделю. Конкретное расписание может предусматривать различное соотношение занятий с

преподавателем и самостоятельной работы. Допускается обучение по индивидуальному учебному плану, в том числе ускоренное, при сохранении общего объема и достижении планируемых результатов.

1.12. Документ об обучении

Обучающемуся, освоившему Программу, выполнившему обязательные задания и успешно прошедшему промежуточную аттестацию по завершении Программы в форме защиты итогового проекта, выдается сертификат об обучении по образцу, самостоятельно установленному ООО «Иннопрог».

Сертификат является документом об обучении, подтверждает факт и объем освоения дополнительной общеобразовательной общеразвивающей программы, но не является документом об образовании и (или) о квалификации, не присваивает квалификацию и не заменяет диплом о профессиональной переподготовке.

Лицу, освоившему часть Программы, отчисленному до ее завершения или не прошедшему аттестацию, по его заявлению может быть выдана справка об обучении или о периоде обучения по образцу, установленному организацией.

1.13. Профориентационная и практическая направленность

Содержание Программы соотнесено с типовыми современными задачами по направлению «Java-разработка» исключительно как профориентационный и содержательный ориентир. Профессиональные стандарты могут использоваться при актуализации тем и критериев качества проектов, однако Программа не заявляет подготовку к выполнению трудовых функций в полном объеме и не устанавливает уровень квалификации выпускника.

1.14. Планируемые результаты освоения программы

1.14.1. Перечень формируемых предметных результатов

Таблица 1 – Формируемые предметные результаты

Код	Наименование предметного результата
ПР-1	Разрабатывать консольные и серверные программы на языке Java, применять базовые конструкции языка, стандартную библиотеку, обработку исключений, ввод-вывод и принципы надежной организации кода.
ПР-2	Проектировать программные компоненты с использованием объектно-ориентированного подхода, инкапсуляции, наследования, полиморфизма, интерфейсов, абстракций и принципов качественного кода.
ПР-3	Использовать Git, Maven, Gradle, средства отладки, статической проверки и документирования для индивидуальной и командной разработки Java-проектов.
ПР-4	Проектировать реляционные структуры данных, писать SQL-запросы, обеспечивать целостность данных и применять базовые приемы оптимизации запросов.
ПР-5	Выбирать и реализовывать алгоритмы и структуры данных с учетом сложности, ограничений задачи, требований к памяти и читаемости реализации.
ПР-6	Понимать клиент-серверное взаимодействие, HTTP, JSON, основы web-интерфейсов и проектировать REST API для прикладных Java-сервисов.
ПР-7	Использовать Linux-среду, JDK, переменные окружения, журналы, Docker и базовые серверные настройки для запуска и сопровождения Java-приложений.
ПР-8	Использовать PostgreSQL, JDBC, JPA, Hibernate, транзакции, миграции и ORM-подход для интеграции Java-приложений с базами данных.
ПР-9	Разрабатывать клиентские настольные приложения на JavaFX, проектировать формы, обработчики событий, навигацию, таблицы данных и связь интерфейса с бизнес-логикой.
ПР-10	Разрабатывать backend-приложения на Spring Boot, включая REST-контроллеры, сервисный слой, Spring Data JPA, Spring Security, тестирование, документацию API и подготовку к развертыванию.

1.14.2. Планируемые результаты обучения

В результате освоения Программы обучающийся должен:

знать:

- синтаксис, типовую структуру проекта и основные конструкции языка Java;
- принципы алгоритмизации, структуры данных и типовые алгоритмы;

- основы объектно-ориентированного проектирования, SOLID-принципов на прикладном уровне и разделения ответственности;
- основы командной разработки, Git, Maven, Gradle и жизненного цикла сборки;
- основы проектирования реляционных баз данных, SQL, PostgreSQL, JDBC, JPA и Hibernate;
- базовые принципы клиент-серверного взаимодействия, HTTP, JSON, REST API и web-интерфейсов;
- устройство и базовые средства администрирования Linux-среды для запуска Java-приложений;
- архитектуру JavaFX-приложений и Spring Boot-сервисов;
- требования к качеству кода, тестированию, логированию и подготовке проекта к развертыванию;
- назначение Docker, переменных окружения, reverse proxy, базовых CI/CD-процессов и документации API;
- базовые требования к безопасности backend-приложений: хранение секретов, разграничение доступа, проверка пользовательского ввода, защита маршрутов и обработка ошибок.

уметь:

- писать, читать и отлаживать код на Java;
- проектировать и использовать классы, интерфейсы, пакеты, модули и слои приложения;
- работать с коллекциями, исключениями, файлами, Stream API и внешними библиотеками;
- использовать Git, Maven и Gradle для сопровождения проекта и командной работы;
- писать SQL-запросы и проектировать структуру реляционной базы данных;
- создавать JavaFX-приложения с формами, таблицами, событиями и сервисным слоем;
- создавать Spring Boot REST API, интегрировать приложение с PostgreSQL и реализовывать бизнес-логику;

- применять Spring Security, роли, валидацию, обработку ошибок и документацию API;
- использовать Linux-командную строку и базовые средства настройки окружения;
- писать модульные и интеграционные тесты, настраивать логирование и хранить конфигурацию проекта через переменные окружения;
- подготавливать проект к запуску через Docker, JVM, Nginx и базовые сценарии CI/CD;
- документировать решения, подготавливать проект к передаче и защите.

владеть навыками:

- практической разработки программ на Java;
- выбора и применения структур данных и алгоритмов;
- организации Java-проекта и работы с исходным кодом в репозитории;
- проектирования и сопровождения БД;
- разработки клиентских JavaFX-приложений и backend-логики Spring Boot-сервисов;
- взаимодействия с ORM и SQL-инструментами;
- запуска и сопровождения проекта в Linux-среде;
- базового контейнеризированного запуска, настройки переменных окружения, ведения логов и подготовки инструкции для деплоя;
- выполнения итогового проекта от постановки задачи до защиты результата.

1.14.3. Метапредметные и личностные результаты

По завершении Программы обучающийся способен:

- ставить учебную задачу, разбивать ее на этапы, планировать сроки и контролировать продвижение;
- искать, сопоставлять и критически оценивать техническую информацию, документацию и примеры;
- объяснять выбранное решение, воспринимать обратную связь, аргументированно вносить изменения и представлять результат;
- соблюдать правила командного взаимодействия, академической добросовестности и корректного указания источников;
- ответственно работать с персональными данными, учетными записями, секретами и пользовательской информацией;
- использовать генеративные инструменты и иные средства искусственного интеллекта как вспомогательный ресурс, проверяя результат и сохраняя личную ответственность за выполненную работу;
- проявлять самостоятельность, аккуратность, настойчивость при поиске ошибок и уважение к результатам чужого труда.

2. Содержание программы

2.1. Учебный план

Таблица 2 – Учебный план

№	Наименование модуля	Всего, ак. ч.	Лек- ции	Прак- тика	Сам. работа	Пром. атт.	Итог. атт.	Компетенции
1	Введение в профессию и организация обучения	12	6	3	3	0	0	ПР-1–ПР-10
2	Java Core	116	41	46	23	6	0	ПР-1
3	Объектно-ориентированное программирование на Java	80	26	31	17	6	0	ПР-2
4	Коллекции, обобщения, исключения и Stream API	56	20	20	11	5	0	ПР-1, ПР-2
5	Инструменты Java-разработчика: Git, Maven, Gradle и отладка	36	12	12	6	6	0	ПР-3
6	Основы SQL	36	12	12	6	6	0	ПР-4
7	Алгоритмы и структуры данных	68	23	23	17	5	0	ПР-1, ПР-5
8	Web-основы, HTTP, JSON и клиентская часть	56	17	20	14	5	0	ПР-6
9	Linux и среда исполнения Java-приложений	44	14	14	11	5	0	ПР-7
10	PostgreSQL, JDBC, JPA и Hibernate	68	23	23	17	5	0	ПР-4, ПР-8
11	JavaFX и клиентские приложения на Java	56	17	20	14	5	0	ПР-9
12	Spring Boot: REST API, безопасность, тестирование и деплой	104	32	37	26	9	0	ПР-6, ПР-8, ПР-10
13	Проектная практика и консультации	36	6	18	12	0	0	ПР-1–ПР-10

Продолжение таблицы 2

№	Наименование модуля	Всего, ак. ч.	Лек- ции	Прак- тика	Сам. работа	Пром. атт.	Итог. атт.	Компетенции
14	Промежуточная аттестация по завершении Программы	32	0	0	0	0	32	ПР-1–ПР-10
	Итого	800	249	279	177	63	32	

2.2. Календарный учебный график

Календарный учебный график приведен для рекомендуемого режима освоения Программы продолжительностью 10 месяцев (40 учебных недель). Конкретные даты занятий, консультаций, промежуточной аттестации и промежуточной аттестации по завершении Программы определяются расписанием, договором об образовании и приказом о зачислении.

Таблица 3 – Календарный учебный график

Учебные недели	Модуль	Трудоемкость, ак. ч.	Основные виды учебной деятельности
1	Введение в профессию и организация обучения	12	изучение материалов, практические задания, консультации, контрольные мероприятия
2–7	Java Core	116	изучение материалов, практические задания, консультации, контрольные мероприятия
8–11	Объектно-ориентированное программирование на Java	80	изучение материалов, практические задания, консультации, контрольные мероприятия
12–14	Коллекции, обобщения, исключения и Stream API	56	изучение материалов, практические задания, консультации, контрольные мероприятия
15–16	Инструменты Java-разработчика: Git, Maven, Gradle и отладка	36	изучение материалов, практические задания, консультации, контрольные мероприятия
17–18	Основы SQL	36	изучение материалов, практические задания, консультации, контрольные мероприятия
19–21	Алгоритмы и структуры данных	68	изучение материалов, практические задания, консультации, контрольные мероприятия

Учебные недели	Модуль	Трудоемкость, ак. ч.	Основные виды учебной деятельности
22–24	Web-основы, HTTP, JSON и клиентская часть	56	изучение материалов, практические задания, консультации, контрольные мероприятия
25–26	Linux и среда исполнения Java-приложений	44	изучение материалов, практические задания, консультации, контрольные мероприятия
27–29	PostgreSQL, JDBC, JPA и Hibernate	68	изучение материалов, практические задания, консультации, контрольные мероприятия
30–32	JavaFX и клиентские приложения на Java	56	изучение материалов, практические задания, консультации, контрольные мероприятия
33–37	Spring Boot: REST API, безопасность, тестирование и деплой	104	изучение материалов, практические задания, консультации, контрольные мероприятия
38–39	Проектная практика и консультации	36	изучение материалов, практические задания, консультации, контрольные мероприятия
40	Промежуточная аттестация по завершении Программы	32	изучение материалов, практические задания, консультации, контрольные мероприятия

2.3. Рабочие программы модулей

Рабочие программы модулей раскрывают содержание обучения, практическую подготовку, самостоятельную работу, промежуточный контроль и ожидаемые результаты. Каждый модуль связан с проектной траекторией: обучающийся постепенно формирует репозиторий, реализует отдельные

компоненты будущего приложения, проходит ревью и получает обратную связь по качеству кода, архитектуре, тестам и документации.

Проектная траектория программы. В течение обучения обучающийся выполняет серию практических работ, которые могут быть объединены в итоговый проект. Проектная траектория строится по принципу усложнения: от консольного приложения и объектной модели к базе данных, клиентскому интерфейсу, REST API, Spring Boot-сервису, тестам и подготовке к развертыванию.

Таблица 4 – Примерная проектная траектория и темы проектных работ

№	Тема проекта	Краткая характеристика результата
1	Консольный менеджер задач на Java	приложение для создания, редактирования, поиска, фильтрации и закрытия задач с хранением данных в файле или базе данных.
2	Система учета товаров на складе	сервис для ведения номенклатуры, остатков, цен, поступлений и списаний с отчетом по доступным позициям.
3	База сотрудников и зарплат	приложение для хранения данных сотрудников, расчета начислений, поиска, сортировки и формирования управленческого отчета.
4	Сервис бронирования поездок	прикладная система для учета клиентов, автомобилей, маршрутов, времени поездки и статуса бронирования.
5	Сервис управления пассажирскими перевозками	приложение для учета транспорта, вместимости, пассажиров, посадки, свободных мест и истории рейсов.
6	Key-Value хранилище на дереве поиска	собственная реализация структуры хранения с добавлением, поиском, обновлением, удалением и обходом данных.
7	Файловый каталог книг и читателей	консольное или JavaFX-приложение для учета книг, читателей, выдачи, возврата и просроченных записей.
8	SQL-каталог товаров	Java-приложение с PostgreSQL, CRUD-операциями, фильтрацией, сортировкой и отчетами по ассортименту.

№	Тема проекта	Краткая характеристика результата
9	JavaFX-приложение для личных финансов	настольное приложение с формами, таблицами, категориями расходов, диаграммами и сохранением данных.
10	JavaFX-система бронирования кабинетов	клиентское приложение для выбора даты, времени, помещения, ответственного лица и статуса заявки.
11	REST API менеджера задач на Spring Boot	backend-сервис с задачами, пользователями, статусами, валидацией, обработкой ошибок и документацией API.
12	Сервис учета заказов с PostgreSQL	Spring Boot приложение с товарами, заказами, клиентами, транзакциями, миграциями и интеграционными тестами.
13	Интернет-магазин на Spring Boot	web/backend-проект с каталогом, корзиной, заказами, ролями пользователей, административной частью и PostgreSQL.
14	Сервис авторизации и личного кабинета	приложение с регистрацией, входом, ролями, Spring Security, защищенными маршрутами и пользовательским профилем.
15	Продакшн-версия Java backend-приложения	итоговый проект с Docker, переменными окружения, логированием, тестами, README, инструкцией деплоя и демонстрацией работоспособности.

2.3.1. Модуль 1. Введение в профессию и организация обучения

Цель модуля – сформировать понимание целей программы, требований к результатам, роли Java-разработчика и правил организации образовательного процесса.

Планируемые результаты освоения модуля:

- понимать структуру программы и требования к результатам обучения;
- знать правила работы в электронной среде, порядок контроля и аттестации;

- выбрать предварительную проектную траекторию и подготовить репозиторий.

Таблица 5 – Учебно-тематический план модуля 1

№	Тема	Объем, ак. ч.
1	Профессия Java-разработчика и типовые задачи	2
2	Структура программы, правила обучения и аттестации	2
3	Среда разработки, JDK, IDE и репозиторий	2
4	Индивидуальная траектория и выбор проектной темы	2

Содержание модуля раскрывается через последовательное изучение тем, выполнение практических заданий и подготовку результата, который включается в индивидуальное портфолио обучающегося. Материал модуля связан с итоговым проектом: обучающийся не только выполняет отдельные упражнения, но и переносит освоенные приемы в собственное приложение.

Практическая подготовка включает:

- создание учебного репозитория и первичной структуры проекта;
- настройка среды разработки, JDK и системы контроля версий;
- заполнение индивидуальной карты обучения и выбор предварительной темы проекта.

Самостоятельная работа включает:

- изучение регламентов обучения и критериев оценивания;
- проверка доступа к электронной информационно-образовательной среде;
- подготовка списка вопросов по выбранной проектной траектории.

Промежуточная проверка результата. зачет проводится в форме проверки готовности среды разработки, репозитория и индивидуального плана обучения.

Минимальный уровень освоения предполагает, что обучающийся выполняет обязательную часть задания, может запустить проект, объяснить основные решения и исправить типовые ошибки по замечаниям преподавателя. Повышенный уровень предполагает аккуратную архитектуру, обработку крайних случаев, наличие тестов, понятную документацию, расширенную функциональность и самостоятельное обоснование выбранных решений.

Типовые ошибки, учитываемые при проверке:

- не настроен JDK или IDE;
- отсутствует доступ к репозиторию;
- не выбран предварительный проектный контур;
- обучающийся не ознакомился с правилами аттестации.

2.3.2. Модуль 2. Java Core

Цель модуля – сформировать базовые навыки программирования на Java, чтения и написания корректного кода, решения прикладных задач и отладки программ.

Планируемые результаты освоения модуля:

- писать Java-код с переменными, условиями, циклами, массивами, методами и строками;
- разбивать задачу на функции и проверять корректность результата;
- читать сообщения компилятора и выполнять базовую отладку.

Таблица 6 – Учебно-тематический план модуля 2

№	Тема	Объем, ак. ч.
1	Установка JDK, IDE и первая Java-программа	4
2	Структура класса, метод main и компиляция	6
3	Переменные, типы данных и преобразования	6
4	Операторы, выражения и приоритеты	6
5	Ввод, вывод и форматирование данных	6
6	Условные конструкции	6
7	Циклы и вложенные циклы	8
8	Массивы и базовая обработка коллекций данных	6
9	Методы, параметры, возвращаемые значения и декомпозиция	8
10	Строки, StringBuilder и регулярные выражения на базовом уровне	6
11	Отладка, типовые ошибки и чтение stack trace	6
12	Мини-проект Java Core	6

Содержание модуля раскрывается через последовательное изучение тем, выполнение практических заданий и подготовку результата, который включается в индивидуальное портфолио обучающегося. Материал модуля

связан с итоговым проектом: обучающийся не только выполняет отдельные упражнения, но и переносит освоенные приемы в собственное приложение.

Практическая подготовка включает:

- выполнение практикума по темам модуля 2;
- доработка учебного или итогового проекта с применением изученных инструментов;
- оформление результата в репозитории с понятной историей изменений и кратким описанием.

Самостоятельная работа включает:

- повторение теоретического материала и разбор типовых ошибок;
- самостоятельное решение дополнительных задач;
- подготовка фрагмента проекта к проверке и защите.

Промежуточная проверка результата. зачет проводится по результатам практического задания, проверки репозитория, краткого устного опроса и анализа качества выполненного кода.

Минимальный уровень освоения предполагает, что обучающийся выполняет обязательную часть задания, может запустить проект, объяснить основные решения и исправить типовые ошибки по замечаниям преподавателя. Повышенный уровень предполагает аккуратную архитектуру, обработку крайних случаев, наличие тестов, понятную документацию, расширенную функциональность и самостоятельное обоснование выбранных решений.

Типовые ошибки, учитываемые при проверке:

- код не компилируется или не запускается;
- решение не покрывает обязательные требования задания;
- отсутствует разделение ответственности между компонентами;
- не обработаны типовые ошибки пользователя;
- нет понятной инструкции запуска.

2.3.3. Модуль 3. Объектно-ориентированное программирование на Java

Цель модуля – сформировать навыки объектно-ориентированного проектирования и построения поддерживаемой доменной модели приложения.

Планируемые результаты освоения модуля:

- проектировать классы и объекты предметной области;

- применять инкапсуляцию, наследование, полиморфизм и интерфейсы;
- строить простую архитектуру приложения с разделением ответственности.

Таблица 7 – Учебно-тематический план модуля 3

№	Тема	Объем, ак. ч.
1	Классы, объекты и состояние	6
2	Инкапсуляция, модификаторы доступа и неизменяемость	6
3	Наследование и повторное использование кода	6
4	Полиморфизм и динамическое связывание	6
5	Абстрактные классы и интерфейсы	6
6	Пакеты, слои приложения и архитектура учебного проекта	4
7	Исключения в объектной модели	6
8	Enum, value objects и доменные сущности	4
9	Диаграмма классов и проектирование модели	6
10	Объектный мини-проект	6

Содержание модуля раскрывается через последовательное изучение тем, выполнение практических заданий и подготовку результата, который включается в индивидуальное портфолио обучающегося. Материал модуля связан с итоговым проектом: обучающийся не только выполняет отдельные упражнения, но и переносит освоенные приемы в собственное приложение.

Практическая подготовка включает:

- выполнение практикума по темам модуля 3;
- доработка учебного или итогового проекта с применением изученных инструментов;
- оформление результата в репозитории с понятной историей изменений и кратким описанием.

Самостоятельная работа включает:

- повторение теоретического материала и разбор типовых ошибок;
- самостоятельное решение дополнительных задач;
- подготовка фрагмента проекта к проверке и защите.

Промежуточная проверка результата. зачет проводится по результатам практического задания, проверки репозитория, краткого устного опроса и анализа качества выполненного кода.

Минимальный уровень освоения предполагает, что обучающийся выполняет обязательную часть задания, может запустить проект, объяснить основные решения и исправить типовые ошибки по замечаниям преподавателя. Повышенный уровень предполагает аккуратную архитектуру, обработку крайних случаев, наличие тестов, понятную документацию, расширенную функциональность и самостоятельное обоснование выбранных решений.

Типовые ошибки, учитываемые при проверке:

- код не компилируется или не запускается;
- решение не покрывает обязательные требования задания;
- отсутствует разделение ответственности между компонентами;
- не обработаны типовые ошибки пользователя;
- нет понятной инструкции запуска.

2.3.4. Модуль 4. Коллекции, обобщения, исключения и Stream API

Цель модуля – научить обучающегося использовать коллекции, обобщения, исключения, файловый ввод-вывод и Stream API для обработки данных в Java-приложениях.

Планируемые результаты освоения модуля:

- использовать коллекции, обобщения и Stream API;
- обрабатывать ошибки и крайние случаи;
- реализовывать файловое хранение и обработку наборов данных.

Таблица 8 – Учебно-тематический план модуля 4

№	Тема	Объем, ак. ч.
1	Коллекции List, Set и Map	6
2	Обобщения и типобезопасность	4
3	Comparator, Comparable и сортировка объектов	4
4	Исключения и устойчивость приложения	4
5	Файловый ввод-вывод и сериализация данных	6
6	Lambda-выражения и функциональные интерфейсы	4

№	Тема	Объем, ак. ч.
7	Stream API и обработка наборов данных	6
8	Optional и работа с отсутствующими значениями	3
9	Практикум по коллекциям и потоковой обработке	3

Содержание модуля раскрывается через последовательное изучение тем, выполнение практических заданий и подготовку результата, который включается в индивидуальное портфолио обучающегося. Материал модуля связан с итоговым проектом: обучающийся не только выполняет отдельные упражнения, но и переносит освоенные приемы в собственное приложение.

Практическая подготовка включает:

- выполнение практикума по темам модуля 4;
- доработка учебного или итогового проекта с применением изученных инструментов;
- оформление результата в репозитории с понятной историей изменений и кратким описанием.

Самостоятельная работа включает:

- повторение теоретического материала и разбор типовых ошибок;
- самостоятельное решение дополнительных задач;
- подготовка фрагмента проекта к проверке и защите.

Промежуточная проверка результата. зачет проводится по результатам практического задания, проверки репозитория, краткого устного опроса и анализа качества выполненного кода.

Минимальный уровень освоения предполагает, что обучающийся выполняет обязательную часть задания, может запустить проект, объяснить основные решения и исправить типовые ошибки по замечаниям преподавателя. Повышенный уровень предполагает аккуратную архитектуру, обработку крайних случаев, наличие тестов, понятную документацию, расширенную функциональность и самостоятельное обоснование выбранных решений.

Типовые ошибки, учитываемые при проверке:

- код не компилируется или не запускается;

- решение не покрывает обязательные требования задания;
- отсутствует разделение ответственности между компонентами;
- не обработаны типовые ошибки пользователя;
- нет понятной инструкции запуска.

2.3.5. Модуль 5. Инструменты Java-разработчика: Git, Maven, Gradle и отладка

Цель модуля – сформировать инженерную культуру Java-разработчика, включая Git, Maven, Gradle, отладку, оформление репозитория и проверку качества кода.

Планируемые результаты освоения модуля:

- вести проект в Git и оформлять историю изменений;
- собирать проект через Maven или Gradle;
- настраивать отладку, стиль кода и базовую проверку качества.

Таблица 9 – Учебно-тематический план модуля 5

№	Тема	Объем, ак. ч.
1	Git и жизненный цикл изменений	4
2	Структура Java-проекта и соглашения о коде	4
3	Maven: зависимости, фазы сборки и структура проекта	5
4	Gradle: задачи, плагины и сборка проекта	4
5	Отладка, breakpoints, watch expressions и диагностика	4
6	Code style, статический анализ и подготовка pull request	3

Содержание модуля раскрывается через последовательное изучение тем, выполнение практических заданий и подготовку результата, который включается в индивидуальное портфолио обучающегося. Материал модуля связан с итоговым проектом: обучающийся не только выполняет отдельные упражнения, но и переносит освоенные приемы в собственное приложение.

Практическая подготовка включает:

- выполнение практикума по темам модуля 5;
- доработка учебного или итогового проекта с применением изученных инструментов;

- оформление результата в репозитории с понятной историей изменений и кратким описанием.

Самостоятельная работа включает:

- повторение теоретического материала и разбор типовых ошибок;
- самостоятельное решение дополнительных задач;
- подготовка фрагмента проекта к проверке и защите.

Промежуточная проверка результата. зачет проводится по результатам практического задания, проверки репозитория, краткого устного опроса и анализа качества выполненного кода.

Минимальный уровень освоения предполагает, что обучающийся выполняет обязательную часть задания, может запустить проект, объяснить основные решения и исправить типовые ошибки по замечаниям преподавателя. Повышенный уровень предполагает аккуратную архитектуру, обработку крайних случаев, наличие тестов, понятную документацию, расширенную функциональность и самостоятельное обоснование выбранных решений.

Типовые ошибки, учитываемые при проверке:

- код не компилируется или не запускается;
- решение не покрывает обязательные требования задания;
- отсутствует разделение ответственности между компонентами;
- не обработаны типовые ошибки пользователя;
- нет понятной инструкции запуска.

2.3.6. Модуль 6. Основы SQL

Цель модуля – сформировать базовые навыки работы с реляционными данными и SQL, необходимые для разработки приложений с постоянным хранением данных.

Планируемые результаты освоения модуля:

- проектировать таблицы и связи;
- писать SQL-запросы для выборки, изменения и агрегации данных;
- подготавливать схему БД для учебного приложения.

Таблица 10 – Учебно-тематический план модуля 6

№	Тема	Объем, ак. ч.
1	Реляционная модель данных	4
2	SELECT, WHERE, ORDER BY и LIMIT	4
3	JOIN и связи таблиц	4
4	Группировка, агрегаты и HAVING	4
5	INSERT, UPDATE, DELETE и транзакционная аккуратность	4
6	Проектирование простой схемы данных	4

Содержание модуля раскрывается через последовательное изучение тем, выполнение практических заданий и подготовку результата, который включается в индивидуальное портфолио обучающегося. Материал модуля связан с итоговым проектом: обучающийся не только выполняет отдельные упражнения, но и переносит освоенные приемы в собственное приложение.

Практическая подготовка включает:

- выполнение практикума по темам модуля 6;
- доработка учебного или итогового проекта с применением изученных инструментов;
- оформление результата в репозитории с понятной историей изменений и кратким описанием.

Самостоятельная работа включает:

- повторение теоретического материала и разбор типовых ошибок;
- самостоятельное решение дополнительных задач;
- подготовка фрагмента проекта к проверке и защите.

Промежуточная проверка результата. зачет проводится по результатам практического задания, проверки репозитория, краткого устного опроса и анализа качества выполненного кода.

Минимальный уровень освоения предполагает, что обучающийся выполняет обязательную часть задания, может запустить проект, объяснить основные решения и исправить типовые ошибки по замечаниям преподавателя. Повышенный уровень предполагает аккуратную архитектуру, обработку

крайних случаев, наличие тестов, понятную документацию, расширенную функциональность и самостоятельное обоснование выбранных решений.

Типовые ошибки, учитываемые при проверке:

- код не компилируется или не запускается;
- решение не покрывает обязательные требования задания;
- отсутствует разделение ответственности между компонентами;
- не обработаны типовые ошибки пользователя;
- нет понятной инструкции запуска.

2.3.7. Модуль 7. Алгоритмы и структуры данных

Цель модуля – развить алгоритмическое мышление, навыки выбора структур данных и умение оценивать эффективность решений.

Планируемые результаты освоения модуля:

- оценивать сложность алгоритмов;
- реализовывать базовые структуры данных;
- подбирать алгоритм под ограничения задачи и тестировать его.

Таблица 11 – Учебно-тематический план модуля 7

№	Тема	Объем, ак. ч.
1	Оценка сложности алгоритмов	4
2	Линейный поиск и бинарный поиск	4
3	Сортировки и анализ компромиссов	6
4	Стек, очередь и deque	6
5	Связные списки	4
6	Деревья поиска и обходы	8
7	HashMap, хеширование и коллизии	6
8	Алгоритмический практикум	6
9	Качество алгоритмического кода и тестовые наборы	4

Содержание модуля раскрывается через последовательное изучение тем, выполнение практических заданий и подготовку результата, который включается в индивидуальное портфолио обучающегося. Материал модуля

связан с итоговым проектом: обучающийся не только выполняет отдельные упражнения, но и переносит освоенные приемы в собственное приложение.

Практическая подготовка включает:

- выполнение практикума по темам модуля 7;
- доработка учебного или итогового проекта с применением изученных инструментов;
- оформление результата в репозитории с понятной историей изменений и кратким описанием.

Самостоятельная работа включает:

- повторение теоретического материала и разбор типовых ошибок;
- самостоятельное решение дополнительных задач;
- подготовка фрагмента проекта к проверке и защите.

Промежуточная проверка результата. зачет проводится по результатам практического задания, проверки репозитория, краткого устного опроса и анализа качества выполненного кода.

Минимальный уровень освоения предполагает, что обучающийся выполняет обязательную часть задания, может запустить проект, объяснить основные решения и исправить типовые ошибки по замечаниям преподавателя. Повышенный уровень предполагает аккуратную архитектуру, обработку крайних случаев, наличие тестов, понятную документацию, расширенную функциональность и самостоятельное обоснование выбранных решений.

Типовые ошибки, учитываемые при проверке:

- код не компилируется или не запускается;
- решение не покрывает обязательные требования задания;
- отсутствует разделение ответственности между компонентами;
- не обработаны типовые ошибки пользователя;
- нет понятной инструкции запуска.

2.3.8. Модуль 8. Web-основы, HTTP, JSON и клиентская часть

Цель модуля – сформировать понимание клиент-серверной архитектуры, HTTP, JSON и базовых web-технологий, необходимых для разработки REST-сервисов.

Планируемые результаты освоения модуля:

- понимать HTTP, JSON и REST-подход;
- описывать структуру API и статусы ответов;
- создавать простую клиентскую страницу для проверки backend-сервиса.

Таблица 12 – Учебно-тематический план модуля 8

№	Тема	Объем, ак. ч.
1	Клиент-серверная модель	4
2	HTTP, методы, заголовки и статус-коды	6
3	JSON и сериализация данных	5
4	Основы HTML и CSS для backend-разработчика	6
5	Основы JavaScript и взаимодействие с API	5
6	Проектирование REST API	6
7	Интеграция клиентской части и backend-сервиса	4
8	Практикум web-взаимодействия	4

Содержание модуля раскрывается через последовательное изучение тем, выполнение практических заданий и подготовку результата, который включается в индивидуальное портфолио обучающегося. Материал модуля связан с итоговым проектом: обучающийся не только выполняет отдельные упражнения, но и переносит освоенные приемы в собственное приложение.

Практическая подготовка включает:

- выполнение практикума по темам модуля 8;
- доработка учебного или итогового проекта с применением изученных инструментов;
- оформление результата в репозитории с понятной историей изменений и кратким описанием.

Самостоятельная работа включает:

- повторение теоретического материала и разбор типовых ошибок;
- самостоятельное решение дополнительных задач;
- подготовка фрагмента проекта к проверке и защите.

Промежуточная проверка результата. зачет проводится по результатам практического задания, проверки репозитория, краткого устного опроса и анализа качества выполненного кода.

Минимальный уровень освоения предполагает, что обучающийся выполняет обязательную часть задания, может запустить проект, объяснить основные решения и исправить типовые ошибки по замечаниям преподавателя. Повышенный уровень предполагает аккуратную архитектуру, обработку крайних случаев, наличие тестов, понятную документацию, расширенную функциональность и самостоятельное обоснование выбранных решений.

Типовые ошибки, учитываемые при проверке:

- код не компилируется или не запускается;
- решение не покрывает обязательные требования задания;
- отсутствует разделение ответственности между компонентами;
- не обработаны типовые ошибки пользователя;
- нет понятной инструкции запуска.

2.3.9. Модуль 9. Linux и среда исполнения Java-приложений

Цель модуля – научить обучающегося запускать, настраивать и диагностировать Java-приложения в Linux-среде и готовить их к серверной эксплуатации.

Планируемые результаты освоения модуля:

- запускать Java-приложения в Linux;
- использовать переменные окружения, логи и конфигурацию;
- подготавливать Docker-сценарий учебного приложения.

Таблица 13 – Учебно-тематический план модуля 9

№	Тема	Объем, ак. ч.
1	Командная строка Linux и файловая система	4
2	JDK, JAVA HOME и переменные окружения	4
3	Запуск Java-приложений и параметры JVM на базовом уровне	5
4	Логи, диагностика и анализ ошибок запуска	5
5	Конфигурация через переменные окружения и профили	4
6	Docker для Java-приложений	6
7	Nginx и reverse proxy на учебном уровне	4

Содержание модуля раскрывается через последовательное изучение тем, выполнение практических заданий и подготовку результата, который включается в индивидуальное портфолио обучающегося. Материал модуля связан с итоговым проектом: обучающийся не только выполняет отдельные упражнения, но и переносит освоенные приемы в собственное приложение.

Практическая подготовка включает:

- выполнение практикума по темам модуля 9;
- доработка учебного или итогового проекта с применением изученных инструментов;
- оформление результата в репозитории с понятной историей изменений и кратким описанием.

Самостоятельная работа включает:

- повторение теоретического материала и разбор типовых ошибок;
- самостоятельное решение дополнительных задач;
- подготовка фрагмента проекта к проверке и защите.

Промежуточная проверка результата. зачет проводится по результатам практического задания, проверки репозитория, краткого устного опроса и анализа качества выполненного кода.

Минимальный уровень освоения предполагает, что обучающийся выполняет обязательную часть задания, может запустить проект, объяснить основные решения и исправить типовые ошибки по замечаниям преподавателя. Повышенный уровень предполагает аккуратную архитектуру, обработку крайних случаев, наличие тестов, понятную документацию, расширенную функциональность и самостоятельное обоснование выбранных решений.

Типовые ошибки, учитываемые при проверке:

- код не компилируется или не запускается;
- решение не покрывает обязательные требования задания;
- отсутствует разделение ответственности между компонентами;
- не обработаны типовые ошибки пользователя;
- нет понятной инструкции запуска.

2.3.10. Модуль 10. PostgreSQL, JDBC, JPA и Hibernate

Цель модуля – сформировать навыки интеграции Java-приложений с PostgreSQL с использованием JDBC, JPA, Hibernate, миграций и транзакционного подхода.

Планируемые результаты освоения модуля:

- подключать Java-приложение к PostgreSQL;
- использовать JDBC, JPA и Hibernate;
- проектировать транзакции, миграции и слой доступа к данным.

Таблица 14 – Учебно-тематический план модуля 10

№	Тема	Объем, ак. ч.
1	PostgreSQL для Java-проектов	4
2	Проектирование схемы БД и ограничений	6
3	JDBC и безопасное выполнение запросов	6
4	DAO и repository-подход	5
5	JPA, сущности и связи	6
6	Hibernate и жизненный цикл сущностей	6
7	Миграции БД и версионирование схемы	5
8	Транзакции, индексы и целостность данных	5
9	Интеграционный практикум Java и PostgreSQL	5

Содержание модуля раскрывается через последовательное изучение тем, выполнение практических заданий и подготовку результата, который включается в индивидуальное портфолио обучающегося. Материал модуля связан с итоговым проектом: обучающийся не только выполняет отдельные упражнения, но и переносит освоенные приемы в собственное приложение.

Практическая подготовка включает:

- выполнение практикума по темам модуля 10;
- доработка учебного или итогового проекта с применением изученных инструментов;
- оформление результата в репозитории с понятной историей изменений и кратким описанием.

Самостоятельная работа включает:

- повторение теоретического материала и разбор типовых ошибок;
- самостоятельное решение дополнительных задач;
- подготовка фрагмента проекта к проверке и защите.

Промежуточная проверка результата. зачет проводится по результатам практического задания, проверки репозитория, краткого устного опроса и анализа качества выполненного кода.

Минимальный уровень освоения предполагает, что обучающийся выполняет обязательную часть задания, может запустить проект, объяснить основные решения и исправить типовые ошибки по замечаниям преподавателя. Повышенный уровень предполагает аккуратную архитектуру, обработку крайних случаев, наличие тестов, понятную документацию, расширенную функциональность и самостоятельное обоснование выбранных решений.

Типовые ошибки, учитываемые при проверке:

- код не компилируется или не запускается;
- решение не покрывает обязательные требования задания;
- отсутствует разделение ответственности между компонентами;
- не обработаны типовые ошибки пользователя;
- нет понятной инструкции запуска.

2.3.11. Модуль 11. JavaFX и клиентские приложения на Java

Цель модуля – сформировать навыки разработки настольных клиентских приложений на JavaFX с разделением интерфейса, бизнес-логики и слоя данных.

Планируемые результаты освоения модуля:

- создавать JavaFX-интерфейсы;
- связывать формы с сервисным слоем;
- обрабатывать события, валидацию и отображение табличных данных.

Таблица 15 – Учебно-тематический план модуля 11

№	Тема	Объем, ак. ч.
1	Назначение JavaFX и сценарии применения	4
2	Компоненты интерфейса, контейнеры и layout	6

№	Тема	Объем, ак. ч.
3	Обработка событий и пользовательский ввод	5
4	FXML, контроллеры и разделение ответственности	6
5	Табличные данные, формы и валидация	5
6	Интеграция интерфейса с сервисным слоем	5
7	Интеграция с базой данных или REST API	5
8	Практикум JavaFX-приложения	4

Содержание модуля раскрывается через последовательное изучение тем, выполнение практических заданий и подготовку результата, который включается в индивидуальное портфолио обучающегося. Материал модуля связан с итоговым проектом: обучающийся не только выполняет отдельные упражнения, но и переносит освоенные приемы в собственное приложение.

Практическая подготовка включает:

- выполнение практикума по темам модуля 11;
- доработка учебного или итогового проекта с применением изученных инструментов;
- оформление результата в репозитории с понятной историей изменений и кратким описанием.

Самостоятельная работа включает:

- повторение теоретического материала и разбор типовых ошибок;
- самостоятельное решение дополнительных задач;
- подготовка фрагмента проекта к проверке и защите.

Промежуточная проверка результата. зачет проводится по результатам практического задания, проверки репозитория, краткого устного опроса и анализа качества выполненного кода.

Минимальный уровень освоения предполагает, что обучающийся выполняет обязательную часть задания, может запустить проект, объяснить основные решения и исправить типовые ошибки по замечаниям преподавателя. Повышенный уровень предполагает аккуратную архитектуру, обработку крайних случаев, наличие тестов, понятную документацию, расширенную функциональность и самостоятельное обоснование выбранных решений.

Типовые ошибки, учитываемые при проверке:

- код не компилируется или не запускается;
- решение не покрывает обязательные требования задания;
- отсутствует разделение ответственности между компонентами;
- не обработаны типовые ошибки пользователя;
- нет понятной инструкции запуска.

2.3.12. Модуль 12. Spring Boot: REST API, безопасность, тестирование и деплой

Цель модуля – сформировать навыки разработки полноценных backend-приложений на Spring Boot с REST API, базой данных, безопасностью, тестированием и подготовкой к развертыванию.

Планируемые результаты освоения модуля:

- создавать Spring Boot REST API;
- использовать Spring Data JPA и Spring Security;
- писать модульные и интеграционные тесты, готовить сервис к деплою.

Таблица 16 – Учебно-тематический план модуля 12

№	Тема	Объем, ак. ч.
1	Spring Core и внедрение зависимостей	6
2	Spring Boot и структура проекта	6
3	REST-контроллеры и маршрутизация запросов	8
4	DTO, валидация и обработка ошибок	8
5	Сервисный слой и бизнес-логика	6
6	Spring Data JPA и работа с PostgreSQL	8
7	Spring Security, роли и базовая защита	8
8	Тестирование Spring Boot: JUnit, Mockito, интеграционные тесты	8
9	Документация API и качество backend-сервиса	4
10	Деплой, JVM-процесс, Docker, Nginx и CI/CD на базовом уровне	6
11	Итоговый backend-практикум	4

Содержание модуля раскрывается через последовательное изучение тем, выполнение практических заданий и подготовку результата, который

включается в индивидуальное портфолио обучающегося. Материал модуля связан с итоговым проектом: обучающийся не только выполняет отдельные упражнения, но и переносит освоенные приемы в собственное приложение.

Практическая подготовка включает:

- выполнение практикума по темам модуля 12;
- доработка учебного или итогового проекта с применением изученных инструментов;
- оформление результата в репозитории с понятной историей изменений и кратким описанием.

Самостоятельная работа включает:

- повторение теоретического материала и разбор типовых ошибок;
- самостоятельное решение дополнительных задач;
- подготовка фрагмента проекта к проверке и защите.

Промежуточная проверка результата. зачет проводится по результатам практического задания, проверки репозитория, краткого устного опроса и анализа качества выполненного кода.

Минимальный уровень освоения предполагает, что обучающийся выполняет обязательную часть задания, может запустить проект, объяснить основные решения и исправить типовые ошибки по замечаниям преподавателя. Повышенный уровень предполагает аккуратную архитектуру, обработку крайних случаев, наличие тестов, понятную документацию, расширенную функциональность и самостоятельное обоснование выбранных решений.

Типовые ошибки, учитываемые при проверке:

- код не компилируется или не запускается;
- решение не покрывает обязательные требования задания;
- отсутствует разделение ответственности между компонентами;
- не обработаны типовые ошибки пользователя;
- нет понятной инструкции запуска.

2.3.13. Модуль 13. Проектная практика и консультации

Цель модуля – обеспечить методическое сопровождение итогового проекта от выбора темы до предзащиты и доведения результата до работоспособного состояния.

Планируемые результаты освоения модуля:

- сформировать техническое задание итогового проекта;
- реализовать основную функциональность и подготовить демонстрацию;
- устранить замечания по коду, тестам и документации.

Таблица 17 – Учебно-тематический план модуля 13

№	Тема	Объем, ак. ч.
1	Выбор темы итогового проекта и согласование требований	4
2	Проектирование архитектуры, БД и пользовательских сценариев	4
3	Разработка основной функциональности	6
4	Тестирование, документация и подготовка демонстрации	4
5	Консультации, ревью кода и предзащита	6

Содержание модуля раскрывается через последовательное изучение тем, выполнение практических заданий и подготовку результата, который включается в индивидуальное портфолио обучающегося. Материал модуля связан с итоговым проектом: обучающийся не только выполняет отдельные упражнения, но и переносит освоенные приемы в собственное приложение.

Практическая подготовка включает:

- выполнение практикума по темам модуля 13;
- доработка учебного или итогового проекта с применением изученных инструментов;
- оформление результата в репозитории с понятной историей изменений и кратким описанием.

Самостоятельная работа включает:

- повторение теоретического материала и разбор типовых ошибок;
- самостоятельное решение дополнительных задач;
- подготовка фрагмента проекта к проверке и защите.

Промежуточная проверка результата. зачет проводится по результатам практического задания, проверки репозитория, краткого устного опроса и анализа качества выполненного кода.

Минимальный уровень освоения предполагает, что обучающийся выполняет обязательную часть задания, может запустить проект, объяснить

основные решения и исправить типовые ошибки по замечаниям преподавателя. Повышенный уровень предполагает аккуратную архитектуру, обработку крайних случаев, наличие тестов, понятную документацию, расширенную функциональность и самостоятельное обоснование выбранных решений.

Типовые ошибки, учитываемые при проверке:

- код не компилируется или не запускается;
- решение не покрывает обязательные требования задания;
- отсутствует разделение ответственности между компонентами;
- не обработаны типовые ошибки пользователя;
- нет понятной инструкции запуска.

2.3.14. Модуль 14. Промежуточная аттестация по завершении Программы

Цель модуля – оценить готовность обучающегося к выполнению учебной и проектной деятельности в выбранной технической области на основе защиты итогового проекта и анализа его инженерного качества.

Планируемые результаты освоения модуля:

- представить итоговый проект;
- обосновать архитектурные решения;
- ответить на вопросы по материалам всех модулей.

Таблица 18 – Учебно-тематический план модуля 14

№	Тема	Объем, ак. ч.
1	Подготовка итогового проекта к защите	6
2	Проверка репозитория, README, инструкции запуска и демонстрационных данных	6
3	Защита итогового проекта	8
4	Ответы на вопросы и экспертная оценка результата	4

Содержание модуля раскрывается через последовательное изучение тем, выполнение практических заданий и подготовку результата, который включается в индивидуальное портфолио обучающегося. Материал модуля связан с итоговым проектом: обучающийся не только выполняет отдельные упражнения, но и переносит освоенные приемы в собственное приложение.

Практическая подготовка включает:

- финальная проверка проекта по чек-листу;
- подготовка демонстрационного сценария и презентации;
- защита проекта и ответы на вопросы комиссии.

Самостоятельная работа включает:

- устранение замечаний предзащиты;
- подготовка README, инструкции запуска и тестовых данных;
- самопроверка по критериям промежуточной аттестации по завершении Программы.

Промежуточная проверка результата. промежуточная аттестация по завершении Программы проводится в форме защиты итогового проекта с демонстрацией работоспособности, репозитория, документации и ответами на вопросы.

Минимальный уровень освоения предполагает, что обучающийся выполняет обязательную часть задания, может запустить проект, объяснить основные решения и исправить типовые ошибки по замечаниям преподавателя. Повышенный уровень предполагает аккуратную архитектуру, обработку крайних случаев, наличие тестов, понятную документацию, расширенную функциональность и самостоятельное обоснование выбранных решений.

Типовые ошибки, учитываемые при проверке:

- проект не запускается по инструкции;
- отсутствует значимая часть заявленной функциональности;
- нет репозитория или истории разработки;
- не представлены доказательства самостоятельного выполнения;
- обучающийся не может объяснить ключевые решения.

2.4. Проектная практика и консультации

Проектная практика направлена на интеграцию результатов всех модулей в единый прикладной продукт. В ходе проектной практики обучающийся уточняет тему итогового проекта, формирует требования, проектирует архитектуру, определяет структуру базы данных, реализует основную функциональность, готовит тесты, демонстрационные данные, README и инструкцию запуска.

Консультации проводятся по вопросам выбора архитектуры, декомпозиции проекта, проектирования базы данных, настройки Spring Boot, JavaFX, JPA/Hibernate, Spring Security, Docker, тестирования и подготовки к итоговой защите. Результаты консультаций фиксируются через комментарии преподавателя, статусы заданий, историю репозитория и индивидуальные рекомендации по доработке.

3. Организационно-педагогические условия реализации программы

3.1. Общие условия реализации

Реализация Программы осуществляется с применением электронного обучения и дистанционных образовательных технологий. Образовательный процесс организуется через электронную информационно-образовательную среду, обеспечивающую доступ к учебным материалам, расписанию, заданиям, средствам обратной связи, результатам текущего контроля, промежуточной и промежуточной аттестации по завершении Программы.

3.2. Материально-технические условия

Для освоения Программы обучающемуся требуется персональный компьютер или ноутбук с устойчивым доступом к сети Интернет, возможностью установки JDK, IDE, Git, PostgreSQL, Docker и средств видеосвязи. Рекомендуемая конфигурация: не менее 8 ГБ оперативной памяти, свободное место на диске не менее 20 ГБ, современная операционная система Windows, macOS или Linux.

Организация обеспечивает доступ к учебным материалам, видеозанятиям или их записям, заданиям, репозиториям, проверочным материалам, каналам коммуникации и инструкциям по настройке среды разработки. Для выполнения практических заданий используются JDK, IntelliJ IDEA Community Edition или аналогичная IDE, Git, Maven, Gradle, PostgreSQL, JavaFX, Spring Boot, JUnit, Mockito, Docker, Nginx на учебном уровне и дополнительные библиотеки, указанные в заданиях.

3.3. Электронная информационно-образовательная среда

Электронная информационно-образовательная среда обеспечивает идентификацию обучающегося, предоставление учебных материалов, фиксацию результатов обучения, хранение выполненных заданий, доступ к обратной связи и организацию контрольных мероприятий. Идентификация и аутентификация обучающихся, порядок применения прокторинга при необходимости, фиксация прохождения аттестации и хранение результатов

определяются локальным актом организации об электронном обучении и дистанционных образовательных технологиях.

Выполненные задания хранятся в электронной среде, репозитории или ином цифровом хранилище, указанном организацией. Результаты проверок, замечания преподавателя, статусы зачета, результаты промежуточной и промежуточной аттестации по завершении Программы фиксируются в учетных системах организации.

3.4. Кадровые условия

Реализация Программы обеспечивается преподавателями и наставниками, обладающими профессиональной компетентностью в сфере программирования, Java-разработки, баз данных, web-разработки, тестирования, разработки на Spring Boot и сопровождения программных продуктов. К проверке итоговых проектов могут привлекаться специалисты-практики, имеющие опыт промышленной разработки программного обеспечения.

3.5. Условия для несовершеннолетних обучающихся

При обучении несовершеннолетних организация учитывает возрастные особенности, обеспечивает информационную безопасность и педагогически обоснованную продолжительность непрерывной работы с экранными устройствами. Расписание предусматривает перерывы и смену видов деятельности. Коммуникация, публикация проектов, использование внешних сервисов, обработка изображений, голосовых данных и иных персональных данных осуществляются с учетом согласий законных представителей и локальных правил.

Преподаватель не допускает включения в открытые репозитории ключей доступа, паролей, персональных данных и материалов, нарушающих права третьих лиц. При необходимости используются учебные учетные записи, закрытые репозитории и обезличенные наборы данных.

3.6. Условия для обучающихся с ограниченными возможностями здоровья

При наличии запроса и организационной возможности предоставляются увеличенный срок выполнения заданий, доступные форматы материалов, субтитры или текстовые конспекты, альтернативные способы представления проекта и иные специальные условия в соответствии с индивидуальными потребностями и локальными актами организации.

3.7. Учебно-методическое обеспечение

Учебно-методическое обеспечение включает рабочие программы модулей, видеоматериалы или конспекты занятий, практические задания, чек-листы, примеры кода, инструкции по настройке окружения, шаблоны README, критерии оценивания, перечень типовых ошибок, методические рекомендации по выполнению итогового проекта и оценочные материалы.

3.8. Применение электронного обучения и дистанционных образовательных технологий

Программа реализуется с применением электронного обучения и дистанционных образовательных технологий. Электронное обучение используется для предоставления учебных материалов, заданий, записей, методических рекомендаций, средств коммуникации, фиксации результатов контроля и обратной связи. Дистанционные образовательные технологии используются для организации взаимодействия обучающегося и педагогических работников на расстоянии, проведения консультаций, проверки работ, промежуточной и промежуточной аттестации по завершении Программы.

При реализации Программы организация обеспечивает идентификацию обучающегося в электронной информационно-образовательной среде, доступ к материалам в течение периода обучения, возможность сдачи работ, получение обратной связи и фиксацию результатов освоения Программы. Технические требования к устройству, программному обеспечению, каналам связи и учетным данным доводятся до обучающегося до начала обучения или при предоставлении доступа к образовательной среде.

3.9. Информационная безопасность, персональные данные и этика

При реализации Программы обеспечивается соблюдение требований законодательства о персональных данных, правил работы с учетными записями, учебными материалами, проектными файлами и результатами аттестации. Доступ к электронной информационно-образовательной среде предоставляется индивидуально и не подлежит передаче третьим лицам.

Обучающийся обязан соблюдать академическую добросовестность, не представлять чужие работы как собственные, корректно указывать источники заимствований, не нарушать права третьих лиц на программный код, данные, изображения, тексты и иные материалы. При выявлении признаков несамостоятельного выполнения работы организация вправе назначить дополнительную проверку, собеседование, повторную сдачу или иные процедуры, предусмотренные локальными нормативными актами.

3.10. Локальные процедуры обучения

Организация реализации Программы обеспечивается локальными нормативными актами ООО «Иннопрог», регулирующими прием и зачисление обучающихся, режим занятий, применение электронного обучения и дистанционных образовательных технологий, текущий контроль, текущий контроль и промежуточную аттестацию, пересдачи, апелляции, отчисление, восстановление, выдачу документов об обучении, обработку персональных данных.

Локальные процедуры применяются в части, не противоречащей законодательству Российской Федерации, условиям договора об образовании и настоящей Программе. Обучающийся информируется о значимых правилах реализации Программы до начала обучения или при предоставлении доступа к образовательной среде.

3.11. Обновление программы

Программа подлежит периодическому обновлению с учетом изменений законодательства, профессиональных стандартов, рынка труда, технологического стека Java-разработки, требований работодателей, результатов внутренней оценки качества и обратной связи обучающихся.

4. Оценка качества освоения программы

4.1. Общие положения

Оценка качества освоения Программы включает текущий контроль успеваемости, промежуточную аттестацию по модулям, промежуточную аттестацию по завершении Программы и мониторинг удовлетворенности обучающихся. Оценка проводится по критериям, позволяющим проверить достижение планируемых результатов, самостоятельность выполнения работ, работоспособность программного продукта, качество кода, архитектуры, тестов и документации.

4.2. Текущий контроль успеваемости

Текущий контроль осуществляется в форме проверки практических заданий, устных или письменных опросов, мини-проектов, задач на программирование, ревью репозиторий, тестов по теоретическому материалу, анализа коммитов и выполнения контрольных чек-листов.

При текущем контроле учитываются:

- своевременность выполнения задания;
- соответствие результата техническому заданию;
- корректность, читаемость и структурированность кода;
- способность обучающегося объяснить реализованное решение;
- наличие инструкции запуска, тестовых данных и фиксации результата в репозитории;
- соблюдение требований к самостоятельности выполнения работы.

4.3. Промежуточная аттестация

Промежуточная аттестация проводится по модулям в форме зачета. Зачет может включать практическое задание, проверку репозитория, краткий опрос, разбор ошибок и демонстрацию работоспособности результата.

Результат промежуточной аттестации фиксируется в форме «зачтено» или «не зачтено». Основанием для зачета является выполнение обязательной части задания, корректный запуск проекта, соответствие результата заявленным

требованиям, способность обучающегося объяснить решение и устранить замечания в установленные сроки.

4.4. Пересдача, апелляция и подтверждение самостоятельности

Обучающийся, получивший результат «не зачтено», вправе доработать задание и представить его на повторную проверку в порядке и сроки, установленные локальными нормативными актами организации. При несогласии с результатом проверки обучающийся может подать апелляцию в порядке, установленном локальным актом о текущем контроле, промежуточной и промежуточной аттестации по завершении Программы.

Самостоятельность выполнения работ подтверждается устной защитой, анализом истории репозитория, проверкой структуры проекта, сопоставлением решения с требованиями задания, демонстрацией работоспособности и ответами на вопросы по коду.

4.5. Допуск к промежуточной аттестации по завершении Программы

К промежуточной аттестации по завершении Программы допускаются обучающиеся, освоившие Программу, выполнившие обязательные практические задания, получившие зачеты по модулям, подготовившие итоговый проект, репозиторий, README, инструкцию запуска и демонстрационные данные.

4.6. Промежуточная аттестация по завершении Программы

Промежуточная аттестация по завершении Программы проводится в форме защиты итогового проекта. Защита включает демонстрацию работоспособности приложения, представление архитектуры, базы данных, ключевых фрагментов кода, тестов, инструкции запуска, репозитория и ответов на вопросы по материалам всех модулей.

4.6.1. Примерная тематика проектных работ

Таблица 19 – Примерная тематика итоговых проектных работ

№	Тема проекта	Ожидаемый результат
1	Консольный менеджер задач на Java	приложение для создания, редактирования, поиска, фильтрации и закрытия задач с хранением данных в файле или базе данных.
2	Система учета товаров на складе	сервис для ведения номенклатуры, остатков, цен, поступлений и списаний с отчетом по доступным позициям.
3	База сотрудников и зарплат	приложение для хранения данных сотрудников, расчета начислений, поиска, сортировки и формирования управленческого отчета.
4	Сервис бронирования поездок	прикладная система для учета клиентов, автомобилей, маршрутов, времени поездки и статуса бронирования.
5	Сервис управления пассажирскими перевозками	приложение для учета транспорта, вместимости, пассажиров, посадки, свободных мест и истории рейсов.
6	Key-Value хранилище на дереве поиска	собственная реализация структуры хранения с добавлением, поиском, обновлением, удалением и обходом данных.
7	Файловый каталог книг и читателей	консольное или JavaFX-приложение для учета книг, читателей, выдачи, возврата и просроченных записей.
8	SQL-каталог товаров	Java-приложение с PostgreSQL, CRUD-операциями, фильтрацией, сортировкой и отчетами по ассортименту.
9	JavaFX-приложение для личных финансов	настольное приложение с формами, таблицами, категориями расходов, диаграммами и сохранением данных.
10	JavaFX-система бронирования кабинетов	клиентское приложение для выбора даты, времени, помещения, ответственного лица и статуса заявки.

№	Тема проекта	Ожидаемый результат
11	REST API менеджера задач на Spring Boot	backend-сервис с задачами, пользователями, статусами, валидацией, обработкой ошибок и документацией API.
12	Сервис учета заказов с PostgreSQL	Spring Boot приложение с товарами, заказами, клиентами, транзакциями, миграциями и интеграционными тестами.
13	Интернет-магазин на Spring Boot	web/backend-проект с каталогом, корзиной, заказами, ролями пользователей, административной частью и PostgreSQL.
14	Сервис авторизации и личного кабинета	приложение с регистрацией, входом, ролями, Spring Security, защищенными маршрутами и пользовательским профилем.
15	Продакшн-версия Java backend-приложения	итоговый проект с Docker, переменными окружения, логированием, тестами, README, инструкцией деплоя и демонстрацией работоспособности.

4.6.2. Обязательные требования к итоговому проекту

Итоговый проект должен соответствовать следующим требованиям:

- иметь понятную предметную область, цель, роли пользователей и сценарии использования;
- содержать работоспособное Java-приложение, JavaFX-клиент или Spring Boot backend-сервис;
- использовать объектную модель, разделение ответственности и понятную структуру пакетов;
- при необходимости использовать PostgreSQL, JDBC, JPA, Hibernate и миграции;
- содержать обработку ошибок, валидацию данных и защиту от типовых некорректных действий пользователя;
- включать тесты для ключевых компонентов или сценариев;

- иметь репозиторий с историей разработки, README, инструкцией запуска и демонстрационными данными;
- быть подготовленным к демонстрации и устной защите.

4.6.3. Порядок проведения промежуточной аттестации по завершении Программы

Защита итогового проекта проводится комиссией или уполномоченным преподавателем. Обучающийся демонстрирует запуск проекта, основные сценарии работы, структуру репозитория, архитектуру приложения, базу данных, тесты и документацию. После демонстрации обучающийся отвечает на вопросы по коду, архитектуре, выбранным технологиям, возможным улучшениям и материалам всех модулей Программы.

4.6.4. Критерии оценивания итогового проекта

Таблица 20 – Критерии оценивания итогового проекта

Критерий	Описание	Максимум, баллов
Функциональная полнота	Реализованы заявленные сценарии, основные функции работают корректно, приложение можно использовать для решения прикладной задачи.	20
Архитектура и качество кода	Код структурирован по слоям, классы и методы имеют понятную ответственность, нет грубого дублирования, соблюдены базовые правила Java-разработки.	20
Работа с данными	Корректно реализовано хранение данных, схема БД или файловое хранение, запросы, транзакции, миграции и обработка ошибок.	15
Spring Boot, JavaFX или интеграционный стек	Применены технологии, соответствующие выбранной теме проекта: REST API, JavaFX, JPA, Spring Security, внешние API или другие компоненты.	15

Критерий	Описание	Максимум, баллов
Тестирование и надежность	Есть тесты ключевой логики, обработка некорректного ввода, логирование и проверка типовых ошибок.	10
Документация и запуск	README содержит описание, стек, инструкцию запуска, тестовые данные и сведения о структуре проекта.	10
Защита и ответы на вопросы	Обучающийся демонстрирует проект, объясняет решения, понимает код и отвечает на вопросы по материалам программы.	10

Итоговая оценка выставляется по 100-балльной шкале. Рекомендуемый минимальный порог успешного прохождения промежуточной аттестации по завершении Программы – 70 баллов при условии отсутствия критических нарушений. Критическими нарушениями признаются невозможность запуска проекта, отсутствие значимой части заявленной функциональности, отсутствие самостоятельности выполнения, отсутствие репозитория или неспособность обучающегося объяснить ключевые решения.

4.7. Внутренняя и внешняя оценка качества реализации программы

Внутренняя оценка качества реализации Программы проводится на основе анализа результатов текущего контроля, промежуточной и промежуточной аттестации по завершении Программы, обратной связи обучающихся, качества итоговых проектов, соблюдения сроков проверки и полноты методического обеспечения.

Внешняя независимая оценка качества может проводиться путем привлечения специалистов-практиков, работодателей, внешних экспертов, анализа портфолио обучающихся, завершивший Программу, экспертной оценки итоговых проектов и сопоставления результатов обучения с требованиями рынка труда. Порядок проведения внешней оценки определяется локальными нормативными актами организации.

5. Методические материалы

5.1. Методические указания для обучающихся

Обучающемуся рекомендуется изучать материалы последовательно, выполнять практические задания в срок, вести репозиторий с понятной историей изменений, фиксировать вопросы по заданию, регулярно запускать проект, проверять работу на тестовых данных и поддерживать README в актуальном состоянии. При выполнении проекта следует уделять внимание не только функциональности, но и читаемости кода, архитектуре, обработке ошибок, тестам и инструкции запуска.

5.2. Методические указания для преподавателей и проверяющих

Преподавателю рекомендуется проверять не только факт выполнения задания, но и инженерное качество результата: структуру проекта, понятность названий, корректность архитектурных решений, обработку исключений, работу с данными, тестируемость, самостоятельность выполнения и способность обучающегося объяснить код. Обратная связь должна быть конкретной, проверяемой и ориентированной на доработку результата.

5.3. Рекомендуемая литература и ресурсы

- официальная документация Java и JDK;
- официальная документация Spring Boot, Spring Framework, Spring Security и Spring Data JPA;
- официальная документация PostgreSQL;
- официальная документация JavaFX;
- официальная документация Maven и Gradle;
- официальная документация JUnit и Mockito;
- официальная документация Docker и Git;
- учебные материалы и практические задания, размещенные в электронной информационно-образовательной среде организации.

Приложение 1. Матрица формирования компетенций

Таблица 21 – Матрица формирования компетенций

№	Наименование модуля	Компетенции	Формы контроля
1	Введение в профессию и организация обучения	ПР-1–ПР-10	практические задания, репозиторий, зачет, устный опрос
2	Java Core	ПР-1	практические задания, репозиторий, зачет, устный опрос
3	Объектно-ориентированное программирование на Java	ПР-2	практические задания, репозиторий, зачет, устный опрос
4	Коллекции, обобщения, исключения и Stream API	ПР-1, ПР-2	практические задания, репозиторий, зачет, устный опрос
5	Инструменты Java-разработчика: Git, Maven, Gradle и отладка	ПР-3	практические задания, репозиторий, зачет, устный опрос
6	Основы SQL	ПР-4	практические задания, репозиторий, зачет, устный опрос
7	Алгоритмы и структуры данных	ПР-1, ПР-5	практические задания, репозиторий, зачет, устный опрос
8	Web-основы, HTTP, JSON и клиентская часть	ПР-6	практические задания, репозиторий, зачет, устный опрос
9	Linux и среда исполнения Java-приложений	ПР-7	практические задания, репозиторий, зачет, устный опрос
10	PostgreSQL, JDBC, JPA и Hibernate	ПР-4, ПР-8	практические задания, репозиторий, зачет, устный опрос
11	JavaFX и клиентские приложения на Java	ПР-9	практические задания, репозиторий, зачет, устный опрос

Продолжение таблицы 21

№	Наименование модуля	Компетенции	Формы контроля
12	Spring Boot: REST API, безопасность, тестирование и деплой	ПР-6, ПР-8, ПР-10	практические задания, репозиторий, зачет, устный опрос
13	Проектная практика и консультации	ПР-1–ПР-10	проект, ревью, предзащита
14	Промежуточная аттестация по завершении Программы	ПР-1–ПР-10	защита итогового проекта

Приложение 2. Оценочные материалы промежуточной аттестации

Промежуточная аттестация проводится в форме зачета по каждому содержательному модулю. Оценочные материалы включают практические задания, контрольные вопросы, чек-листы проверки результата и перечень критических ошибок.

Таблица 22 – Оценочные материалы промежуточной аттестации

Модуль	Проверяемый результат	Форма задания	Критерий зачета
Модуль 2	Java Core	практическое задание, проверка репозитория, краткий опрос	зачтено при выполнении обязательной части, запуске проекта и объяснении решения
Модуль 3	Объектно-ориентированное программирование на Java	практическое задание, проверка репозитория, краткий опрос	зачтено при выполнении обязательной части, запуске проекта и объяснении решения
Модуль 4	Коллекции, обобщения, исключения и Stream API	практическое задание, проверка репозитория, краткий опрос	зачтено при выполнении обязательной части, запуске проекта и объяснении решения
Модуль 5	Инструменты Java-разработчика: Git, Maven, Gradle и отладка	практическое задание, проверка репозитория, краткий опрос	зачтено при выполнении обязательной части, запуске проекта и объяснении решения
Модуль 6	Основы SQL	практическое задание, проверка репозитория, краткий опрос	зачтено при выполнении обязательной части, запуске проекта и объяснении решения
Модуль 7	Алгоритмы и структуры данных	практическое задание, проверка репозитория, краткий опрос	зачтено при выполнении обязательной части, запуске проекта и объяснении решения
Модуль 8	Web-основы, HTTP, JSON и клиентская часть	практическое задание, проверка репозитория, краткий опрос	зачтено при выполнении обязательной части, запуске проекта и объяснении решения

Модуль	Проверяемый результат	Форма задания	Критерий зачета
Модуль 9	Linux и среда исполнения Java-приложений	практическое задание, проверка репозитория, краткий опрос	зачтено при выполнении обязательной части, запуске проекта и объяснении решения
Модуль 10	PostgreSQL, JDBC, JPA и Hibernate	практическое задание, проверка репозитория, краткий опрос	зачтено при выполнении обязательной части, запуске проекта и объяснении решения
Модуль 11	JavaFX и клиентские приложения на Java	практическое задание, проверка репозитория, краткий опрос	зачтено при выполнении обязательной части, запуске проекта и объяснении решения
Модуль 12	Spring Boot: REST API, безопасность, тестирование и деплой	практическое задание, проверка репозитория, краткий опрос	зачтено при выполнении обязательной части, запуске проекта и объяснении решения

Примеры промежуточных заданий:

- реализовать консольное Java-приложение с вводом, обработкой данных и отчетом;
- спроектировать объектную модель предметной области и реализовать ее в виде набора классов;
- создать проект Maven или Gradle, оформить README и историю изменений в Git;
- спроектировать базу данных PostgreSQL и написать набор SQL-запросов;
- реализовать JavaFX-интерфейс с формой, таблицей и обработчиками событий;
- создать Spring Boot REST API с валидацией, обработкой ошибок, PostgreSQL и тестами.

Приложение 3. Оценочные материалы промежуточной аттестации по завершении Программы

Промежуточная аттестация по завершении Программы проводится в форме защиты итогового проекта. Оценка включает анализ работоспособности приложения, архитектуры, кода, базы данных, тестов, документации, истории репозитория и ответов обучающегося на вопросы.

Оценочный лист итоговой защиты.

Таблица 23 – Оценочный лист итоговой защиты

Блок оценки	Проверяемые показатели	Максимум, баллов
Функциональная полнота	Реализованы заявленные сценарии, основные функции работают корректно, приложение можно использовать для решения прикладной задачи.	20
Архитектура и качество кода	Код структурирован по слоям, классы и методы имеют понятную ответственность, нет грубого дублирования, соблюдены базовые правила Java-разработки.	20
Работа с данными	Корректно реализовано хранение данных, схема БД или файловое хранение, запросы, транзакции, миграции и обработка ошибок.	15
Spring Boot, JavaFX или интеграционный стек	Применены технологии, соответствующие выбранной теме проекта: REST API, JavaFX, JPA, Spring Security, внешние API или другие компоненты.	15
Тестирование и надежность	Есть тесты ключевой логики, обработка некорректного ввода, логирование и проверка типовых ошибок.	10
Документация и запуск	README содержит описание, стек, инструкцию запуска, тестовые данные и сведения о структуре проекта.	10

Блок оценки	Проверяемые показатели	Максимум, баллов
Защита и ответы на вопросы	Обучающийся демонстрирует проект, объясняет решения, понимает код и отвечает на вопросы по материалам программы.	10

Контрольные вопросы итоговой защиты могут включать:

- какие классы и слои приложения отвечают за ключевые сценарии;
- почему выбрана конкретная структура базы данных;
- какие ошибки пользователя обрабатываются в проекте;
- как устроены REST-маршруты, DTO, валидация и обработка ошибок;
- как применены JPA, Hibernate, транзакции и миграции;
- какие тесты реализованы и какие сценарии они проверяют;
- как запустить проект на другой машине;
- какие улучшения можно внести в следующей версии проекта.

Приложение 6. Паспорт итогового Java-проекта и требования к выпускному результату. Паспорт итогового проекта оформляется до начала активной разработки и уточняется по мере прохождения проектной практики. Паспорт нужен для фиксации требований, границ проекта, критериев приемки и состава демонстрационных материалов.

Таблица 24 – Паспорт итогового проекта

Раздел паспорта	Содержание раздела
Название проекта	краткое, отражает предметную область и основной результат
Цель проекта	какую прикладную задачу решает приложение и для кого оно создается
Пользователи и роли	какие роли существуют, какие действия выполняет каждая роль
Функциональные требования	перечень обязательных сценариев, которые должны работать на защите
Нефункциональные требования	запуск, устойчивость, читаемость, безопасность, производительность на учебном уровне

Раздел паспорта	Содержание раздела
Архитектура	слои приложения, основные пакеты, ответственность компонентов и схема взаимодействия
Модель данных	сущности, связи, ограничения, индексы, миграции и демонстрационные данные
Интерфейсы	консольный интерфейс, JavaFX-экран, REST API или их сочетание
Тестирование	какие сценарии проверяются модульными, интеграционными или ручными тестами
Развертывание	как запустить проект локально, через Docker или в Linux-среде
Документация	README, схема БД, описание API, скриншоты, сценарий демонстрации
Критерии приемки	условия, при которых проект считается готовым к итоговой защите

Минимальная конфигурация итогового проекта. Минимальная конфигурация итогового проекта включает работоспособное приложение, объектную модель, хранение данных, обработку пользовательского ввода, инструкцию запуска и демонстрационный сценарий. Для Spring Boot-проекта обязательны контроллеры, сервисный слой, DTO, валидация, обработка ошибок и подключение к базе данных или подготовленная имитация хранения данных. Для JavaFX-проекта обязательны формы, обработчики событий, отображение данных, связь с сервисным слоем и сохранение данных.

Повышенная конфигурация итогового проекта. Повышенная конфигурация включает авторизацию и роли, Spring Security, миграции базы данных, интеграционные тесты, Docker, документацию API, логирование, обработку ошибок в едином формате, настройку профилей окружения, базовый CI/CD-сценарий и расширенную пользовательскую документацию. Наличие повышенной конфигурации учитывается при выставлении максимальных баллов по критериям архитектуры, надежности, документации и готовности к сопровождению.

Требования к README итогового проекта. README итогового проекта должен позволять другому пользователю запустить приложение без устных пояснений автора. В README указываются назначение проекта, стек технологий, версия JDK, используемая СУБД, переменные окружения, инструкция локального запуска, инструкция запуска тестов, порядок создания демонстрационных данных, описание основных сценариев, структура проекта и сведения о возможных ограничениях.

Требования к демонстрации итогового проекта. Демонстрация строится по заранее подготовленному сценарию. Обучающийся показывает запуск приложения, основные пользовательские действия, работу с данными, обработку ошибки, структуру проекта, тесты, README и ключевые архитектурные решения. Демонстрация должна подтверждать, что проект представляет завершённый учебный программный продукт.

Таблица 25 – Типовые темы итоговых проектов и расширенные критерии результата

Тема	Обязательный пользовательский результат
Консольный менеджер задач на Java	пользователь может создать задачу, указать приоритет и срок, изменить статус, отфильтровать задачи, сохранить данные и получить отчет
Система учета товаров на складе	пользователь может добавлять товары, изменять остатки, фиксировать поступления и списания, искать позиции и формировать отчет
База сотрудников и зарплат	система хранит сотрудников, должности, ставки, начисления, выполняет поиск и формирует расчетный отчет
Сервис бронирования поездок	сервис хранит клиентов, автомобили, маршруты, даты, статусы и позволяет оформить или отменить бронирование
Сервис управления пассажирскими перевозками	система учитывает рейсы, транспорт, пассажиров, вместимость, посадку и доступные места

Тема	Обязательный пользовательский результат
Key-Value хранилище на дереве поиска	реализованы операции добавления, поиска, обновления, удаления, обхода и проверки корректности дерева
Файловый каталог книг и читателей	приложение ведет каталог книг, читателей, выдачи, возвраты и просроченные записи
SQL-каталог товаров	приложение использует PostgreSQL для хранения каталога, фильтров, цен, остатков и отчетов
JavaFX-приложение для личных финансов	интерфейс позволяет вести доходы, расходы, категории, таблицы, фильтры и сводку
JavaFX-система бронирования кабинетов	пользователь выбирает кабинет, дату, время, ответственного, а система проверяет конфликты
REST API менеджера задач на Spring Boot	API реализует пользователей, задачи, статусы, валидацию, обработку ошибок и документацию
Сервис учета заказов с PostgreSQL	backend хранит товары, клиентов, заказы, позиции заказа, статусы и транзакции
Интернет-магазин на Spring Boot	система включает каталог, корзину, заказы, роли, административные функции и базу данных
Сервис авторизации и личного кабинета	проект реализует регистрацию, вход, роли, защищенные маршруты и профиль пользователя
Продакшн-версия Java backend-приложения	проект включает Docker, переменные окружения, логи, тесты, README, инструкцию деплоя и демонстрационные данные

Приложение 4. Перечень локальных нормативных актов, на которые должна опираться реализация программы

Реализация Программы должна обеспечиваться локальными нормативными актами организации, включая:

- положение о разработке, утверждении и обновлении дополнительных профессиональных программ;
- положение об организации образовательного процесса по дополнительным профессиональным программам;
- положение о применении электронного обучения и дистанционных образовательных технологий;
- положение о текущем контроле, промежуточной и промежуточной аттестации по завершении Программы;
- положение о порядке приема, перевода, отчисления и восстановления обучающихся;
- положение о порядке выдачи, учета и хранения документов об обучении;
- регламент выдачи и учета сертификатов об обучении и справок об обучении;
- положение о защите персональных данных обучающихся;
- порядок рассмотрения апелляций и обращений обучающихся;
- правила внутреннего распорядка обучающихся и порядок оказания платных образовательных услуг.

Приложение 5. Лист актуализации программы

Приложение 8. Лист актуализации программы. Лист актуализации используется для фиксации изменений Программы «Java-разработчик», связанных с обновлением законодательства, профессиональных стандартов, требований к содержанию и уровню проектов, учебных материалов, оценочных средств, цифровой образовательной среды и проектной практики.

Таблица 26 – Лист актуализации программы

№	Дата	Основание изменения	Содержание изменения
1	02.08.2026	Первичное утверждение	Программа введена в действие приказом организации.
2			
3			

Приложение 6. Требования к репозиторию, README и демонстрации

Приложение 5. Детализированные чек-листы практических работ.

Детализированные чек-листы используются для единообразной проверки практических работ, промежуточных заданий и проектных результатов. Чек-лист не заменяет задание модуля, но уточняет признаки достаточного результата, требования к доказательствам выполнения и типовые основания для возврата работы на доработку.

Таблица 27 – Чек-лист проверки практических результатов по модулям

№	Модуль	Ожидаемый артефакт	Проверяемые признаки	Критические ошибки
1	Введение в профессию и организация обучения	готовность среды, репозитория и индивидуального плана	проверка доступа, структуры репозитория, выбранной траектории и понимания правил аттестации	не настроен JDK, не создан репозиторий, не выбран проектный контур
2	Java Core	консольная программа с вводом, обработкой данных, методами и отчетом	компиляция, запуск, корректность логики, обработка ввода, декомпозиция на методы	код не компилируется, нет методов, не обработаны пустые или некорректные данные
3	Объектно-ориентированное программирование на Java	объектная модель предметной области с классами, интерфейсами и сервисным слоем	соответствие модели предметной области, инкапсуляция, разделение ответственности, объяснение связей	все реализовано в одном классе, нарушена инкапсуляция, наследование применено без смысла
4	Коллекции, обобщения, исключения и Stream API	приложение с коллекциями, сортировкой, фильтрацией, файловым хранением и обработкой ошибок	корректный выбор коллекций, типобезопасность, устойчивость к ошибкам, читаемость Stream API	используются сырые типы, исключения подавляются, коллекции выбраны случайно

№	Модуль	Ожидаемый артефакт	Проверяемые признаки	Критические ошибки
5	Инструменты Java-разработчика	структурированный Maven или Gradle проект с Git-историей и инструкцией запуска	история коммитов, структура проекта, зависимости, сборка, README и базовая отладка	нет сборочного файла, зависимости добавлены вручную, репозиторий не отражает ход работы
6	Основы SQL	схема БД, набор SQL-запросов и демонстрационные данные	наличие связей, первичных и внешних ключей, выборки с фильтрами, JOIN и агрегатами	таблицы не связаны, запросы не проверены, отсутствуют ограничения целостности
7	Алгоритмы и структуры данных	набор алгоритмических решений и собственная структура данных	оценка сложности, корректность на крайних случаях, тестовые наборы, объяснение выбора структуры	решение работает только на одном примере, сложность не объяснена, структура данных нарушает инварианты
8	Web-основы, HTTP, JSON и клиентская часть	описание REST API и простая клиентская проверка обмена данными	корректность HTTP-методов, статусов, JSON-структуры, ошибок и документации сценариев	методы HTTP выбраны случайно, статусы не отражают результат, JSON невалиден
9	Linux и среда исполнения Java-приложений	инструкция запуска Java-приложения в Linux и Docker-сценарий	запуск через командную строку, переменные окружения, логи, Dockerfile или compose-файл	запуск зависит от IDE, секреты защиты в коде, нет логов и инструкции

№	Модуль	Ожидаемый артефакт	Проверяемые признаки	Критические ошибки
10	PostgreSQL, JDBC, JPA и Hibernate	Java-приложение с PostgreSQL, сущностями, репозиториями и транзакциями	корректность схемы, ORM-модели, запросов, миграций, транзакций и слоя доступа к данным	SQL и ORM смешаны хаотично, связи описаны неверно, транзакции отсутствуют
11	JavaFX и клиентские приложения на Java	настольное приложение с формами, таблицами, навигацией и сервисным слоем	структура FXML/контроллеров, обработка событий, валидация, связь с данными	вся логика в контроллере, интерфейс зависит, данные не проверяются
12	Spring Boot	REST API с сервисным слоем, PostgreSQL, безопасностью, тестами и документацией	слой контроллеров, DTO, валидация, обработка ошибок, Spring Data JPA, Spring Security, тесты и запуск	контроллер содержит всю логику, нет валидации, ошибки возвращаются хаотично, тесты отсутствуют
13	Проектная практика	готовый итоговый проект, прошедший предзащиту	реализация обязательных сценариев, качество кода, документация, демонстрация и устранение замечаний	проект не доведен до работоспособного состояния, замечания не исправлены
14	Промежуточная аттестация по завершении Программы	защита итогового проекта и ответы на вопросы	демонстрация, репозиторий, архитектура, тесты, документация, самостоятельность и объяснение решений	проект не запускается, отсутствует репозиторий, обучающийся не понимает собственный код

Единые требования к репозиторию практических работ. Каждая практическая работа должна размещаться в репозитории или в ином цифровом

хранилище, указанном преподавателем. Для работ, связанных с кодом, обязательны понятная структура каталогов, файл README, инструкция запуска, описание используемой версии JDK, перечень зависимостей, тестовые данные и комментарии к нестандартным решениям.

Рекомендуемая структура репозитория:

- каталог `src/main/java` для исходного кода;
- каталог `src/test/java` для тестов;
- файл `pom.xml` или `build.gradle`;
- файл `README.md` с назначением проекта, стеком, инструкцией запуска и примерами сценариев;
- каталог `docs` для схем, описания API, скриншотов, SQL-скриптов и пояснительных материалов;
- каталог `db` или `migrations` для схемы базы данных и миграций, если проект использует PostgreSQL;
- файл с демонстрационными данными или инструкция по их созданию.

Единые требования к коду. Код должен быть читаемым, структурированным и пригодным для сопровождения. Проверяющий оценивает осмысленность названий, отсутствие чрезмерно длинных методов, разделение ответственности, обработку исключений, отсутствие дублирования, корректность работы с ресурсами, отсутствие секретов в коде, единый стиль форматирования и способность обучающегося объяснить каждую существенную часть реализации.

Единые требования к защите промежуточной работы. При защите промежуточной работы обучающийся демонстрирует запуск проекта, показывает ключевые классы и методы, объясняет структуру данных, отвечает на вопросы по выбранным решениям и показывает, как проверялась корректность результата. При необходимости преподаватель может предложить небольшое изменение условия, чтобы убедиться в самостоятельности выполнения и понимании кода.

Приложение 7. Рубрикатор типовых ошибок и рекомендации по доработке.

Рубрикатор используется для единообразной обратной связи по практическим работам и итоговым проектам. Он помогает отделить критические ошибки,

препятствующие зачету, от замечаний, которые снижают качество результата, но могут быть устранены в ходе доработки.

Таблица 28 – Рубрикатор типовых ошибок

Блок проверки	Типовая проблема	Рекомендация по доработке
Компиляция и запуск	проект не собирается, не указана версия JDK, отсутствуют зависимости, запуск возможен только в IDE автора	добавить сборочный файл, проверить чистый запуск, описать команды сборки и запуска в README
Архитектура	контроллеры или классы содержат всю логику, нет сервисного слоя, пакеты названы случайно	разделить приложение на слои, выделить доменные сущности, сервисы, репозитории и интерфейсный слой
Объектная модель	классы не отражают предметную область, поля открыты, методы нарушают инварианты	уточнить сущности, закрыть состояние, добавить конструкторы, методы изменения и проверки корректности
Работа с данными	запросы формируются небезопасно, схема БД не имеет связей и ограничений, миграции отсутствуют	нормализовать схему, добавить ключи, ограничения, подготовить миграции и тестовые данные
REST API	методы HTTP выбраны неверно, статусы не отражают результат, ошибки возвращаются в произвольном виде	описать ресурсы, DTO, статусы, формат ошибки и правила валидации
Spring Security	доступ к защищенным операциям открыт, роли не проверяются, секреты хранятся в коде	вынести секреты в переменные окружения, описать роли, закрыть маршруты, проверить сценарии доступа
JavaFX	интерфейс связан напрямую с БД, контроллер перегружен логикой, ввод не проверяется	вынести бизнес-логику в сервисы, добавить валидацию, обработку ошибок и обновление таблиц

Блок проверки	Типовая проблема	Рекомендация по доработке
Тестирование	тесты отсутствуют или проверяют только успешный запуск без логики	добавить тесты сервисов, граничных случаев, ошибок валидации и интеграции с хранилищем
Документация	README не позволяет запустить проект, нет описания сценариев и демонстрационных данных	добавить назначение, стек, команды запуска, переменные окружения, тестовые данные и сценарий демонстрации
Самостоятельность	обучающийся не может объяснить код, коммиты отсутствуют, проект выглядит как единая загрузка готовых файлов	провести устную защиту, запросить доработку, уточнить историю изменений и потребовать объяснение решений

Рекомендации по доработке проекта после предзащиты. После предзащиты обучающийся получает перечень замечаний. Замечания рекомендуется разделить на критические, существенные и улучшения. Критические замечания устраняются в первую очередь, поскольку они связаны с запуском, целостностью функциональности, сохранением данных, безопасностью, самостоятельностью и возможностью проверить результат. Существенные замечания связаны с архитектурой, тестами, документацией и обработкой ошибок. Улучшения могут касаться расширения интерфейса, дополнительных фильтров, оптимизации запросов, улучшения пользовательского опыта и оформления демонстрации.

Рекомендации по подготовке к устной защите. Перед защитой обучающемуся рекомендуется подготовить краткое описание предметной области, перечень реализованных сценариев, схему архитектуры, схему базы данных, список ключевых классов и компонентов, перечень тестов и инструкцию запуска. Отдельно следует подготовить ответы на вопросы о выборе технологий, структуре пакетов, обработке ошибок, безопасности, тестировании, ограничениях проекта и возможных направлениях развития.

Основания для направления проекта на повторную доработку. Проект направляется на повторную доработку, если он не запускается по инструкции,

не содержит обязательной функциональности, не имеет репозитория или документации, использует данные без понятного хранения, не позволяет проверить самостоятельность выполнения, содержит грубые нарушения безопасности или обучающийся не может объяснить основные решения. Повторная проверка проводится после устранения замечаний в сроки, установленные локальными нормативными актами организации.

Приложение 9. Требования к репозиторию, README и демонстрации.

Для подтверждения освоения Программы и выполнения итогового проекта обучающийся представляет проектные материалы в форме, позволяющей проверить самостоятельность, воспроизводимость и качество результата.

Минимальный состав проектных материалов:

- репозиторий или архив проекта с исходным кодом и структурой каталогов;
- файл README.md или иной документ с описанием цели, функций, структуры, установки и запуска проекта;
- описание используемых технологий, зависимостей, версии языка/фреймворка/инструментов;
- инструкции по подготовке окружения, настройке переменных, запуску приложения, тестов или демонстрационного сценария;
- демонстрационные данные, скриншоты, ссылка на развернутую версию или видеодемонстрация при наличии такой возможности;
- краткое описание ограничений, известных проблем и возможных направлений развития проекта.

Материалы должны быть достаточны для повторной проверки проекта преподавателем, экспертом или членом аттестационной комиссии без дополнительных устных пояснений, за исключением вопросов по защите.

Приложение 7. Дорожная карта подготовки итогового проекта

Приложение 10. Дорожная карта подготовки итогового проекта. Дорожная карта используется как методический ориентир для поэтапной подготовки итогового аттестационного проекта по направлению «Java-разработчик».

1. Выбор темы и согласование прикладной задачи с преподавателем или куратором.
2. Формулирование цели проекта, целевой аудитории, пользовательских сценариев и критериев приемки.
3. Подготовка структуры проекта, репозитория, базовой документации и плана реализации.
4. Реализация минимально работоспособной версии и фиксация результата в репозитории.
5. Расширение функциональности, обработка ошибок, подготовка данных, интерфейса или демонстрационного сценария в соответствии со спецификой направления.
6. Проверка качества, тестирование, рефакторинг, оформление инструкций по запуску и использованию.
7. Предзащита или консультационная проверка, устранение замечаний, подготовка презентации и демонстрации.
8. Итоговая защита проекта с ответами на вопросы комиссии и фиксацией результата промежуточной аттестации по завершении Программы.

Приложение 8. Контрольный лист соответствия требованиям программы

Приложение 11. Контрольный лист соответствия требованиям программы.

Контрольный лист применяется для внутренней проверки полноты Программы и готовности ее реализации.

Таблица 29 – Контрольный лист соответствия требованиям программы

№	Проверяемый элемент	Статус	Комментарий
1	Указаны вид, направленность и уровень программы, трудоемкость, форма обучения, язык и срок освоения.	выполнено	
2	Представлены учебный план, календарный учебный график и рабочие программы модулей.	выполнено	
3	Описаны планируемые результаты, предметные результаты и связь с практическими ориентирами.	выполнено	
4	Определены организационно-педагогические условия, ЭИОС, ЭО и ДОТ, кадровые и методические условия.	выполнено	
5	Описаны текущий контроль, промежуточная аттестация, промежуточная аттестация по завершении Программы, пересдача и апелляция.	выполнено	

Продолжение таблицы 29

№	Проверяемый элемент	Статус	Комментарий
6	Определены документы по итогам обучения и перечень локальных актов, обеспечивающих реализацию.	выполнено	