

Министерство науки и высшего образования Российской Федерации
федеральное государственное автономное образовательное учреждение
высшего образования
«Московский физико-технический институт
(национальный исследовательский университет)»
(МФТИ, Физтех)

УТВЕРЖДАЮ

Ректор МФТИ

д-р физ.-мат. наук, профессор



Д.В. Ливанов

« 4 »

сентября 2025 г.

**Дополнительная профессиональная программа
профессиональной переподготовки
«Программирование на Java: создание современных веб-
приложений»**

для приоритетной отрасли экономики «Экономика, финансы и управление»

Москва 2025

Лист согласования

Лукичев
Андрей
Владимирович

Директор программ
Департамента по развитию
стратегических проектов
ООО «Ростелеком
Информационные
Технологии»

ФИО

Должность, наименование компании
(организации)



Лист согласования

Апурин Николай
Андреевич

ФИО

Генеральный директор
ООО «АРТВЕЛЛ»

Должность, наименование компании
(организации)

08.08.2025

Дата, подпись, печать



Лист согласования

Уткин В.Г.

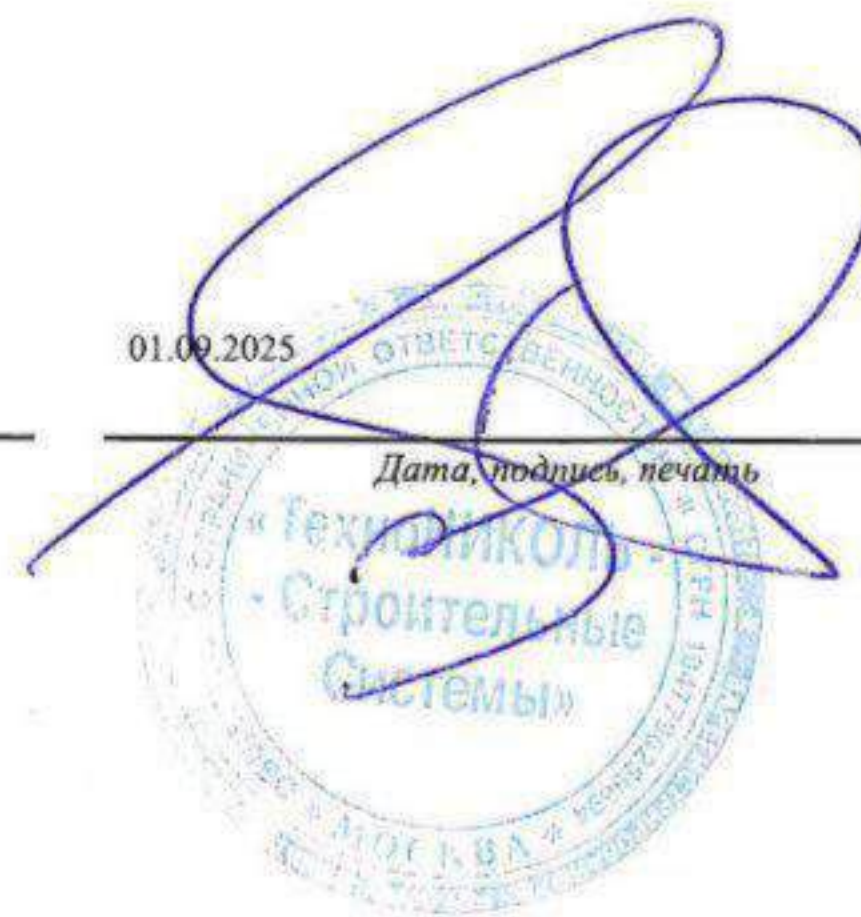
Директор по ИТ ООО «ТехноНИКОЛЬ-Строительные Системы»

01.09.2025

ФИО

Должность, наименование компании (организации)

Дата, подпись, печать



Аннотация

Дополнительная профессиональная программа профессиональной переподготовки ИТ-профиля (далее – Программа) предназначена для лиц, освоивших ОП ВО бакалавриата – в объеме не менее первого курса (бакалавры 2-го курса), освоивших ОП ВО специалитета – не менее первого курса (специалисты 2-го курса), учащихся ОП ВО магистратур не ИТ-профиля.

Целью профессиональной переподготовки является получение актуальной для приоритетной отрасли экономики «Экономика, финансы и управление» дополнительной ИТ-квалификации «Программист».

Все модули программы предусматривают выполнение обучающимися проектов, направленных на разработку концепций и/или прототипов новых цифровых решений для приоритетной отрасли экономики «Экономика, финансы и управление».

Программа является стратегически важной для отрасли «Экономика, финансы и управление» в контексте цифровой трансформации бизнеса, финансового сектора и государственного управления. Ее соответствие данной отрасли проявляется через следующие ключевые аспекты:

1. Разработка корпоративных банковских систем.
2. Создание платформ для электронной коммерции.
3. Автоматизация бизнес-процессов.
4. Финтех и инновационные финансовые решения.
5. Государственное управление и регулирование.

Программа «Программирование на Java: создание современных веб-приложений» является фундаментальной для цифровой экономики, поскольку:

- обеспечивает технологическую основу для финансового сектора;
- поддерживает процессы цифровизации бизнеса и государственного управления;
- способствует развитию инновационных финансовых технологий;
- готовит специалистов для высокотехнологичных секторов экономики.

Нормативный срок освоения программы 270 часов при сетевой очной форме обучения (с применением электронного обучения, дистанционных образовательных технологий).

Авторы и преподаватели:

№ п/п	ФИО, должность, ученая степень (при наличии), звание (при наличии), должность, место работы,	Модули (темы, лекции)	Часов, всего
1	Апурин Николай Андреевич генеральный директор ООО «АРТВЕЛЛ»	Работа с фреймворками (Spring) Java Core Работа с базами данных Работа на сервере	37 10 20 30
2	Вареница Михаил Android разработчик, Ozon Tech	Работа с фреймворками (Spring)	30
3	Дайняк Александр Борисович доцент кафедры информационных технологий в авиации, доцент кафедра дискретной математики МФТИ	Java Core	20

4	Куртасов Андрей Михайлович системный аналитик, ГК «Цифровые привычки»	Основы программирования на Java	30
5	Максимова Дарья Максимовна инженер по тестированию, Toloka.ai	Работа с базами данных	35
6	Яценко Дмитрий Владимирович ООО «ЦОА», главный инженер учебного центра, преподаватель Центра "Пуск" МФТИ	Работа на сервере	40

I. Общие положения

I.1. Нормативная правовая основа Программы

Федеральный закон от 29 декабря 2012 г. № 273-ФЗ «Об образовании в Российской Федерации»;

постановление Правительства РФ от 11 октября 2023 г. № 1678 «Об утверждении Правил применения организациями, осуществляющими образовательную деятельность, электронного обучения, дистанционных образовательных технологий при реализации образовательных программ»;

приказ Минобрнауки России от 24 марта 2025 г. № 266 «Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным профессиональным программам»;

приказ Минобрнауки России от 19 октября 2020 г. № 1316 «Об утверждении порядка разработки дополнительных профессиональных программ, содержащих сведения, составляющие государственную тайну, и дополнительных профессиональных программ в области информационной безопасности» (далее – приказ Минобрнауки России № 1316);

методические рекомендации по разработке основных профессиональных образовательных программ и дополнительных профессиональных программ с учетом соответствующих профессиональных стандартов (утв. Минобрнауки России 22 января 2015 г. № ДЛ-1/05вн);

федеральный государственный образовательный стандарт 09.03.01 Информатика и вычислительная техника (далее вместе – ФГОС ВО);

собственный образовательный стандарт МФТИ 09.03.01 Информатика и вычислительная техника (далее вместе – СОС ВО МФТИ);

профессиональный стандарт 06.001 «Программист»;

профессиональный стандарт 06.035 «Разработчик Web и мультимедийных приложений».

I.2. Термины и определения, используемые в Программе

Итоговая аттестация (аттестация) – оценка степени и уровня освоения обучающимися ДПП ПП или ИТ-модуля в формате демонстрационного экзамена, предусматривающая выполнение обучающимся профессиональных задач и оценку результатов и (или) процесса выполнения – проверку сформированности цифровых компетенций в ходе обучения по ДПП ПП или ИТ-модулям.

Демонстрационный экзамен – аттестационное испытание, предусматривающее выполнение профессиональных задач и оценку результатов и (или) процесса выполнения профессиональных задач для подтверждения применения обучающимися цифровых компетенций на практике.

Дистанционные образовательные технологии – это образовательные технологии, реализуемые в основном с применением информационно-телекоммуникационных сетей

при опосредованном (на расстоянии) взаимодействии обучающихся и педагогических работников.

Дополнительная профессиональная программа профессиональной переподготовки (Программа) – комплекс основных характеристик образования (объем, содержание, планируемые результаты) и организационно-педагогических условий, который представлен в виде учебного плана, календарного учебного графика, рабочих программ учебных курсов, дисциплин (модулей), оценочных и методических материалов, а также программ учебной и производственной практик, стажировок и форм аттестации, иных компонентов и обеспечивает приобретение дополнительной квалификации. Программа может разрабатываться с учетом положений профессиональных стандартов, федеральных государственных образовательных стандартов, требований рынка труда (индустрии).

Знание (З) – информация о свойствах объектов, закономерностях процессов и явлений, правилах использования этой информации для принятия решений, присвоенная обучающимся на одном из уровней, позволяющих выполнять над ней мыслительные операции.

Матрица компетенций – матрица компетенций, актуальных для цифровой экономики, с приоритетом компетенций в ИТ-сфере, разработанная Университетом Иннополис при участии ИТ-компаний и университетов-участников программы «Приоритет-2030», представляющая собой перечень компетенций, структурированный по сферам применения, типу компетенций, уровням их сформированности и характеристикам.

Междисциплинарный курс (МДК) – структурный элемент Программы или программы профессионального модуля, предназначенный для формирования знаний и умений, объединенных по прагматическим основаниям с нарушением академических границ отраслей знаний.

Опыт практической деятельности (ОПД) – образовательный результат, включающий выполнение обучающимся деятельности, завершающейся получением результата / продукта (элемента продукта), значимого при выполнении трудовой функции, в условиях реального производства или в модельной ситуации.

Оценочные средства (ОС) – дидактические средства для оценки качества подготовленности обучающихся.

Практика (практическая подготовка) – форма организации образовательной деятельности при освоении образовательной программы в условиях выполнения обучающимися определенных видов работ, связанных с будущей профессиональной деятельностью и направленных на формирование, закрепление, развитие практических навыков и компетенции по профилю соответствующей образовательной программы.

Профессиональный модуль (ПМ) – структурный элемент Программы, предназначенный для формирования определенной компетенции или нескольких компетенций.

Рабочая программа – нормативный документ в составе Программы, регламентирующий взаимодействие преподавателя и обучающихся в ходе учебного процесса при реализации структурных элементов Программы (модуль, дисциплина, курс).

Стажировка – формирование и закрепление полученных в результате теоретической подготовки профессиональных знаний и умений в рамках выполнения практических заданий (функций) на базе профильной компании (организации). Допускается заключение срочных трудовых договоров, предусматривающих прохождение обучающимся оплачиваемой стажировки. Время прохождения стажировки целесообразно учитывать в качестве учебной или производственной практики.

Умение (У) – освоенный субъектом способ выполнения действия, обеспечиваемый совокупностью приобретенных знаний и навыков; операция (действие), выполняемая определенным способом и с определенным качеством (умение, выполнение которого доведено до автоматизма, является навыком).

Учебная дисциплина (УД) – структурный элемент Программы, предназначенный для формирования знаний и умений в соответствующей сфере профессиональной деятельности.

Фонды оценочных средств (ФОС) – совокупность оценочных средств, используемых на различных этапах педагогической диагностики.

Хакатон — это командное соревнование среди студентов цифровой кафедры, нацеленное на быстрое создание инновационных проектов и решений при поддержке индустриальных партнеров.

Целевой уровень сформированности компетенции – определенный в соответствии с Матрицей цифровых компетенций и указанный в ДПП ПП и ИТ-модулях в качестве планируемого результата обучения уровень сформированности цифровой компетенции. Цифровая компетенция (компетенция) – образовательный результат, формируемый при освоении ДПП ПП или ИТ-модулей и необходимый для приобретения дополнительной *ИТ-квалификации*, необходимой для выполнения нового вида деятельности по внедрению и (или) развитию, и (или) разработке цифровых технологий, в том числе алгоритмов и программ, пригодных для практического применения, в одной из приоритетных отраслей экономики.

Электронное обучение – организация образовательной деятельности с применением содержащейся в базах данных и используемой при реализации образовательных программ информации и обеспечивающих ее обработку информационных технологий, технических средств, а также информационно-телекоммуникационных сетей, обеспечивающих передачу по линиям связи указанной информации, взаимодействие обучающихся и педагогических работников.

I.3. Требования к поступающим

К обучению по Программе допускаются обучающиеся по очной или по очно-заочной форме за счет бюджетных средств или по договорам об оказании платных образовательных услуг, освоившие программы бакалавриата в объеме не менее 1 курса (бакалавры 2 курса) и магистратуры (магистры) по направлениям/специальностям:

- 01.04.02 Прикладная математика и информатика
- 03.03.02 Физика
- 03.03.03 Радиофизика
- 03.03.01 Прикладные математика и физика
- 03.04.01 Прикладные математика и физика
- 03.04.02 Физика
- 05.03.02 География
- 05.03.06 Экология и природопользование
- 06.04.01 Биология
- 09.04.01 Информатика и вычислительная техника
- 11.04.02 Инфокоммуникационные технологии и системы связи
- 11.04.04 Электроника и наноэлектроника
- 12.04.03 Фотоника и оптоинформатика
- 14.04.02 Ядерная физика и технологии
- 16.04.01 Техническая физика
- 18.03.01 Химическая технология
- 19.04.01 Биотехнология
- 19.03.01 Биотехнология
- 22.04.01 Материаловедение и технологии материалов
- 24.04.03 Баллистика и гидроаэродинамика

- 27.04.03 Системный анализ и управление
- 27.04.06 Организация и управление наукоемкими производствами
- 27.04.07 Наукоемкие технологии и экономика инноваций
- 44.04.01 Педагогическое образование
- 44.04.04 Профессиональное обучение (по отраслям)

в приоритетной отрасли экономики «Экономика, финансы и управление».

I.4. Квалификационная характеристика выпускника

Выпускникам Программы присваивается дополнительная ИТ-квалификация в областях разработки и отладки программного кода, технической поддержки процессов создания (модификации) и сопровождения информационных ресурсов в отрасли «Экономика, финансы и управление».

Выпускник Программы будет готов к выполнению трудовой деятельности в качестве программиста.

Квалификационный уровень по национальной рамке квалификаций: А, В.

II. Планируемые результаты обучения и структура Программы

Получение дополнительной ИТ-квалификации «Специалист по аналитике управления и организации» обеспечивается формированием приведенных в таблице цифровых компетенций:

Наименование сферы	ID и наименование компетенции	Инструменты профессиональной деятельности	Целевой уровень формирования компетенций в Программе			
			Минимальный (исходный)	Базовый	Продвинутый	Экспертный
Средства программной разработки	28 Применяет языки программирования для решения профессиональных задач	Яндекс Telegram VK Мессенджер Java JavaScript			Самостоятельно применяет языки программирования и настраиваемые программные инструменты для автоматизации процессов в профессиональной деятельности	
Java (версия 11)	333 Применяет основные концепции и инструменты языка Java, использует потоки ввода-вывода				Работает с основными типами данных и конструкциями языка, использует пакет java.io при обработке потоков ввода-вывода, строит линейные алгоритмы исполнения при выполнении	

					стандартных профессиональных задач	
Java (версия 11)	335 Использует коллекции, стримы (массивы, коллекции, stream api) языка Java				Использует массивы и коллекции языка Java в соответствии с общепринятыми практиками, использует стримы (stream api) для обработки коллекций	
Средства программной разработки	32 Использует СУБД при разработке ПО	Яндекс Telegram VK Мессенджер SQL		Знает основы баз данных, знаком с нормализацией, ACID, транзакциями, может написать простые выборки («select'ы»). Участвует в проектах по созданию ПО с использованием СУБД под контролем опытных специалистов		

II.1. Структура образовательных результатов

Формирование цифровых компетенций, необходимых для получения обучающимися дополнительной ИТ-квалификации, обеспечивается последовательным формированием промежуточных образовательных результатов, начиная со знаний.

ID и формулировка целевого уровня формирования компетенций	Промежуточные образовательные результаты		
	Опыт практической деятельности (ОПД)	Умения (У)	Знания (З)
28 Применяет языки программирования для решения профессиональных задач Самостоятельно применяет языки программирования и настраиваемые программные инструменты для автоматизации процессов в профессиональной деятельности	ОПД 1. Формирование базового понимания принципов работы языка, изучение его синтаксических конструкций, управляющих операторов, методов и массивов	У2. писать и запускать элементарные программы У3. определять круг задач и выбирать способы их решения	32. как работать с переменными, типами данных, операторами 33. основные этапы разработки программного продукта: анализ требований, проектирование, разработка, тестирование, развертывание и поддержка
	ОПД 2. Углубленное изучение ключевых возможностей языка Java, необходимых для разработки современных программных решений		34. принципы работы с переменными, типами данных, операторами 35. принципы выбора технологий и инструментов для разработки с учетом целей проекта 37. принципы и алгоритмы работы с коллекциями
333 Применяет основные концепции и инструменты языка Java, использует потоки ввода-вывода Работает с основными типами данных и	ОПД 1. Формирование базового понимания принципов работы языка, изучение его синтаксических конструкций, управляющих	У1. устанавливать и настраивать JDK, IDE и переменные среды	31. принципы работы среды разработки

конструкциями языка, использует пакет java.io при обработке потоков ввода-вывода, строит линейные алгоритмы исполнения при выполнении стандартных профессиональных задач	операторов, методов и массивов		
	ОПД 2. Углубленное изучение ключевых возможностей языка Java, необходимых для разработки современных программных решений	У4. применять основные концепции алгоритмов и структур данных при разработке программ на Java У5. Создавать, запускать и отлаживать Java-программы в командной строке и IDE У6. устанавливать JDK, настраивать IDE	36. как устанавливать и настраивать JDK и среду разработки (IntelliJ IDEA, Eclipse, NetBeans), компилировать и запускать Java-программы через командную строку и IDE.
	ОПД 4. Основы клиент-серверной архитектуры, принципов работы HTTP, а также основ HTML и CSS	У11. создавать и использовать методы, передавать параметры У12. анализировать стек вызовов и выявлять ошибки в логике программы У13. разрабатывать веб-приложения на Java (сервлеты, JSP, JavaBeans) У14. использовать наследование, инкапсуляцию, полиморфизм	313. основные методологии разработки ПО 314. многопоточность, синхронизацию потоков 315. основы объектно-ориентированного программирования (ООП)
335 Использует коллекции, стримы (массивы, коллекции, stream api) языка Java	ОПД 4. Основы клиент-серверной архитектуры, принципов работы HTTP, а также основ HTML и CSS	У15. создавать RESTful API, обрабатывать HTTP-запросы У16. применять паттерны проектирования (DAO, MVC)	316. принципы разработки RESTful API
Использует массивы и коллекции языка Java в соответствии с общепринятыми практиками, использует стримы (stream api) для обработки коллекций	ОПД 5. Основы Spring Framework и его ключевых компонентов, включая Spring Core и принципы работы с бинами и контекстами	У17. применять ООП (классы, объекты, конструкторы, this) У18. работать с массивами и коллекциями У19. использовать средства контроля версий У22. использовать Spring Framework, Spring Boot для создания приложений	317. алгоритмы поиска и устранения ошибок в коде 320. применение шаблонов проектирования (DAO, MVC)
32 Использует СУБД при разработке ПО	ОПД 3. Изучение основ реляционных	У7. использовать управляющие конструкции (if-else, switch, циклы)	38. основные структуры данных и алгоритмы

Знает основы баз данных, знаком с нормализацией, ACID, транзакциями, может написать простые выборки («select'ы»). Участвует в проектах по созданию ПО с использованием СУБД под контролем опытных специалистов	баз данных и языка SQL	У8. использовать объектно-ориентированное программирование (ООП) для создания масштабируемых программных решений У9. работать с многопоточным программированием У10. выполнять операции чтения и записи файлов У23. распознавать потенциальные угрозы безопасности и утечки информации в цифровых системах и программных продуктах У24. подбирать и применять базовые меры защиты информации в контексте разработки и эксплуатации цифровых решений	39. принципы построения и функционирования операционных систем и сетей 310. основы архитектуры ПО 311. механизмы обработки исключений и отладки в Java 312. принципы клиент-серверной архитектуры и HTTP-протокола 321. Основные понятия информационной безопасности, включая конфиденциальность, угрозы, уязвимости и каналы утечки информации 322. Типы технических каналов утечки информации и принципы их функционирования 323. Базовые методы и средства технической защиты информации, включая активные и пассивные меры
	ОПД 5. Основы Spring Framework и его ключевых компонентов, включая Spring Core и принципы работы с бинами и контекстами	У20. работать с JDBC и выполнять CRUD-операции с БД У21. подключаться к базам данных через ORM (Hibernate, JPA)	318. основы реляционных баз данных и SQL 319. принципы работы с JDBC и ORM (Hibernate, JPA)

II.2. Структура Программы

Структура Программы регулирует образовательные траектории обучающихся, последовательность освоения структурных элементов (разделов) Программы, соответственно, последовательность формирования всех образовательных результатов.

Структурные элементы (разделы Программы)	Шифры образовательных результатов	Вариатив / инвариант и целевые группы обучающихся
1. Основы программирования на Java	Компетенции 28, 333	Инвариант для всех групп обучающихся
	У1, У2, У3. 31, 32, 33.	
2. Java Core	Компетенции 28, 333	Инвариант для всех групп обучающихся
	У4, У5, У6. 34, 35, 36, 37.	
3. Работа с базами данных	Компетенция 32	Инвариант для всех групп обучающихся
	У7, У8, У9, У10, У23, У24 38, 39, 310, 311, 312.	
4. Работа на сервере	Компетенции 333, 335	Инвариант для всех групп обучающихся
	У11, У12, У13, У14, У15, У16 313, 314, 315, 316, 321, 322, 323	
5. Работа с фреймворками (Spring)	Компетенции 335, 32	Инвариант для всех групп обучающихся
	У17, У18, У19, У20, У21, У22. 317, 318, 319, 320.	
6. Производственная практика ¹	Компетенции 28, 333, 32, 335	Инвариант для всех групп обучающихся
	У1, У6, У2, У7, У11, У17, У18, У10, У13, У4, У8, У3, У9, У12, У19 31, 32, 312, 38, 33, 35, 310, 311, 314, 317	
Итоговая аттестация	Компетенции 28, 32, 333, 335	Инвариант для всех групп обучающихся
	У3, У4, У13, У16, У8 33, 35	

¹ Производственная практика может проходить в формате стажировки

III. Учебный план программы

Объем Программы составляет 270 академических часов.

Учебный план Программы определяет перечень, последовательность, общую трудоемкость разделов и формы контроля знаний

Структурные элементы (разделы Программы)	Общая трудоемкость, часов	Обязательная аудиторная учебная нагрузка		Самостоятельная работа, часов		Практики, стажировки, часов	Промежуточная аттестация, часов
		всего, часов	в т.ч. практические занятия, часов	всего, часов	в т.ч. практические задания, часов		
1. Основы программирования на Java	30	19	11	8	5		3
2. Java Core	30	14	7	10	6	12	4
3. Работа с базами данных	47	15	11	13	9	15	4
4. Работа на сервере	70	14	4	25	15	25	6
5. Работа с фреймворками (Spring)	67	18	8	25	10	20	4
6. Производственная практика ²	16					16	
7. Итоговая аттестация	10			3			7
ИТОГО	270	80	41	84	45	78	28

Проведение практических занятий и практики предусмотрено на базе ИТ-компаний, в том числе разрабатывающих цифровые решения для применения в выбранной отрасли (ООО «Артвелл»).

² Производственная практика может проходить в формате стажировки

IV. Календарный учебный график

Календарный учебный график представляет собой график учебного процесса, устанавливающий последовательность и продолжительность обучения, включая практику /стажировку, и итоговой аттестации по 9 месяцев, а также этапы оценки цифровых компетенций. При этом время, выделяемое на прохождение оценки сформированности цифровых компетенций, в общей трудоемкости Программы, отраженной в Учебном плане, не учитывается.

Структурные элементы (разделы Программы) и этапы оценки компетенций	Недели																			
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1. Основы программирования на Java	+	+	+	+	+															
2. Java Core						+	+	+	+	+										
3. Работа с базами данных											+	+	+	+	+					
4. Работа на сервере																+	+	+	+	+

Структурные элементы (разделы Программы) и этапы оценки компетенций	Недели																			
	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40
5. Работа с фреймворками (Spring)	+	+	+	+	+	+	+	+												
6. Производственная практика ³									+	+	+	+	+	+	+	+	+	+		
7. Итоговая аттестация																			+	+

У слушателей предусмотрено прохождение практики на базе ИТ-компаний, в том числе разрабатывающих цифровые решения для применения в выбранной отрасли (ООО «Артвелл»).

³ Производственная практика может проходить в формате стажировки

V. Рабочие программы модулей (курсов, дисциплин)

Рабочие программы разрабатываются для структурных элементов (разделов) Программы, указанных в Структуре Программы и Учебном плане, и содержат: перечень тем, включающих лекции, семинары, мастер-классы, практические занятия, самостоятельную работу, консультации и иные виды учебной работы с указанием краткого содержания и трудоемкости, образцы оценочных средств, методические материалы для преподавателей и обучающихся, сведения о кадровом обеспечении образовательного процесса. Рабочая программа практики / стажировки предусматривает определение цели и задач практической деятельности обучающихся, площадку (площадки) прохождения практики, задания (индивидуальные или групповые), критерии оценки результатов практической деятельности обучающихся.

Основы программирования на Java

1. Область применения рабочей программы

Рабочая программа «Основы программирования на Java» является частью дополнительной профессиональной программы профессиональной переподготовки ИТ-профиля «Программирование на Java: создание современных веб-приложений» и направлена на формирование базового понимания принципов работы языка, изучение его синтаксических конструкций, управляющих операторов, методов и массивов. Слушатели освоят фундаментальные подходы к разработке программ, разберут принципы передачи параметров и научатся использовать базовые инструменты для создания приложений.

Изучение данного модуля создаст прочную основу для дальнейшего освоения сложных концепций программирования, работы с фреймворками и разработки полноценных веб-приложений на Java.

Освоение рабочей программы является обязательным для всех обучающихся по Программе.

2. Структура и краткое содержание рабочей программы

№ п/п	Наименование модуля, разделов и тем	Содержание обучения, наименование и тематика практических занятий (семинаров), самостоятельной работы	Объем, ак.час.
1.	Модуль 1. Основы программирования на Java		30
1.11	Введение в Java и базовый синтаксис	Лекции Основы Java: Особенности Java, установка JDK, настройка IDE, создание и запуск первой программы. Переменные, типы данных, литералы, приведение типов. Основные операторы и выражения. Инкремент/декремент, сокращение операций присваивания. Логические операторы. Практика. Методы и переменные. Практика. Исключения. Практическая работа Выполнение заданий на программирование и итогового теста по теме лекций Самостоятельная работа Подготовка к семинарам и текущему контролю успеваемости. Самостоятельное выполнение заданий по теме лекции. Изучение учебной и научной литературы по теме лекции.	3
1.2	Управляющие конструкции (условные операторы, switch, циклы)	Лекции Условные конструкции. Конструкция if-else. Вложенные операторы if. Тернарный оператор.	3

		Традиционный оператор switch. Циклы. Использование цикла for. Циклы while, do-while, бесконечный цикл. Использование в циклах операторов break и continue. Практика Практическая работа Выполнение заданий на программирование и итогового теста по теме лекций Самостоятельная работа Подготовка к семинарам и текущему контролю успеваемости. Самостоятельное выполнение заданий по теме лекции. Изучение учебной и научной литературы по теме лекции.	
1.3	Методы и передача параметров	Лекции Введение в методы Параметры и аргументы в методах. Область видимости переменных. Возврат значения из метода. Рекурсивные методы. Введение в обработку исключений Практика Практическая работа Выполнение заданий на программирование и итогового теста по теме лекций Самостоятельная работа Подготовка к семинарам и текущему контролю успеваемости. Самостоятельное выполнение заданий по теме лекции. Изучение учебной и научной литературы по теме лекции.	5
1.4	Введение в ООП	Лекции Основы ООП. Понятия класс и объект. Оператор new, создание экземпляров класса. Конструкторы и сборщик мусора. Передача по ссылке и передача по значению. Объекты как аргументы методов. Ключевое слово this. Практическая работа Выполнение заданий на программирование и итогового теста по теме лекций Самостоятельная работа Подготовка к семинарам и текущему контролю успеваемости. Самостоятельное выполнение заданий по теме лекции.	5

		Изучение учебной и научной литературы по теме лекции.	
1.5	Работа с массивами	Лекции Введение в массивы Инициализация и доступ к элементам массива Обход массива. Цикл for в стиле for-each. Многомерные массивы. Массивы строк. Базовые операции с массивами Передача массивов в методы Практика: показ кода Практическая работа Выполнение заданий на программирование и итогового теста по теме лекций Самостоятельная работа Подготовка к семинарам и текущему контролю успеваемости. Самостоятельное выполнение заданий по теме лекции. Изучение учебной и научной литературы по теме лекции.	5
1.6	Практика	Практика кодинга. Знакомство с инструментом Jenkins.	6
1.7	Промежуточная аттестация	Задания на программирование Итоговый тест по теме урока	3
	Итого:		30

3. Учебно-тематический план рабочей программы

№ п/п	Наименование и краткое содержание структурного элемента (раздела) Программы	Количество часов			
		аудиторных		самостоятельной работы	
		всего, часов	практические занятия	всего, часов	практические занятия
1.1	Введение в Java и базовый синтаксис	2	1	1	
1.2	Управляющие конструкции (условные операторы, switch, циклы)	2	1	1	
1.3	Методы и передача параметров	4	2	1	
1.4	Введение в ООП	4	2	1	
1.5	Работа с массивами	4	2	1	
1.6	Практика: примеры и показ кода	3	3	3	3
1.7	Промежуточная аттестация			3	3

4. Контроль и оценка результатов освоения рабочей программы

Итоговая аттестация по программе является обязательной.

Форма итоговой аттестации предусматривает зачет в форме тестирования.

Оценка «зачтено» выставляется обучающемуся, если он показал всесторонние, систематизированные, глубокие знания учебной программы дисциплины и умение уверенно применять их на практике при решении конкретных задач, свободное и правильное обоснование принятых решений.

Оценка «не зачтено» выставляется обучающемуся, который не знает большей части основного содержания учебной программы дисциплины, допускает грубые ошибки в формулировках основных понятий дисциплины и не умеет использовать полученные знания при решении типовых практических задач.

Максимальная сумма, которую можно набрать, успешно выполнив все контрольные мероприятия, составляет 100 баллов. Для получения положительной оценки «зачтено» необходимо набрать не менее 30 баллов.

В зачетно-экзаменационную ведомость оценка выставляется в соответствии с нижеприведенной таблицей 14.

Таблица 14

Сумма баллов	Оценка
30-100	Зачтено
менее 30	Не зачтено

Составляющие процесса обучения, которые оцениваются в ходе обучения, и их вклад в итоговую оценку представлены в таблице 15.

Таблица 15

№ п/п	Основные показатели оценки	Вклад в итоговую оценку
1	Практические занятия	30%
2	Выполнение домашних заданий	40%
3	Промежуточная (итоговая) аттестация	30%

В процессе обучения студенты получают регулярную обратную связь, включающую следующие формы взаимодействия:

- Практические синхронные занятия: преподаватель проводит детальный разбор допущенных ошибок в заданиях.
- Проверка домашних заданий с подробными комментариями в системе дистанционного обучения (LMS).
- Асинхронная коммуникация: преподаватели оперативно отвечают на возникающие вопросы учащихся посредством кураторов курса.
- Промежуточные тестирования: регулярно проводятся контрольные испытания, позволяющие оценить уровень усвоенного материала и прогресс каждого студента.

4.1. Примеры оценочных средств

Оценочные материалы

Наименование модуля, разделов и тем	Основные показатели оценки	Формы и методы контроля и оценки	Вес задания, %
1. Основы программирования на Java	Компетенции 28, 333	Итоговый тест по теме "Основы программирования на Java"; выполнение	3,2

Наименование модуля, разделов и тем	Основные показатели оценки	Формы и методы контроля и оценки	Вес задания, %
		заданий на программирование	
2. Управляющие конструкции (условные операторы, switch, циклы)	Компетенции 28, 333	Итоговый тест по теме "Управляющие конструкции"; выполнение заданий на программирование	3,2
3. Методы и передача параметров	Компетенции 28, 333	Итоговый тест по теме "Методы и передача параметров"; выполнение заданий на программирование	3,2
4. Введение в ООП	Компетенции 28, 333	Итоговый тест по теме "Введение в ООП"; выполнение заданий на программирование	3,2
5. Работа с массивами	Компетенции 28, 333	Итоговый тест по теме "Работа с массивами"; выполнение заданий на программирование	3,2
Промежуточная аттестация	Компетенции 28, 333	Тест. Задание на программирование	16

Примеры заданий

Типовые вопросы из тестов с вариантами ответов для промежуточной аттестации

Итоговый тест по теме: Введение в Java и базовый синтаксис. Задание с автопроверкой

- Какую роль выполняет JDK (Java Development Kit) в процессе разработки на Java?**
 - а) Он выполняет программу
 - б) Он компилирует и запускает Java-программы
 - в) Он используется только для написания кода
 - г) Он предназначен для работы с базами данных
- Какие основные особенности выделяют язык Java? (выберите несколько вариантов)**
 - а) Платформонезависимость
 - б) Низкий уровень безопасности
 - в) Автоматическое управление памятью
 - г) Сложность освоения для начинающих
- Для чего используется IDE (среда разработки)?**
 - а) Для выполнения Java-программ
 - б) Для упрощения написания, отладки и запуска программ

- c) Только для компиляции кода
- d) Для создания и запуска программ только на C++

4. Какой результат выполнения следующей программы?

```
public class HelloWorld {
    public static void main(String[] args) {
        System.out.println("Hello, Java!");
    }
}
```

- a) Ошибка компиляции
- b) Ошибка времени выполнения
- c) "Hello, Java!"
- d) "Hello World!"

5. Какой из этих типов данных в Java используется для хранения десятичных чисел?

- a) int
- b) double
- c) char
- d) boolean

5. Образцы учебно-методических материалов для обучающихся и преподавателей

Задания на программирование по теме: Введение в Java и базовый синтаксис.

1. Приветствие

Напишите программу, которая выводит в консоль строку «Привет, мир!»

Ввод	Результат
-	Привет, мир!

Ответ:

```
public class Main {
    public static void main(String[] args) {

        System.out.println("Привет, мир!");

    }
}
```

2. Приветствие пользователя.

Напишите программу, которая будет приветствовать пользователя в формате «Привет, пользователь!». Имя должно передаваться при обращении к классу через командную строку, например: *java Main.java Сергей*

Ввод	Результат
Иван	Привет, Иван!
Анна	Привет, Анна!
1234	Привет, 1234!

Ответ:

```
public class Main {  
    public static void main(String[] args) {  
  
        System.out.println("Привет, " + args[0] + "!");  
  
    }  
}
```

3. Программа о себе.

Напишите программу, которая выводит в консоль ваше имя, возраст, страну и любимое занятие. Данные должны передаваться при обращении к классу через командную строку, например: *java Main.java Сергей 22 Россия Плавать*

Ввод	Результат
<i>Сергей 22 Россия Плавать</i>	Имя: Сергей Возраст: 22 Страна: Россия Хобби: Плавать
Анна 12 Италия Гулять	Имя: Анна Возраст: 12 Страна: Италия Хобби: Гулять
Павел 44 Япония Спать	Имя: Павел Возраст: 44 Страна: Япония Хобби: Спать

Ответ:

```
public class Main {  
    public static void main(String[] args) {  
  
        System.out.println("Имя: " + args[0]);  
  
    }  
}
```

```

System.out.println("Возраст: " + args[1]);
System.out.println("Страна: " + args[2]);
System.out.println("Хобби: " + args[3]);

```

Задания на программирование по теме: Работа с массивами.

1. Создание и вывод массива чисел.

Создайте массив из пяти целых чисел от 1 до 5 и выведите его на экран.

Ввод	Результат
-	1 2 3 4 5

Ответ:

```

public class Main {
    public static void main(String[] args) {
        int[] numbers = {1, 2, 3, 4, 5};
        for (int i = 0; i < numbers.length; i++) {
            System.out.print(numbers[i] + " ");
        }
    }
}

```

2. Инициализация массива и доступ к элементам.

Создайте массив из трех строк, заполните его значениями "Яблоко", "Банан", "Апельсин". Выведите в консоль «Яблоко Апельсин»

Ввод	Результат
-	Яблоко Апельсин

Ответ:

```

public class Main {
    public static void main(String[] args) {
        String[] fruits = new String[3];
        fruits[0] = "Яблоко";
        fruits[1] = "Банан";
        fruits[2] = "Апельсин";

        System.out.println(fruits[0] + " " + fruits[2]);
    }
}

```

3. Вывод элементов массива с помощью for-each.

Создайте массив чисел 5, 10, 15, 20. С помощью for-each выведите значения массива в консоль.

Ввод	Результат

-	10 20 30 40 50
---	----------------

Ответ:

```
public class Main {
    public static void main(String[] args) {
        int[] numbers = {10, 20, 30, 40, 50};

        for (int number : numbers) {
            System.out.print(number + " ");
        }
    }
}
```

6. Кадровое обеспечение образовательного процесса

ФИО	Место работы, должность	Учебная степень/звание	Высшее и (или) дополнительное образование в ИТ сфере	Модули (темы, лекции)	Стаж педагогической работы в образовательной организации высшего образования/работы в компании ИТ сферы	Дополнительная информация (реализованные проекты и (или) исследования, и (или) патенты, связанные с разработкой, внедрением цифровых технологий
Куртасов Андрей Михайлович	системный аналитик, ГК «Цифровые привычки»		2011 — Вологодский государственный технический университет, инженер по специальности «Информационн	Основы программирования на Java (30 часов: лекции и практические занятия)	10 лет / 6 лет	

			ые системы и технологии», специалитет			
--	--	--	---------------------------------------	--	--	--

7. Материально-техническое обеспечение реализации рабочей программы

Наименование специализированных аудиторий, кабинетов, лабораторий	Вид занятий	Наименование оборудования, программного обеспечения
Система дистанционного обучения	Лекция	Слушателю необходимо наличие доступа в сеть интернет, компьютер. Преподавателю курса необходимо наличие доступа администратора курса и оборудование для проведения дистанционных семинаров (вебинаров), качественный отказоустойчивый доступ в сеть интернет.
Система дистанционного обучения	Практическая работа	Слушателю необходимо наличие доступа в сеть интернет, компьютер. Преподавателю необходимо оборудование для проведения дистанционных семинаров (вебинаров), качественный отказоустойчивый доступ в сеть интернет.
Система дистанционного обучения	Самостоятельная работа	Слушателю необходимо наличие доступа в сеть интернет, компьютер.
Система дистанционного обучения	Промежуточная (итоговая) аттестация	Слушателю необходимо наличие доступа в сеть интернет, компьютер. Преподавателю необходимо оборудование для проведения дистанционных семинаров (вебинаров), качественный отказоустойчивый доступ в сеть интернет.

8. Информационное обеспечение реализации рабочей программы

Основная литература

1. Бек К. Экстремальное программирование: разработка через тестирование. Издательство “Питер”, 2020. - 224 с.
2. Блох, Д. Java Эффективное программирование / Д. Блох. - М.: Лори, 2016. - 440 с.
3. Java: руководство для начинающих, 9-е изд./Герберт Шилдт; пер. с англ. Ю.Н. Артеменко. — Киев. : “Науковий Свгг”, 2023. — 752 с.

4. Хорстманн Кей С. Java. Библиотека профессионала. Том 1. Основы / Хорстманн Кей С. - М.: Диалектика / Вильямс, 2020 г. - 864 с.
5. Хорстманн Кей С. Java. Библиотека профессионала. Том 2. Расширенные средства программирования, 11-е изд.: Пер с англ. - СПб.: / Диалектика, 2020 г. - 1008 с.
6. Эккель К. Философия Java. 4-е полное изд. Издательство "Питер", 2024. - 1168 с.

Дополнительная литература

1. Доуни А. Алгоритмы и структуры данных. Извлечение информации на языке Java. Издательство "Питер", 2018. - 240 с.
2. Дубаков А.А. Введение в объектно-ориентированное программирование на Java: учебное пособие [Электронный ресурс]. – СПб, Университет ИТМО, 2016. – 250 с. – Режим доступа: <http://books.ifmo.ru/file/pdf/2139.pdf>
3. Дэвид, Флэнаган JavaScript. Подробное руководство / Флэнаган Дэвид. - М.: Символ-плюс, 2017. - 555 с.
4. Перри, Брюс У. Java сервлеты и JSP: сб. рецептов, 2009 г. - 768 с.
5. Кожомбердиева, Г. И. Программирование на языке Java: многопоточные приложения: учебное пособие / Г. И. Кожомбердиева. — Санкт-Петербург: ПГУПС, 2012. — 44 с.
6. Хорстманн, Кей С. Java SE 8. Вводный курс / Хорстманн Кей С. - М.: Диалектика / Вильямс, 2014. - 898 с.

Ресурсы информационно-телекоммуникационной сети "Интернет"

1. Руководство по языку программирования Java <https://metanit.com/java/tutorial/>
2. Электронный ресурс <https://www.codewars.com/>
3. Java и Вы [Электронный ресурс]. – Режим доступа: <https://www.java.com/ru/>

Java Core

1. Область применения рабочей программы

Рабочая программа «Java Core» является частью дополнительной профессиональной программы профессиональной переподготовки ИТ-профиля «Программирование на Java: создание современных веб-приложений» и направлена на углубленное изучение ключевых возможностей языка Java, необходимых для разработки современных программных решений. В рамках модуля слушатели освоят работу с коллекциями, принципы объектно-ориентированного программирования (ООП), научатся работать с файлами и потоками ввода-вывода, познакомятся с основами многопоточности и сетевого взаимодействия. Также будет изучено использование лямбда-выражений и Stream API для обработки данных, что позволит разрабатывать более эффективные и производительные приложения. Освоение рабочей программы является обязательным для всех обучающихся по Программе.

2. Структура и краткое содержание рабочей программы

№ п/п	Наименование модуля, разделов и тем	Содержание обучения, наименование и тематика практических занятий (семинаров), самостоятельной работы	Объем, ак.час.
2	Модуль 2. Java Core		30
2.1	Введение в коллекции	Лекции Введение в коллекции List и его реализация Set и его реализация Map и его реализация Производительность коллекций Многопоточность и коллекции Практика Практическая работа Выполнение заданий на программирование и итогового теста по теме лекций Самостоятельная работа Подготовка к семинарам и текущему контролю успеваемости. Самостоятельное выполнение заданий по теме лекции. Изучение учебной и научной литературы по теме лекции.	3
2.2	Продвинутое ООП	Лекции Наследование в Java Инкапсуляция в Java Полиморфизм в Java Модификаторы доступа в Java Примеры и демонстрация Практическая работа Выполнение заданий на программирование и итогового теста по теме лекций Самостоятельная работа Подготовка к семинарам и текущему контролю успеваемости.	3

		Самостоятельное выполнение заданий по теме лекции. Изучение учебной и научной литературы по теме лекции.	
2.3	Работа с файлами и потоками	Лекции Введение в работу с файлами в Java Потоки ввода-вывода в Java (I/O Streams) Буферизированные потоки Форматированный ввод-вывод Работа с сериализацией объектов Работа с каналами и буферами в пакете java.nio Асинхронный ввод-вывод в Java Работа с потоками данных (Data Streams) Чтение и запись с использованием RandomAccessFile Практическая работа Выполнение заданий на программирование и итогового теста по теме лекций Самостоятельная работа Подготовка к семинарам и текущему контролю успеваемости. Самостоятельное выполнение заданий по теме лекции. Изучение учебной и научной литературы по теме лекции.	5
2.4	Введение в многопоточность	Лекции Основы многопоточности и создание потоков Управление потоками: приоритеты, ожидание и завершение Проблемы многопоточности Работа с высокоуровневыми механизмами синхронизации Несколько областей рисования с помощью Matplotlib Практическая работа Выполнение заданий на программирование и итогового теста по теме лекций Самостоятельная работа Подготовка к семинарам и текущему контролю успеваемости. Самостоятельное выполнение заданий по теме лекции. Изучение учебной и научной литературы по теме лекции.	4
2.5	Введение в работу с сетью	Лекции Основы сетевого программирования в Java Класс InetAddress и работа с IP-адресами Протокол TCP и работа с сокетами Протокол UDP и работа с датаграммами Протокол HTTP и класс HttpURLConnection	3

		Введение в NIO для сетевого программирования Асинхронное сетевое программирование с NIO2 Многопоточность и сеть Сетевые протоколы высокого уровня и работа с API Безопасность в сетевых приложениях Практическая работа Выполнение заданий на программирование и итогового теста по теме лекций Самостоятельная работа Подготовка к семинарам и текущему контролю успеваемости. Самостоятельное выполнение заданий по теме лекции. Изучение учебной и научной литературы по теме лекции.	
2.6	Лямбда-выражения и Stream API	Лекции Введение в функциональное программирование и лямбда-выражения Функциональные интерфейсы в Java Ссылки на методы и конструкторы Введение в Stream API Промежуточные операции Stream Терминальные операции Stream Коллекция результатов потока с Collectors Параллельные потоки Обработка ошибок и отладка при работе с лямбдами и Stream API Практическое применение лямбда-выражений и Stream API Практическая работа Выполнение заданий на программирование и итогового теста по теме лекций Самостоятельная работа Подготовка к семинарам и текущему контролю успеваемости. Самостоятельное выполнение заданий по теме лекции. Изучение учебной и научной литературы по теме лекции.	4
2.7	Практика: примеры и показ кода	Практика кодинга	4
2.8	Промежуточная аттестация	Задания на программирование Итоговый тест по теме урока	4

3. Учебно-тематический план рабочей программы

№ п/п	Наименование и краткое содержание структурного элемента (раздела) Программы	Количество часов			
		аудиторных		самостоятельной работы	
		всего, часов	практические занятия	всего, часов	практические занятия
2.1	Введение в коллекции	2	1	1	
2.2	Продвинутое ООП	2	1	1	
2.3	Работа с файлами и потоками	3	1	2	
2.4	Введение в многопоточность	3	1	1	
2.5	Введение в работу с сетью	1		2	
2.6	Лямбда-выражения и Stream API	2	1	1	
2.7	Практика: примеры и показ кода	2	2	2	2
2.8	Промежуточная аттестация			4	4

4. Контроль и оценка результатов освоения рабочей программы

Итоговая аттестация по программе является обязательной.

Форма итоговой аттестации предусматривает зачет в форме тестирования.

Оценка «зачтено» выставляется обучающемуся, если он показал всесторонние, систематизированные, глубокие знания учебной программы дисциплины и умение уверенно применять их на практике при решении конкретных задач, свободное и правильное обоснование принятых решений.

Оценка «не зачтено» выставляется обучающемуся, который не знает большей части основного содержания учебной программы дисциплины, допускает грубые ошибки в формулировках основных понятий дисциплины и не умеет использовать полученные знания при решении типовых практических задач.

Максимальная сумма, которую можно набрать, успешно выполнив все контрольные мероприятия, составляет 100 баллов. Для получения положительной оценки «зачтено» необходимо набрать не менее 30 баллов.

В зачетно-экзаменационную ведомость оценка выставляется в соответствии с нижеприведенной таблицей 22.

Таблица 22.

Сумма баллов	Оценка
30-100	Зачтено
менее 30	Не зачтено

Составляющие процесса обучения, которые оцениваются в ходе обучения, и их вклад в итоговую оценку представлены в таблице 23.

Таблица 23.

№ п/п	Основные показатели оценки	Вклад в итоговую оценку
1	Практические занятия	30%
2	Выполнение домашних заданий	40%
3	Промежуточная (итоговая) аттестация	30%

В процессе обучения студенты получают регулярную обратную связь, включающую следующие формы взаимодействия:

- Практические синхронные занятия: преподаватель проводит детальный разбор допущенных ошибок в заданиях.
- Проверка домашних заданий с подробными комментариями в системе дистанционного обучения (LMS).
- Асинхронная коммуникация: преподаватели оперативно отвечают на возникающие вопросы учащихся посредством кураторов курса.
- Промежуточные тестирования: регулярно проводятся контрольные испытания, позволяющие оценить уровень усвоенного материала и прогресс каждого студента.

4.1. Примеры оценочных средств

Оценочные материалы

Наименование модуля, разделов и тем	Основные показатели оценки	Формы и методы контроля и оценки	Вес задания, %
1. Введение в коллекции	Компетенции 28, 333	Итоговый тест по теме "Введение в коллекции"; выполнение заданий на программирование	2,6
2. Продвинутое ООП	Компетенции 28, 333	Итоговый тест по теме "Продвинутое ООП"; выполнение заданий на программирование	2,6
3. Работа с файлами и потоками	Компетенции 28, 333	Итоговый тест по теме "Работа с файлами и потоками"; выполнение заданий на программирование	2,7
4. Введение в многопоточность	Компетенции 28, 333	Итоговый тест по теме "Введение в многопоточность"; выполнение заданий на программирование	2,7
5. Введение в работу с сетью	Компетенции 28, 333	Итоговый тест по теме "Введение в работу с сетью"; выполнение заданий на программирование	2,7
6. Лямбда-выражения и Stream API	Компетенции 28, 333	Итоговый тест по теме "Лямбда-выражения и Stream API"; выполнение заданий на программирование	2,7
Промежуточная аттестация	Компетенции 28, 333	Тест. Задание на программирование	16

Примеры заданий

Тестовые задания

1. Какой из следующих интерфейсов является корневым в иерархии коллекций?

- a) Collection
- b) List
- c) Iterable
- d) Map

2. Заполните пропуск:

Все коллекции в Java находятся в пакете _____.

- b) java.nio
- c) java.util
- d) java.ArrayList

3. Вопрос с выбором нескольких ответов

Какие типы коллекций не допускают дублирующихся элементов?

- a) Set
- b) List
- c) Map
- d) HashSet

4. Вопрос на соответствие

Соотнесите термин и его определение:

- Итератор
- Генерик
- Пул объектов

- 1. Механизм для перебора элементов коллекции
- 2. Параметризованный тип для коллекций
- 3. Управляемый пул для повторного использования объектов

5. Вопрос с выбором одного ответа

Какой из методов интерфейса List используется для вставки элемента в определенную позицию?

- a) add(Object o)
- b) add(int index, Object o)
- c) insert(Object o)
- d) insert(int index, Object o)

5. Образцы учебно-методических материалов для обучающихся и преподавателей

Задание на программирование

Задание: Основы лямбда-выражений

- 1. **Задание:** Создайте функциональный интерфейс Operation, который будет представлять одну операцию, принимающую два целых числа и возвращающую результат в виде целого числа.

2. **Использование лямбда-выражений:** Реализуйте в методе `main` 3 операции с использованием лямбда-выражений:
 - o Сложение двух чисел
 - o Вычитание двух чисел
 - o Умножение двух чисел
3. Данные должны передаваться при обращении к классу через командную строку, например: `java Main.java 5 10`

Задание: Использование функциональных интерфейсов с коллекциями

1. **Задание:** Создайте список строк с именами пяти людей, данные для списка нужно передать через консоль при запуске кода на выполнение.
2. Используйте метод **`forEach`** и лямбда-выражение для вывода всех имен в верхнем регистре на экран.
3. Используйте метод **`removeIf`** и лямбда-выражение для удаления всех имен, длина которых меньше 4 символов.
4. Используйте метод **`map`** и лямбда-выражение для преобразования всех имен в их длину и выведите результат на экран.

6. Кадровое обеспечение образовательного процесса

ФИО	Место работы, должность	Ученая степень/ звание	Высшее и (или) дополнительное образование в ИТ сфере	Модули (темы, лекции)	Стаж педагогической работы в образовательной организации высшего образования/ работы в компании ИТ сферы	Дополнительная информация (реализованные проекты и (или) исследования, и (или) патенты, связанные с разработкой, внедрением, развитием цифровых технологий)
Апурин Николай Андреевич	генеральный директор		2009 - МИФИ факультет Б, комплексная	Java Core (10 часов: лекции и	-/ 17 лет	

	ООО «АРТВЕЛЛ»		информационная безопасность автоматизируе мых систем, специалитет	практические занятия)		
Дайняк Александр Борисович	Доцент кафедры дискретной математики МФТИ, Кандидат физико- математичес ких наук, доцент	Кан дида т физ ико- мате мати ческ их наук , доце нт	2009 - Московский государственны й университет им. М.В. Ломоносова, специальность «Прикладная математика и информатика», квалификация «математик, системный программист», специалитет 2018 – ФГАОУ ВО "Московский физико- технический институт (государственны й университет)", программа профессиональн ой переподготовки «Математическо е моделирование для анализа данных»	Java Core (20 часов: лекции и практические занятия)	16 лет/ -	+

7. Материально-техническое обеспечение реализации рабочей программы

Наименование специализированных аудиторий, кабинетов, лабораторий	Вид занятий	Наименование оборудования, программного обеспечения
Система дистанционного обучения	Лекция	Слушателю необходимо наличие доступа в сеть интернет, компьютер. Преподавателю курса необходимо наличие доступа администратора курса и оборудование для проведения дистанционных семинаров (вебинаров),

		качественный отказоустойчивый доступ в сеть интернет.
Система дистанционного обучения	Практическая работа	Слушателю необходимо наличие доступа в сеть интернет, компьютер. Преподавателю необходимо оборудование для проведения дистанционных семинаров (вебинаров), качественный отказоустойчивый доступ в сеть интернет.
Система дистанционного обучения	Самостоятельная работа	Слушателю необходимо наличие доступа в сеть интернет, компьютер.
Система дистанционного обучения	Промежуточная (итоговая) аттестация	Слушателю необходимо наличие доступа в сеть интернет, компьютер. Преподавателю необходимо оборудование для проведения дистанционных семинаров (вебинаров), качественный отказоустойчивый доступ в сеть интернет.

8. Информационное обеспечение реализации рабочей программы

Основная литература

1. Бек К. Экстремальное программирование: разработка через тестирование. Издательство “Питер”, 2020. - 224 с.
2. Блох, Д. Java Эффективное программирование / Д. Блох. - М.: Лори, 2016. - 440 с.
3. Java: руководство для начинающих, 9-е изд./Герберт Шилдт; пер. с англ. Ю.Н. Артеменко. —Киев. : “Науковий Свгг”, 2023. — 752 с.
4. Хорстманн Кей С. Java. Библиотека профессионала. Том 1. Основы / Хорстманн Кей С. - М.: Диалектика / Вильямс, 2020 г. - 864 с.
5. Хорстманн Кей С. Java. Библиотека профессионала. Том 2. Расширенные средства программирования, 11-е изд.: Пер с англ. - СПб.: / Диалектика, 2020 г. - 1008 с.
6. Эккель К. Философия Java. 4-е полное изд. Издательство “Питер”, 2024. - 1168 с.

Дополнительная литература

1. Доуни А. Алгоритмы и структуры данных. Извлечение информации на языке Java. Издательство “Питер”, 2018. - 240 с.
2. Дубаков А.А. Введение в объектно-ориентированное программирование на Java: учебное пособие [Электронный ресурс]. – СПб, Университет ИТМО, 2016. – 250 с. – Режим доступа: <http://books.ifmo.ru/file/pdf/2139.pdf>
3. Дэвид, Флэнаган JavaScript. Подробное руководство / Флэнаган Дэвид. - М.: Символ-плюс, 2017. - 555 с.
4. Перри, Брюс У. Java сервлеты и JSP: сб. рецептов, 2009 г. - 768 с.

5. Кожомбердиева, Г. И. Программирование на языке Java: многопоточные приложения: учебное пособие / Г. И. Кожомбердиева. — Санкт-Петербург: ПГУПС, 2012. — 44 с.
6. Хорстманн, Кей С. Java SE 8. Вводный курс / Хорстманн Кей С. - М.: Диалектика / Вильямс, 2014. - 898 с.

Ресурсы информационно-телекоммуникационной сети "Интернет"

1. Руководство по языку программирования Java <https://metanit.com/java/tutorial/>
2. Электронный ресурс <https://www.codewars.com/>
3. Java и Вы [Электронный ресурс]. – Режим доступа: <https://www.java.com/ru/>

Работа с базами данных

1. Область применения рабочей программы

Рабочая программа «Работа с базами данных» является частью дополнительной профессиональной программы профессиональной переподготовки ИТ-профиля «Программирование на Java: создание современных веб-приложений» и направлена на изучение основ реляционных баз данных и языка SQL, освоение работы с базами данных с использованием JDBC, выполнение CRUD-операций, управление транзакциями, а также изучение принципов ORM и работы с Hibernate. Дополнительно в рамках модуля рассматриваются вопросы проектирования баз данных и создания DAO (Data Access Object) для эффективного взаимодействия с хранилищем данных.

Освоение рабочей программы является обязательным для всех обучающихся по Программе.

2. Структура и краткое содержание рабочей программы

№ п/п	Наименование модуля, разделов и тем	Содержание обучения, наименование и тематика практических занятий (семинаров), самостоятельной работы	Объем, ак.час.
3.	Модуль 3. Работа с базами данных		47
3.1	Введение в реляционные базы данных и SQL	Лекции Основы реляционных баз данных (РБД) Основы SQL (Structured Query Language) Практика. Создание таблиц, добавление данных. Связи между таблицами и JOIN-операции Условные операторы и функции Практика. Выборка из одной таблицы Практические аспекты Базовые представления и хранимые процедуры Практика. Многотабличные запросы. Практическая работа Выполнение комплексного практического задания и итогового теста по теме урока.	3
3.2	Подключение к базе данных с использованием JDBC	Лекции Введение в JDBC Основные классы и интерфейсы JDBC Работа с SQL-запросами через JDBC Обработка исключений в JDBC Типы запросов в JDBC Практика. Подключение к БД через Java Практическая работа Выполнение комплексного практического задания и итогового теста по теме урока.	3
3.3	Выполнение CRUD операций через JDBC	Лекции Введение в CRUD-операции Операция Create (вставка данных) Операция Read (чтение данных) Операция Update (обновление данных)	4

		Операция Delete (удаление данных) Практика. Выполнение SQL запросов через Java Работа с параметризованными запросами Практическая работа Выполнение комплексного практического задания и итогового теста по теме урока.	
3.4	Транзакции и управление транзакциями	Лекции Основные концепции транзакций Типы транзакций Механизмы управления транзакциями Изолированность транзакций Распределённые транзакции Практика. Транзакции и подготовленные запросы. Технологические аспекты Практическая работа Выполнение комплексного практического задания и итогового теста по теме урока.	4
3.5	ORM и работа с Hibernate	Лекции Основы ORM Введение в Hibernate Работа с основными сущностями Hibernate Конфигурация Hibernate Работа с БД через Hibernate Запросы в Hibernate Практическая работа Выполнение комплексного практического задания и итогового теста по теме урока.	4
3.6	Проектирование базы данных и создание DAO	Лекции Основы проектирования базы данных Паттерн DAO (Data Access Object) Проектирование слоя DAO Моделирование сущностей Обработка данных в DAO Практическая работа Выполнение задания по проекту и итогового теста по теме урока. Самостоятельная работа Подготовка к семинарам и текущему контролю успеваемости. Самостоятельное выполнение заданий по теме лекции. Изучение учебной и научной литературы по теме лекции.	5
3.7	Практика	Примеры кода	9
3.8	Основы информационной безопасности	Лекция Введение в информационную безопасность Каналы утечки информации и риски безопасности Методы и подходы технической защиты информации Средства защиты данных в цифровых	11

		продуктах и сервисах Самостоятельная работа Тест на самопроверку Дополнительные материалы	
3.9	Промежуточная аттестация	Задания на программирование Итоговый тест по теме урока	4

3. Учебно-тематический план рабочей программы

№ п/п	Наименование и краткое содержание структурного элемента (раздела) Программы	Количество часов			
		аудиторных		самостоятельной работы	
		всего, часов	практические занятия	всего, часов	практические занятия
3.1	Введение в реляционные базы данных и SQL	2	1		
3.2	Подключение к базе данных с использованием JDBC	2	1		
3.3	Выполнение CRUD операций через JDBC	2	1		
3.4	Транзакции и управление транзакциями	2	1		
3.5	ORM и работа с Hibernate	2	1		
3.6	Проектирование базы данных и создание DAO	3	2		
3.7	Практика: примеры и показ кода	4	4	5	5
3.8	Основы информационной безопасности	3		8	4
3.9	Промежуточная аттестация			4	4

4. Контроль и оценка результатов освоения рабочей программы

Итоговая аттестация по программе является обязательной.

Форма итоговой аттестации предусматривает зачет в форме тестирования.

Оценка «зачтено» выставляется обучающемуся, если он показал всесторонние, систематизированные, глубокие знания учебной программы дисциплины и умение уверенно применять их на практике при решении конкретных задач, свободное и правильное обоснование принятых решений.

Оценка «не зачтено» выставляется обучающемуся, который не знает большей части основного содержания учебной программы дисциплины, допускает грубые ошибки в формулировках основных понятий дисциплины и не умеет использовать полученные знания при решении типовых практических задач.

Максимальная сумма, которую можно набрать, успешно выполнив все контрольные мероприятия, составляет 100 баллов. Для получения положительной оценки «зачтено» необходимо набрать не менее 30 баллов.

В зачетно-экзаменационную ведомость оценка выставляется в соответствии с нижеприведенной таблицей 31.

Таблица 31

Сумма баллов	Оценка
30-100	Зачтено
менее 30	Не зачтено

Составляющие процесса обучения, которые оцениваются в ходе обучения, и их вклад в итоговую оценку представлены в таблице 32.

Таблица 32

№ п/п	Основные показатели оценки	Вклад в итоговую оценку
1	Практические занятия	30%
2	Выполнение домашних заданий	40%
3	Промежуточная (итоговая) аттестация	30%

В процессе обучения студенты получают регулярную обратную связь, включающую следующие формы взаимодействия:

- Практические синхронные занятия: преподаватель проводит детальный разбор допущенных ошибок в заданиях.
- Проверка домашних заданий с подробными комментариями в системе дистанционного обучения (LMS).
- Асинхронная коммуникация: преподаватели оперативно отвечают на возникающие вопросы учащихся посредством кураторов курса.
- Промежуточные тестирования: регулярно проводятся контрольные испытания, позволяющие оценить уровень усвоенного материала и прогресс каждого студента.

4.1. Примеры оценочных средств

Оценочные материалы

Наименование модуля, разделов и тем	Основные показатели оценки	Формы и методы контроля и оценки	Вес задания, %
1. Введение в реляционные базы данных и SQL	Компетенция 32	Итоговое тестирование по теме уроков; выполнение заданий на программирование	2,6
2. Подключение к базе данных с использованием JDBC	Компетенция 32	Итоговое тестирование по теме уроков; выполнение заданий на программирование	2,6
3. Выполнение CRUD операций через JDBC	Компетенция 32	Итоговое тестирование по теме уроков; выполнение	2,7

Наименование модуля, разделов и тем	Основные показатели оценки	Формы и методы контроля и оценки	Вес задания, %
		заданий на программирование	
4. Транзакции и управление транзакциями	Компетенция 32	Итоговое тестирование по теме уроков; выполнение заданий на программирование	2,7
5. ORM и работа с Hibernate	Компетенция 32	Итоговое тестирование по теме уроков; выполнение заданий на программирование	2,7
6. Проектирование базы данных и создание DAO	Компетенция 32	Итоговое тестирование по теме уроков; выполнение заданий на программирование	2,7
Промежуточная аттестация	Компетенция 32	Тест. Задание на программирование	16

Примеры заданий

Тестовые задания

1. Вопрос с одним вариантом ответа

Что такое ORM?

- A. Технология для хранения бинарных файлов в базе данных.
- B. Подход для сопоставления объектов приложения с данными в базе данных.
- C. Метод ускорения работы SQL-запросов.
- D. Средство для визуализации данных.

2. Вопрос с множественным выбором

Какие преимущества дает ORM?

- A. Упрощение работы с базами данных.
- B. Сокращение количества SQL-запросов.
- C. Автоматическая генерация кода на языке SQL.
- D. Обеспечение независимости от типа СУБД.
- E. Снижение производительности системы.

3. Задание на сопоставление

Сопоставьте элементы ORM с их описанием:

- 1. Entity → B. Класс, соответствующий таблице в базе данных.
- 2. Session → A. Объект для работы с транзакциями и запросами.
- 3. Mapping → C. Сопоставление классов и полей с таблицами и колонками.
- 4. Persistence Context → D. Контекст для управления состоянием объектов в рамках транзакции.

4. Вопрос с одним вариантом ответа

Что такое Hibernate?

- A. ORM-фреймворк для Java-приложений.
- B. Библиотека для работы с REST API.
- C. SQL-редактор с графическим интерфейсом.
- D. Средство для создания пользовательского интерфейса.

5. Вопрос с множественным выбором

Какие функции выполняет Hibernate?

- A. Генерация SQL-запросов.
- B. Кэширование данных.
- C. Управление транзакциями.
- D. Создание индексов в базе данных.
- E. Сопоставление Java-классов с таблицами.

5. Образцы учебно-методических материалов для обучающихся и преподавателей

Комплексное задание на программирование

В рамках данного задания вы разработаете мини-приложение для управления банковскими счетами с использованием базы данных. Приложение должно позволять:

1. Создавать новые счета.
2. Просматривать информацию о балансе конкретного счета.
3. Переводить средства с одного счета на другой.

Основное внимание нужно уделить правильной реализации транзакций, чтобы избежать неконсистентного состояния данных в случае сбоев.

Требования к реализации

1. Используемые технологии:

- Java (JDK 11+)
- JDBC или ORM (например, Hibernate)
- База данных (например, PostgreSQL, MySQL или H2 для тестов)

2. Основные функциональности:

- Создание банковского счета: метод для создания нового счета с указанием начального баланса.
- Просмотр баланса счета: метод, возвращающий текущий баланс по номеру счета.
- Перевод средств между счетами: метод, осуществляющий перевод денег с одного счета на другой. Этот процесс должен быть обернут в транзакцию.

3. Обработка ошибок:

- Если в процессе перевода средств возникает ошибка (например, недостаточно средств на счете отправителя), транзакция должна быть откатана.
- Не допускайте неконсистентного состояния (например, списание средств произошло, но они не были зачислены на другой счет).

4. Таблица в базе данных: Создайте таблицу `accounts` с колонками:

- `id` (PRIMARY KEY, AUTO_INCREMENT)
- `account_number` (VARCHAR, UNIQUE)
- `balance` (DECIMAL)

5. Примерные методы:

- void createAccount(String accountNumber, BigDecimal initialBalance)
- BigDecimal getBalance(String accountNumber)
- void transfer(String fromAccount, String toAccount, BigDecimal amount)

6. Кадровое обеспечение образовательного процесса

ФИО	Место работы, должность	Ученая степень/ звание	Высшее и (или) дополнительное образование в ИТ сфере	Модули (темы, лекции)	Стаж педагогической работы в образовательной организации высшего образования/ работы в компании ИТ сферы	Дополнительная информация (реализованные проекты и (или) исследования, и (или) патенты, связанные с разработкой, внедрением, развитием цифровых технологий
Апурин Николай Андреевич	генеральный директор ООО «АРТВЕЛЛ»		2009 - МИФИ факультет Б, комплексная информационная безопасность автоматизируемых систем, специалитет	Работа с базами данных (20 часов: лекции и практические занятия)	-/ 17 лет	
Максимова Дарья Максимовна	инженер по тестированию, Toloka.ai		2019 – НИУ ВШЭ, Фундаментальная и прикладная лингвистика, бакалавриат	Работа с базами данных (35 часов: лекции и практические занятия)	6 лет/ 6 лет	

			2021 – НИУ ВШЭ, Цифровые методы в гуманитарных науках, магистратура			
--	--	--	--	--	--	--

7. Материально-техническое обеспечение реализации рабочей программы

Наименование специализированных аудиторий, кабинетов, лабораторий	Вид занятий	Наименование оборудования, программного обеспечения
Система дистанционного обучения	Лекция	Слушателю необходимо наличие доступа в сеть интернет, компьютер. Преподавателю курса необходимо наличие доступа администратора курса и оборудование для проведения дистанционных семинаров (вебинаров), качественный отказоустойчивый доступ в сеть интернет.
Система дистанционного обучения	Практическая работа	Слушателю необходимо наличие доступа в сеть интернет, компьютер. Преподавателю необходимо оборудование для проведения дистанционных семинаров (вебинаров), качественный отказоустойчивый доступ в сеть интернет.
Система дистанционного обучения	Самостоятельная работа	Слушателю необходимо наличие доступа в сеть интернет, компьютер.
Система дистанционного обучения	Промежуточная (итоговая) аттестация	Слушателю необходимо наличие доступа в сеть интернет, компьютер. Преподавателю необходимо оборудование для проведения дистанционных семинаров (вебинаров), качественный отказоустойчивый доступ в сеть интернет.

8. Информационное обеспечение реализации рабочей программы

Основная литература

1. Бек К. Экстремальное программирование: разработка через тестирование. Издательство “Питер”, 2020. - 224 с.
2. Блох, Д. Java Эффективное программирование / Д. Блох. - М.: Лори, 2016. - 440 с.
3. Java: руководство для начинающих, 9-е изд./Герберт Шилдт; пер. с англ. Ю.Н. Артеменко. — Киев. : “Науковий Світ”, 2023. — 752 с.

4. Хорстманн Кей С. Java. Библиотека профессионала. Том 1. Основы / Хорстманн Кей С. - М.: Диалектика / Вильямс, 2020 г. - 864 с.
5. Хорстманн Кей С. Java. Библиотека профессионала. Том 2. Расширенные средства программирования, 11-е изд.: Пер с англ. - СПб.: / Диалектика, 2020 г. - 1008 с.
6. Эккель К. Философия Java. 4-е полное изд. Издательство "Питер", 2024. - 1168 с.

Дополнительная литература

1. Доуни А. Алгоритмы и структуры данных. Извлечение информации на языке Java. Издательство "Питер", 2018. - 240 с.
2. Дубаков А.А. Введение в объектно-ориентированное программирование на Java: учебное пособие [Электронный ресурс]. – СПб, Университет ИТМО, 2016. – 250 с. – Режим доступа: <http://books.ifmo.ru/file/pdf/2139.pdf>
3. Дэвид, Флэнаган JavaScript. Подробное руководство / Флэнаган Дэвид. - М.: Символ-плюс, 2017. - 555 с.
4. Перри, Брюс У. Java сервлеты и JSP: сб. рецептов, 2009 г. - 768 с.
5. Кожомбердиева, Г. И. Программирование на языке Java: многопоточные приложения : учебное пособие / Г. И. Кожомбердиева. — Санкт-Петербург: ПГУПС, 2012. — 44 с.
6. Хорстманн, Кей С. Java SE 8. Вводный курс / Хорстманн Кей С. - М.: Диалектика / Вильямс, 2014. - 898 с.

Ресурсы информационно-телекоммуникационной сети "Интернет"

1. Руководство по языку программирования Java <https://metanit.com/java/tutorial/>
2. Электронный ресурс <https://www.codewars.com/>
3. Java и Вы [Электронный ресурс]. – Режим доступа: <https://www.java.com/ru/>

Работа на сервере

1. Область применения рабочей программы

Рабочая программа «Работа на сервере» является частью дополнительной профессиональной программы профессиональной переподготовки ИТ-профиля «Программирование на Java: создание современных веб-приложений» и направлена на изучение основ клиент-серверной архитектуры, принципов работы HTTP, а также основ HTML и CSS. В рамках модуля обучающиеся познакомятся с сервлетами, JSP и JavaBeans, освоят разработку простых веб-приложений, создание RESTful API, а также управление сессиями и механизмы аутентификации пользователей.

Освоение рабочей программы является обязательным для всех обучающихся по Программе.

2. Структура и краткое содержание рабочей программы

№ п/п	Наименование модуля, разделов и тем	Содержание обучения, наименование и тематика практических занятий (семинаров), самостоятельной работы	Объем, ак.час.
4	Модуль 4. Работа на сервере		70
4.1	Основы клиент-серверной архитектуры, HTTP	Лекции Основы клиент-серверной архитектуры Протокол HTTP Основы HTML Основы CSS Практика Практическая работа Выполнение заданий по проекту. Тесты и задания на самопроверку. Самостоятельная работа Подготовка к семинарам и текущему контролю успеваемости. Самостоятельное выполнение заданий по теме лекции. Изучение учебной и научной литературы по теме лекции.	7
4.2	Введение в сервлеты	Лекции Обзор Servlet API Установка и настройка окружения Основы работы с сервлетами Обработка HTTP-запросов и формирование ответов Деплой и конфигурация сервлетов Практическая работа Выполнение заданий по проекту. Тесты и задания на самопроверку. Самостоятельная работа Подготовка к семинарам и текущему контролю успеваемости. Самостоятельное выполнение заданий по теме лекции.	7

		Изучение учебной и научной литературы по теме лекции.	
4.3	Введение в JSP и JavaBeans	Лекции Основы JSP (JavaServer Pages) JavaBeans и их роль в разработке на JSP Связь JSP и сервлетов Работа с формами и валидация данных Работа с базами данных через JSP и JavaBeans Практическая работа Выполнение заданий по проекту. Тесты и задания на самопроверку. Самостоятельная работа Подготовка к семинарам и текущему контролю успеваемости. Самостоятельное выполнение заданий по теме лекции. Изучение учебной и научной литературы по теме лекции.	12
4.4	Разработка простого веб-приложения	Лекции Введение в веб-разработку на Java HTML, CSS и JavaScript Разработка Servlets Работа с JSP (JavaServer Pages) Работа с формами и обработка данных Использование JavaBeans Подключение к базе данных Практическая работа Выполнение заданий по проекту. Тесты и задания на самопроверку. Самостоятельная работа Подготовка к семинарам и текущему контролю успеваемости. Самостоятельное выполнение заданий по теме лекции. Изучение учебной и научной литературы по теме лекции.	9
4.5	Введение в разработку RESTful API	Лекции Введение в REST и API Подготовка окружения Создание базового HTTP-сервера на Java Реализация CRUD-операций Обработка параметров и тела запроса Практическая работа Выполнение заданий по проекту. Тесты и задания на самопроверку. Самостоятельная работа Подготовка к семинарам и текущему контролю успеваемости. Самостоятельное выполнение заданий по теме лекции.	12

		Изучение учебной и научной литературы по теме лекции.	
4.6	Управление сессиями и аутентификация	Лекции Введение в управление сессиями Работа с HTTP-сессиями в Java (Servlet API) Аутентификация пользователей Использование фильтров для защиты ресурсов Создание системы авторизации (Roles and Permissions) Управление сессиями и аутентификацией Обработка выхода пользователя (Logout). Практическая работа Выполнение заданий по проекту. Тесты и задания на самопроверку. Самостоятельная работа Подготовка к семинарам и текущему контролю успеваемости. Самостоятельное выполнение заданий по теме лекции. Изучение учебной и научной литературы по теме лекции.	8
4.7	Проектирование и реализация DAO в серверных приложениях	Лекции Основы DAO (Data Access Object) Интеграция DAO с сервлетами Создание сервисного слоя Обработка ошибок и логирование Тестирование и отладка серверных приложений Практическая работа Выполнение заданий по проекту. Тесты и задания на самопроверку. Самостоятельная работа Подготовка к семинарам и текущему контролю успеваемости. Самостоятельное выполнение заданий по теме лекции. Изучение учебной и научной литературы по теме лекции.	5
4.8	Практика	Примеры кода	4
4.9	Промежуточная аттестация	Проект	6

3. Учебно-тематический план рабочей программы

№ п/п	Наименование и краткое содержание структурного элемента (раздела) Программы	Количество часов			
		аудиторных		самостоятельной работы	
		всего, часов	практические занятия	всего, часов	практические занятия
4.1	Основы клиент-серверной архитектуры и HTTP	1		5	
4.2	Введение в сервлеты	1		5	
4.3	Введение в JSP и JavaBeans	5	1	7	
4.4	Разработка простого веб-приложения	1		5	
4.5	Введение в разработку RESTful API	2	1	7	
4.6	Управление сессиями и аутентификация	1		6	
4.7	Проектирование и реализация DAO в серверных приложениях	2	1	2	2
4.8	Практика: примеры и показ кода	1	1	3	3
4.9	Промежуточная аттестация			6	6

4. Контроль и оценка результатов освоения рабочей программы

Итоговая аттестация по программе является обязательной.

Форма итоговой аттестации предусматривает зачет в форме тестирования.

Оценка «зачтено» выставляется обучающемуся, если он показал всесторонние, систематизированные, глубокие знания учебной программы дисциплины и умение уверенно применять их на практике при решении конкретных задач, свободное и правильное обоснование принятых решений.

Оценка «не зачтено» выставляется обучающемуся, который не знает большей части основного содержания учебной программы дисциплины, допускает грубые ошибки в формулировках основных понятий дисциплины и не умеет использовать полученные знания при решении типовых практических задач.

Максимальная сумма, которую можно набрать, успешно выполнив все контрольные мероприятия, составляет 100 баллов. Для получения положительной оценки «зачтено» необходимо набрать не менее 30 баллов.

В зачетно-экзаменационную ведомость оценка выставляется в соответствии с нижеприведенной таблицей 41.

Таблица 41

Сумма баллов	Оценка
30-100	Зачтено
менее 30	Не зачтено

Составляющие процесса обучения, которые оцениваются в ходе обучения, и их вклад в итоговую оценку представлены в таблице 42.

Таблица 42

№ п/п	Основные показатели оценки	Вклад в итоговую оценку
1	Практические занятия	30%
2	Выполнение домашних заданий	40%
3	Промежуточная (итоговая) аттестация	30%

В процессе обучения студенты получают регулярную обратную связь, включающую следующие формы взаимодействия:

- Практические синхронные занятия: преподаватель проводит детальный разбор допущенных ошибок в заданиях.
- Проверка домашних заданий с подробными комментариями в системе дистанционного обучения (LMS).
- Асинхронная коммуникация: преподаватели оперативно отвечают на возникающие вопросы учащихся посредством кураторов курса.
- Промежуточные тестирования: регулярно проводятся контрольные испытания, позволяющие оценить уровень усвоенного материала и прогресс каждого студента.

4.1. Примеры оценочных средств

Оценочные материалы

Наименование модуля, разделов и тем	Основные показатели оценки	Формы и методы контроля и оценки	Вес задания, %
1. Основы клиент-серверной архитектуры, HTTP	Компетенции 333, 335	Выполнение заданий по проекту	1,5
2. Введение в сервлеты	Компетенции 333, 335	Выполнение заданий по проекту	1,5
3. Введение в JSP и JavaBeans	Компетенции 333, 335	Выполнение заданий по проекту	2,6
4. Разработка простого веб-приложения	Компетенции 333, 335	Выполнение заданий по проекту	2,6
5. Введение в разработку RESTful API	Компетенции 333, 335	Выполнение заданий по проекту	2,6
6. Управление сессиями и аутентификация	Компетенции 333, 335	Выполнение заданий по проекту	2,6
7. Проектирование и реализация DAO в серверных приложениях	Компетенции 333, 335	Выполнение заданий по проекту	2,6
Промежуточная аттестация	Компетенции 333, 335	Проект	16

Примеры заданий

Тестовые задания

1. **Что такое клиент в клиент-серверной архитектуре?**
 - a) Устройство или программа, которая запрашивает ресурсы.
 - b) Устройство или программа, которая предоставляет ресурсы.
 - c) Устройство, которое только передает данные.
 - d) Устройство, которое хранит данные.
2. **Какие из перечисленных компонентов относятся к клиент-серверной архитектуре? (Выберите несколько вариантов)**
 - a) Браузер
 - b) Сервер базы данных
 - c) Маршрутизатор
 - d) Веб-сервер
3. **Сопоставьте компоненты клиент-серверной архитектуры с их функциями:**
 1. Клиент
 2. Сервер
 3. Сеть
 - a) Обеспечивает передачу данных между клиентом и сервером.
 - b) Обрабатывает запросы и предоставляет ресурсы.
 - c) Отправляет запросы на получение ресурсов.
4. **Что такое тонкий клиент?**
 - a) Клиент, который выполняет минимальную обработку данных.
 - b) Клиент, который выполняет всю обработку данных.
 - c) Клиент, который не взаимодействует с сервером.
 - d) Клиент, который только хранит данные.
5. **Что означает аббревиатура HTTP?**
 - a) HyperText Transfer Protocol
 - b) High-Level Text Protocol
 - c) HyperText Transmission Process
 - d) Hyper Transfer Text Protocol
6. **Какие методы HTTP используются для запросов? (Сопоставьте методы с их описанием)**
 1. GET
 2. POST
 3. PUT
 4. DELETE
 - a) Запрос на получение данных.
 - b) Запрос на добавление или обновление данных.
 - c) Запрос на удаление данных.
 - d) Запрос на создание новых данных.
7. **Какие из перечисленных кодов состояния HTTP относятся к успешным ответам? (Выберите несколько вариантов)**
 - a) 200

- b) 404
- c) 500
- d) 201

8. **Что такое HTTPS?**

- a) HTTP с поддержкой шифрования данных.
- b) HTTP с поддержкой сжатия данных.
- c) HTTP с поддержкой кэширования.
- d) HTTP с поддержкой мультимедиа.

9. **Какой тег используется для создания заголовка первого уровня?**

- a) <h2>
- b) <h1>
- c) <header>
- d) <title>

10. **Какие из перечисленных тегов являются парными? (Выберите несколько вариантов)**

- a)
- b) <p>
- c)

- d) <div>

5. **Образцы учебно-методических материалов для обучающихся и преподавателей**

Комплексное практическое задание

Задание: Создание многостраничного сайта.

Техническое задание

1. **Название проекта:** "Личное портфолио".
2. **Структура сайта:** Сайт должен содержать три страницы:
 - o **Главная страница (index.html):**
 - Краткая информация о владельце портфолио.
 - Ссылки на другие страницы.
 - o **Обо мне (about.html):**
 - Развёрнутый текст о владельце (образование, хобби, интересы).
 - o **Проекты (projects.html):**
 - Список проектов в виде карточек с описанием и кнопками для перехода на отдельные страницы проектов (вымышленные ссылки).
3. **Общие элементы дизайна:**
 - o Шапка и подвал должны быть одинаковыми на всех страницах.
 - o Навигационное меню в шапке с ссылками на все страницы.
 - o Сетка контента должна быть построена с использованием CSS **Flexbox** или **Grid**.
 - o Каждый блок контента должен быть визуально разделён (например, использовать отступы, рамки или фоновые цвета).
4. **Структура навигации:**
 - o Меню шапки:
 - "Главная".
 - "Обо мне".

- "Проекты".
 - Ссылки в подвале: "Контакты" (ведёт на форму на главной странице), "Социальные сети".
5. **Технические требования:**
- Используйте только HTML и CSS (без адаптации и анимации).
 - Стили должны быть подключены через внешнюю таблицу CSS.
 - Строить макет страниц с помощью **Flexbox** или **Grid**.
 - Использовать семантические теги: `header`, `nav`, `main`, `section`, `footer`.

Проект: Разработка веб-приложения "Интернет-магазин"

Описание

проекта:

В рамках учебного проекта вам предстоит разработать веб-приложение "Интернет-магазин". Оно будет включать клиентскую и серверную часть, а также базу данных. Основной функционал приложения охватывает управление пользователями, товарами, категориями, корзиной и заказами.

Вы должны использовать технологии, изученные в предыдущем модуле, включая Java, JDBC, работу с базами данных (модуль 3), а также закрепить полученные знания в модуле 4: сервлеты, JSP, HTML/CSS и SQL.

По итогу проекта должно получиться работоспособное веб-приложение, способное обрабатывать HTTP-запросы и взаимодействовать с базой данных.

На выполнение проекта вам дается 4 недели. На 5 неделе 4 модуля вам предстоит сдать свой проект. Для сдачи проекта прикрепите ссылку на ноутбук с кодом своего интернет-магазина, а также ссылку на скринкаст с показом запуска интернет-магазина.

Функциональные требования

1. Категории товаров:

- Добавление, удаление, обновление категорий (администратор).
- Получение списка категорий (для отображения в магазине).

2. Товары:

- Добавление, удаление, обновление товаров (администратор).
- Получение списка товаров с фильтрацией по категориям.
- Просмотр информации о товаре (детальная страница).

3. Корзина:

- Добавление товаров в корзину.
- Просмотр товаров в корзине.
- Удаление товаров из корзины.
- Оформление заказа на основе товаров в корзине.

4. Заказы:

- Оформление заказа.
- Просмотр истории заказов пользователя.
- Просмотр всех заказов (администратор).

5. Пользователи:

- Регистрация новых пользователей.
- Авторизация (вход в систему).
- Выход из аккаунта.
- Разграничение прав (администратор/покупатель).

Структура проекта

1. База данных

Создание структуры базы данных **ecommerce_db**, включающей таблицы:

- Users (id, username, password, role);
- Category (id, name);
- Product (id, name, description, price, stock, category_id);
- Cart (id, user_id);
- Cart_Product (cart_id, product_id, quantity);
- Orders (id, user_id, total_price, order_date);
- Order_Product (order_id, product_id, quantity).

Форма сдачи: **.sql-файл со структурой базы данных.**

2. Реализация модели данных

Создание классов моделей (User, Category, Product, Cart, Order), соответствующих таблицам в БД. Форма сдачи: **архив с кодом моделей.**

3. Реализация слоя DAO

Разработка интерфейсов и их реализаций для работы с базой данных (UserDAO, CategoryDAO, ProductDAO, CartDAO, OrderDAO). Форма сдачи: **архив с кодом DAO-слоя и тестами CRUD-операций.**

4. Веб-интерфейс

Создание HTML-страниц:

- index.html – главная страница с каталогом товаров;
- product.html – страница товара;
- cart.html – корзина;
- register.html, login.html – регистрация/вход.

Разработка CSS-стилей и базового дизайна. Форма сдачи: **архив с кодом веб-интерфейса.**

5. Реализация бизнес-логики и сервлетов

- UserService, ProductService, CartService, OrderService – сервисы для обработки данных.
- LoginServlet, CartServlet, OrderServlet – обработка HTTP-запросов.

- Связь веб-интерфейса с серверной частью. Форма сдачи: **архив с кодом сервлетов и сервисов.**

6. Финальное тестирование и презентация

- Проверка всех функциональных требований.
- Демонстрация работы приложения.
- Запись видео (скринкаст) с обзором функционала. Форма сдачи: **финальный архив с кодом + видео-презентация.**

Ожидаемый результат

К концу модуля студенты должны представить полностью работоспособное веб-приложение "Интернет-магазин" с авторизацией, базой данных, бизнес-логикой и пользовательским интерфейсом.

Преподаватели оценивают проект по следующим критериям:

1. **Корректность работы функционала** – 40%
2. **Качество кода и архитектуры проекта** – 30%
3. **Полнота реализации (все обязательные модули включены)** – 20%
4. **Оформление и читаемость кода** – 10%

6. Кадровое обеспечение образовательного процесса

ФИО	Место работы, должность	Ученая степень/ звание	Высшее и (или) дополнительное образование в ИТ сфере	Модули (темы, лекции)	Стаж педагогической работы в образовательной организации высшего образования/ работы в компании ИТ сферы	Дополнительная информация (реализованные проекты и (или) исследования, и (или) патенты, связанные с разработкой, внедрением, развитием цифровых

						технологий
Апурин Николай Андреевич	генеральный директор ООО «АРТВЕЛЛ»		2009 - МИФИ факультет Б, комплексная информационная безопасность автоматизируемых систем, специалитет	Работа на сервере (30 часов: лекции и практические занятия)	-/ 17 лет	
Яценко Дмитрий Владимирович	Преподаватель Центра "Пуск" МФТИ; ООО "ЦОА", главный инженер учебного центра		1994 – НГТУ, Системы автоматизированного проектирования, специалитет 2001 – ДГТУ, Системы автоматизированного проектирования, аспирантура 6 курсов повышения квалификации	Работа на сервере (40 часов: лекции, практические занятия)	11 лет/ 17 лет	+

7. Материально-техническое обеспечение реализации рабочей программы

Наименование специализированных аудиторий, кабинетов, лабораторий	Вид занятий	Наименование оборудования, программного обеспечения
Система дистанционного обучения	Лекция	Слушателю необходимо наличие доступа в сеть интернет, компьютер. Преподавателю курса необходимо наличие доступа администратора курса и оборудование для проведения дистанционных семинаров (вебинаров), качественный отказоустойчивый доступ в сеть интернет.
Система дистанционного обучения	Практическая работа	Слушателю необходимо наличие доступа в сеть интернет, компьютер. Преподавателю необходимо оборудование для проведения дистанционных семинаров (вебинаров), качественный отказоустойчивый доступ в сеть интернет.

Система дистанционного обучения	Самостоятельная работа	Слушателю необходимо наличие доступа в сеть интернет, компьютер.
Система дистанционного обучения	Промежуточная (итоговая) аттестация	Слушателю необходимо наличие доступа в сеть интернет, компьютер. Преподавателю необходимо оборудование для проведения дистанционных семинаров (вебинаров), качественный отказоустойчивый доступ в сеть интернет.

8. Информационное обеспечение реализации рабочей программы

Основная литература

1. Бек К. Экстремальное программирование: разработка через тестирование. Издательство “Питер”, 2020. - 224 с.
2. Блох, Д. Java Эффективное программирование / Д. Блох. - М.: Лори, 2016. - 440 с.
3. Java: руководство для начинающих, 9-е изд./Герберт Шилдт; пер. с англ. Ю.Н. Артеменко. — Киев. : “Науковий Світ”, 2023. — 752 с.
4. Хорстманн Кей С. Java. Библиотека профессионала. Том 1. Основы / Хорстманн Кей С. - М.: Диалектика / Вильямс, 2020 г. - 864 с.
5. Хорстманн Кей С. Java. Библиотека профессионала. Том 2. Расширенные средства программирования, 11-е изд.: Пер с англ. - СПб.: / Диалектика, 2020 г. - 1008 с.
6. Эккель К. Философия Java. 4-е полное изд. Издательство “Питер”, 2024. - 1168 с.

Дополнительная литература

1. Доуни А. Алгоритмы и структуры данных. Извлечение информации на языке Java. Издательство “Питер”, 2018. - 240 с.
2. Дубаков А.А. Введение в объектно-ориентированное программирование на Java: учебное пособие [Электронный ресурс]. – СПб, Университет ИТМО, 2016. – 250 с. – Режим доступа: <http://books.ifmo.ru/file/pdf/2139.pdf>
3. Дэвид, Флэнаган JavaScript. Подробное руководство / Флэнаган Дэвид. - М.: Символ-плюс, 2017. - 555 с.
4. Перри, Брюс У. Java сервлеты и JSP: сб. рецептов, 2009 г. - 768 с.
5. Кожомбердиева, Г. И. Программирование на языке Java: многопоточные приложения : учебное пособие / Г. И. Кожомбердиева. — Санкт-Петербург: ПГУПС, 2012. — 44 с.
6. Хорстманн, Кей С. Java SE 8. Вводный курс / Хорстманн Кей С. - М.: Диалектика / Вильямс, 2014. - 898 с.

Ресурсы информационно-телекоммуникационной сети "Интернет"

1. Руководство по языку программирования Java <https://metanit.com/java/tutorial/>
2. Электронный ресурс <https://www.codewars.com/>
3. Java и Вы [Электронный ресурс]. – Режим доступа: <https://www.java.com/ru/>

Работа с фреймворками (Spring)

1. Область применения рабочей программы

Рабочая программа «Работа с фреймворками (Spring)» является частью дополнительной профессиональной программы профессиональной переподготовки ИТ-профиля «Программирование на Java: создание современных веб-приложений» и направлена на изучение основ Spring Framework и его ключевых компонентов, включая Spring Core и принципы работы с бинами и контекстами. Обучающиеся освоят Spring Boot, разработку RESTful сервисов, работу с базами данных с использованием Spring Data JPA, а также методы обработки и трансформации данных.

Освоение рабочей программы является обязательным для всех обучающихся по Программе.

2. Структура и краткое содержание рабочей программы

№ п/п	Наименование модуля, разделов и тем	Содержание обучения, наименование и тематика практических занятий (семинаров), самостоятельной работы	Объем, ак.час.
5.	Модуль 5. Работа с фреймворками (Spring)		60
5.1	Введение в Spring Framework	Лекции Что такое Spring Framework Архитектура Spring Framework Настройка окружения для работы с Spring Конфигурация Spring Пример простого Spring-приложения Практическая работа Выполнение заданий на программирование. Самостоятельная работа Подготовка к семинарам и текущему контролю успеваемости. Самостоятельное выполнение заданий по теме лекции. Изучение учебной и научной литературы по теме лекции.	4
5.2	Spring Core: Бины и контексты	Лекции Понятие Bean в Spring Конфигурация Bean Внедрение зависимостей (Dependency Injection) ApplicationContext и BeanFactory События и listeners в Spring Практическая работа Выполнение заданий на программирование. Самостоятельная работа Подготовка к семинарам и текущему контролю успеваемости. Самостоятельное выполнение заданий по теме лекции. Изучение учебной и научной литературы по теме лекции.	4

5.3	Введение в Spring Boot	Лекции Что такое Spring Boot? Создание первого Spring Boot-приложения Конфигурация Spring Boot Встроенные серверы и инструменты Тестирование Spring Boot-приложений Практическая работа Выполнение заданий на программирование. Самостоятельная работа Подготовка к семинарам и текущему контролю успеваемости. Самостоятельное выполнение заданий по теме лекции. Изучение учебной и научной литературы по теме лекции.	6
5.4	Разработка RESTful сервисов с Spring Boot	Лекции Основы REST Создание REST-контроллеров Обработка данных в REST-сервисах Обработка исключений в REST-сервисах Документирование REST API Практическая работа Семинар по теме “Java после 17: ключевые тренды развития языка” Самостоятельная работа Подготовка к семинарам и текущему контролю успеваемости. Самостоятельное выполнение заданий по теме лекции. Изучение учебной и научной литературы по теме лекции.	7
5.5	Spring Data JPA: Работа с базами данных	Лекции Введение в Spring Data JPA Сущности и репозитории Работа с данными Транзакции в Spring Data JPA Оптимизация работы с базой данных Практическая работа Выполнение заданий на программирование. Практико-ориентированный семинар по теме “Java после 17: ключевые тренды развития языка” Самостоятельная работа Подготовка к семинарам и текущему контролю успеваемости. Самостоятельное выполнение заданий по теме лекции. Изучение учебной и научной литературы по теме лекции.	10
5.6	Обработка данных и трансформации	Лекции Основы обработки данных Spring Batch	10

		Интеграция с внешними системами Трансформация данных Оптимизация и тестирование Практическая работа Выполнение заданий на программирование. Семинар по теме “Project Loom - виртуальные потоки” Самостоятельная работа Подготовка к семинарам и текущему контролю успеваемости. Самостоятельное выполнение заданий по теме лекции. Изучение учебной и научной литературы по теме лекции.	
5.7	Практика	Примеры кодирования	8
	Итоговая аттестация	Защита проекта	11

3. Учебно-тематический план рабочей программы

№ п/п	Наименование и краткое содержание структурного элемента (раздела) Программы	Количество часов			
		аудиторных		самостоятельной работы	
		всего, часов	практические занятия	всего, часов	практические занятия
5.1	Введение в Spring Framework	1		2	
5.2	Spring Core: Бины и контексты	1		2	
5.3	Введение в Spring Boot	1		3	
5.4	Разработка RESTful сервисов с Spring Boot	1		4	
5.5	Spring Data JPA: Работа с базами данных	4	2	4	2
5.6	Обработка данных и трансформации	6	2	4	2
5.7	Практика: примеры и показ кода	4	4	4	4
5.8	Итоговая аттестация по программе		4	7	7

4. Контроль и оценка результатов освоения рабочей программы

Итоговая аттестация по программе является обязательной.

Форма итоговой аттестации предусматривает зачет в форме тестирования.

Оценка «зачтено» выставляется обучающемуся, если он показал всесторонние, систематизированные, глубокие знания учебной программы дисциплины и умение уверенно применять их на практике при решении конкретных задач, свободное и правильное обоснование принятых решений.

Оценка «не зачтено» выставляется обучающемуся, который не знает большей части основного содержания учебной программы дисциплины, допускает грубые ошибки в формулировках основных понятий дисциплины и не умеет использовать полученные знания при решении типовых практических задач.

Максимальная сумма, которую можно набрать, успешно выполнив все контрольные мероприятия, составляет 100 баллов. Для получения положительной оценки «зачтено» необходимо набрать не менее 30 баллов.

В зачетно-экзаменационную ведомость оценка выставляется в соответствии с нижеприведенной таблицей 51.

Таблица 51

Сумма баллов	Оценка
30-100	Зачтено
менее 30	Не зачтено

Составляющие процесса обучения, которые оцениваются в ходе обучения, и их вклад в итоговую оценку представлены в таблице 52.

Таблица 52

№ п/п	Основные показатели оценки	Вклад в итоговую оценку
1	Практические занятия	30%
2	Выполнение домашних заданий	40%
3	Промежуточная (итоговая) аттестация	30%

В процессе обучения студенты получают регулярную обратную связь, включающую следующие формы взаимодействия:

- Практические синхронные занятия: преподаватель проводит детальный разбор допущенных ошибок в заданиях.
- Проверка домашних заданий с подробными комментариями в системе дистанционного обучения (LMS).
- Асинхронная коммуникация: преподаватели оперативно отвечают на возникающие вопросы учащихся посредством кураторов курса.
- Промежуточные тестирования: регулярно проводятся контрольные испытания, позволяющие оценить уровень усвоенного материала и прогресс каждого студента.

4.1. Примеры оценочных средств

Оценочные материалы

Наименование модуля, разделов и тем	Основные показатели оценки	Формы и методы контроля и оценки	Вес задания, %
1. Введение в Spring Framework	Компетенция 335	Выполнение заданий на программирование.	2,6
2. Spring Core: Бины и контексты	Компетенция 335	Выполнение заданий на программирование.	2,6
3. Введение в Spring Boot	Компетенция 335	Выполнение заданий на программирование.	2,7

Наименование модуля, разделов и тем	Основные показатели оценки	Формы и методы контроля и оценки	Вес задания, %
4. Разработка RESTful сервисов с Spring Boot	Компетенция 335	Выполнение заданий на программирование.	2,7
5. Spring Data JPA: Работа с базами данных	Компетенции 335, 32	Выполнение заданий на программирование.	2,7
6. Обработка данных и трансформации	Компетенции 335, 32	Выполнение заданий на программирование.	2,7
Промежуточная аттестация	Компетенции 335, 32	Защита проекта	16

Примеры заданий

Тестовые задания

1. **Что такое Spring Framework?**

- a) Библиотека для работы с базами данных.
- b) Фреймворк для создания веб-приложений на Java.
- c) Язык программирования.
- d) Инструмент для тестирования кода.

Ответ: b) Фреймворк для создания веб-приложений на Java.

2. **Какие из перечисленных модулей входят в Spring Framework? (Выберите несколько вариантов)**

- a) Spring Core
- b) Spring MVC
- c) Spring Data
- d) Spring Security

Ответ: a) Spring Core, b) Spring MVC, c) Spring Data, d) Spring Security.

3. **Spring Framework поддерживает только создание веб-приложений. (True/False)**

Ответ: False. Spring Framework поддерживает создание различных типов приложений, включая standalone, веб-приложения и микросервисы.

4. **Сопоставьте модули Spring с их назначением:**

- 1. Spring Core
- 2. Spring MVC
- 3. Spring Data
- 4. Spring Security

- a) Обработка веб-запросов и создание веб-приложений.
- b) Управление зависимостями и IoC-контейнер.
- c) Работа с базами данных и репозиториями.
- d) Обеспечение безопасности приложений.

Ответ:

- 1 - b) Управление зависимостями и IoC-контейнер.
- 2 - a) Обработка веб-запросов и создание веб-приложений.

3 - c) Работа с базами данных и репозиториями.

4 - d) Обеспечение безопасности приложений.

5. Какой принцип лежит в основе Spring Framework?

a) Dependency Injection (Внедрение зависимостей)

b) Single Responsibility Principle (Принцип единственной ответственности)

c) Model-View-Controller (MVC)

d) Object-Oriented Programming (ООП)

Ответ: a) Dependency Injection (Внедрение зависимостей).

6. Что такое IoC-контейнер в Spring?

a) Контейнер для управления зависимостями и жизненным циклом объектов.

b) Контейнер для хранения данных.

c) Контейнер для управления потоками выполнения.

d) Контейнер для работы с базами данных.

Ответ: a) Контейнер для управления зависимостями и жизненным циклом объектов.

7. Какие из перечисленных аннотаций используются для настройки Spring-приложения? (Выберите несколько вариантов)

a) @Component

b) @Service

c) @Repository

d) @Controller

Ответ: a) @Component, b) @Service, c) @Repository, d) @Controller.

8. Spring Framework использует только XML для конфигурации приложения. (True/False)

Ответ: False. Spring Framework поддерживает как XML, так и аннотации для конфигурации.

9. Сопоставьте аннотации Spring с их назначением:

1. @Component

2. @Service

3. @Repository

4. @Controller

a) Обозначает класс как компонент бизнес-логики.

b) Обозначает класс как компонент уровня доступа к данным.

c) Обозначает класс как компонент уровня представления.

d) Обозначает класс как универсальный компонент.

Ответ:

1 - d) Обозначает класс как универсальный компонент.

2 - a) Обозначает класс как компонент бизнес-логики.

3 - b) Обозначает класс как компонент уровня доступа к данным.

4 - c) Обозначает класс как компонент уровня представления.

10. Какой интерфейс используется для получения контекста Spring?

a) ApplicationContext

b) BeanFactory

c) SpringContext

d) ContextLoader

Ответ: а) ApplicationContext.

5. Образцы учебно-методических материалов для обучающихся и преподавателей

Задание на программирование

Описание:

Создайте простое Spring MVC приложение, которое будет обрабатывать запросы и управлять списком задач. Каждая задача должна иметь уникальный идентификатор, название и статус выполнения.

Требования:

1. Создайте Maven проект и настройте его для работы с Spring MVC.
2. Создайте модель Task с полями:
 - id (уникальный идентификатор)
 - title (название задачи)
 - completed (статус выполнения задачи)
3. Создайте контроллер TaskController с методами для обработки следующих запросов:
 - GET /tasks — возвращает список всех задач.
 - GET /tasks/{id} — возвращает задачу по её идентификатору.
 - POST /tasks — создаёт новую задачу.
 - PUT /tasks/{id} — обновляет существующую задачу.
 - DELETE /tasks/{id} — удаляет задачу по её идентификатору.
4. Используйте in-memory хранилище (например, HashMap или ArrayList) для хранения задач.
5. Настройте Spring MVC в файле конфигурации (applicationContext.xml или Java-конфигурация).
6. Добавьте базовую валидацию для полей задачи (например, название задачи не должно быть пустым).

Шаги для реализации:

1. Создайте Maven проект и добавьте зависимости для Spring MVC и Jackson (для работы с JSON).
2. Настройте DispatcherServlet в web.xml или используйте Java-конфигурацию.
3. Создайте класс Task (модель).
4. Создайте контроллер TaskController с методами для обработки запросов.
5. Используйте аннотации @Controller, @RequestMapping, @RequestParam, @PathVariable, @ResponseBody и другие.
6. Настройте in-memory хранилище для задач (например, Map<Integer, Task>).
7. Добавьте базовую валидацию (например, с использованием аннотации @NotNull или ручной проверки).

Ожидаемый результат: рабочее Spring MVC приложение, которое предоставляет RESTful API для управления списком задач.

6. Кадровое обеспечение образовательного процесса

ФИО	Место работы, должность	Учебная степень/степень	Высшее и (или) дополнительное образование в ИТ сфере	Модули (темы, лекции)	Стаж педагогической работы	Дополнительная информация

		звание			в образовательной организации высшего образования/работы в компании ИТ сферы	ация (реализованные проекты и (или) исследования, и (или) патенты, связанные с разработкой, внедрением, развитием цифровых технологий
Апурин Николай Андреевич	генеральный директор ООО «АРТВЕЛЛ»		2009 - МИФИ факультет Б, комплексная информационная безопасность автоматизируемых систем, специалитет	Работа с фреймворками (Spring) (37 часов: лекции и практические занятия)	-/ 17 лет	
Вареница Михаил	Android разработчик, Ozon Tech		2024 — Московский инженерно-физический институт, Институт Интеллектуальных Кибернетических Систем, Кафедра 12 ИВТ (информатика и вычислительная техника), магистратура 2022 — Московский государственный технический университет	Работа с фреймворками (Spring) (20 часов: лекции и практические занятия)	-/4 года	

			имени Н.Э. Баумана, Информатики и систем управления; ИУ, Кафедра ИУ9 (прикладная математика и информатика), бакалавриат			
--	--	--	---	--	--	--

7. Материально-техническое обеспечение реализации рабочей программы

Наименование специализированных аудиторий, кабинетов, лабораторий	Вид занятий	Наименование оборудования, программного обеспечения
Система дистанционного обучения	Лекция	Слушателю необходимо наличие доступа в сеть интернет, компьютер. Преподавателю курса необходимо наличие доступа администратора курса и оборудование для проведения дистанционных семинаров (вебинаров), качественный отказоустойчивый доступ в сеть интернет.
Система дистанционного обучения	Практическая работа	Слушателю необходимо наличие доступа в сеть интернет, компьютер. Преподавателю необходимо оборудование для проведения дистанционных семинаров (вебинаров), качественный отказоустойчивый доступ в сеть интернет.
Система дистанционного обучения	Самостоятельная работа	Слушателю необходимо наличие доступа в сеть интернет, компьютер.
Система дистанционного обучения	Промежуточная (итоговая) аттестация	Слушателю необходимо наличие доступа в сеть интернет, компьютер. Преподавателю необходимо оборудование для проведения дистанционных семинаров (вебинаров), качественный отказоустойчивый доступ в сеть интернет.

8. Информационное обеспечение реализации рабочей программы

Основная литература

1. Бек К. Экстремальное программирование: разработка через тестирование. Издательство “Питер”, 2020. - 224 с.
2. Блох, Д. Java Эффективное программирование / Д. Блох. - М.: Лори, 2016. - 440 с.
3. Java: руководство для начинающих, 9-е изд./Герберт Шилдт; пер. с англ. Ю.Н. Артеменко. —Киев. : “Науковий Світ”, 2023. — 752 с.
4. Хорстманн Кей С. Java. Библиотека профессионала. Том 1. Основы / Хорстманн Кей С. - М.: Диалектика / Вильямс, 2020 г. - 864 с.
5. Хорстманн Кей С. Java. Библиотека профессионала. Том 2. Расширенные средства программирования, 11-е изд.: Пер с англ. - СПб.: / Диалектика, 2020 г. - 1008 с.
6. Эккель К. Философия Java. 4-е полное изд. Издательство “Питер”, 2024. - 1168 с.

Дополнительная литература

1. Доуни А. Алгоритмы и структуры данных. Извлечение информации на языке Java. Издательство “Питер”, 2018. - 240 с.
2. Дубаков А.А. Введение в объектно-ориентированное программирование на Java: учебное пособие [Электронный ресурс]. – СПб, Университет ИТМО, 2016. – 250 с. – Режим доступа: <http://books.ifmo.ru/file/pdf/2139.pdf>
3. Дэвид, Флэнаган JavaScript. Подробное руководство / Флэнаган Дэвид. - М.: Символ-плюс, 2017. - 555 с.
4. Перри, Брюс У. Java сервлеты и JSP: сб. рецептов, 2009 г. - 768 с.
5. Кожомбердиева, Г. И. Программирование на языке Java: многопоточные приложения : учебное пособие / Г. И. Кожомбердиева. — Санкт-Петербург: ПГУПС, 2012. — 44 с.
6. Хорстманн, Кей С. Java SE 8. Вводный курс / Хорстманн Кей С. - М.: Диалектика / Вильямс, 2014. - 898 с.

Ресурсы информационно-телекоммуникационной сети "Интернет"

1. Руководство по языку программирования Java <https://metanit.com/java/tutorial/>
2. Электронный ресурс <https://www.codewars.com/>
3. Java и Вы [Электронный ресурс]. – Режим доступа: <https://www.java.com/ru/>

Производственная практика

1. Область применения рабочей программы

Рабочая программа «Производственная практика» является частью дополнительной профессиональной программы профессиональной переподготовки ИТ-профиля «Программирование на Java: создание современных веб-приложений» и направлена на закрепление и углубление теоретических знаний, полученных студентами при изучении курса и дальнейшее совершенствование навыков практической работы.

Освоение рабочей программы является обязательным для всех обучающихся по Программе.

2. Структура и краткое содержание рабочей программы

№ п/п	Наименование модуля, разделов и тем	Содержание обучения, наименование и тематика практических занятий (семинаров), самостоятельной работы	Объем, ак.час.
6	<i>Модуль 6. Производственная практика</i>		16
6.1	Практика: выполнение проекта	Выполнение и защита проекта	8
6.2	Практика: выполнение проекта	Выполнение и защита проекта	8

3. Учебно-тематический план рабочей программы

№ п/п	Наименование и краткое содержание структурного элемента (раздела) Программы	Количество часов			
		аудиторных		самостоятельной работы	
		всего, часов	практические занятия	всего, часов	практические занятия
6.1	Практика: выполнение проекта	8	8		
6.2	Практика: выполнение проекта	8	8		

4. Контроль и оценка результатов освоения рабочей программы

Итоговая аттестация по программе является обязательной.

Форма итоговой аттестации предусматривает зачет в форме тестирования.

Оценка «зачтено» выставляется обучающемуся, если он показал всесторонние, систематизированные, глубокие знания учебной программы дисциплины и умение уверенно применять их на практике при решении конкретных задач, свободное и правильное обоснование принятых решений.

Оценка «не зачтено» выставляется обучающемуся, который не знает большей части основного содержания учебной программы дисциплины, допускает грубые ошибки в формулировках основных понятий дисциплины и не умеет использовать полученные знания при решении типовых практических задач.

Максимальная сумма, которую можно набрать, успешно выполнив все контрольные мероприятия, составляет 100 баллов. Для получения положительной оценки «зачтено» необходимо набрать не менее 30 баллов.

В зачетно-экзаменационную ведомость оценка выставляется в соответствии с нижеприведенной таблицей 64.

Таблица 64

Сумма баллов	Оценка
30-100	Зачтено
менее 30	Не зачтено

Составляющие процесса обучения, которые оцениваются в ходе обучения, и их вклад в итоговую оценку представлены в таблице 65.

Таблица 65

№ п/п	Основные показатели оценки	Вклад в итоговую оценку
1	Практические занятия	30%
2	Выполнение домашних заданий	40%
3	Промежуточная (итоговая) аттестация	30%

4.1. Примеры оценочных средств

Оценочные материалы

Наименование модуля, разделов и тем	Основные показатели оценки	Формы и методы контроля и оценки	Вес задания, %
Практика: выполнение проекта	Компетенции 28, 333, 32, 335	Защита проекта	8
Практика: выполнение проекта	Компетенции 28, 333, 32, 335	Защита проекта	8

5. Образцы учебно-методических материалов для обучающихся и преподавателей

Пример проектного задания

Этап 1. Проектирование базы данных

1. Создание базы данных

1. Откройте MySQL Workbench.
2. Напишите команду для создания базы данных с названием, например online_shop:

2. Создание таблиц

В интерфейсе MySQL Workbench напишите запросы на создание необходимых таблиц. Пример таблиц и столбцов указан ниже. Можно разработать свой набор таблиц и полей. При создании таблиц по шаблону обратите внимание на типы данных, связи первичный – внешний ключи в таблицах и отношения. Особого обращения требуют таблицы cart и order_items, в них используются составные первичные ключи. В примере указан минимальный набор полей в таблицах, можно добавить по своему усмотрению.

1. users – Пользователи. Поля: id, name, email, password, role.
2. categories – Категории. Поля: id, name
3. products – Товары. Поля: id, name, description, price, category_id
4. cart – Корзина. Поля: user_id, product_id, quantity
5. orders – Заказы. Поля: id, user_id, order_date, status
6. order_items – Элементы заказа. Поля: order_id, product_id, quantity, price

Этап 2. Реализация моделей (Java Beans)

Следующим шагом в разработке проекта будет создание моделей. Реализация моделей (Java Beans) — это основа структурирования данных в приложении. Каждый класс модели соответствует таблице в БД. Модели упрощают чтение/запись данных из БД и позволяют работать с данными как с объектами, а не сырыми SQL-запросами.

Каждая модель должна называться также как и таблица в БД, иметь те же свойства, что и поля в БД, конструктор и геттеры-сеттеры для каждого из свойств.

Модели можно создать в папке: src/main/java/com/example/beans/

Разберем создание модели на примере таблицы users:

Класс User

Назначение: хранит данные пользователя.

Поля:

- id — уникальный идентификатор,
- name — имя пользователя,
- email — электронная почта (логин),
- password — пароль (после хеширования),
- role — роль (USER или ADMIN).

Заготовка для реализации класса User:

package com.example.beans;

```
public class User {  
    // для каждого поля в таблице приватное свойство  
    private int id;  
    private String name;  
    private String email;  
    private String password;  
    private String role;  
  
    // Конструктор по умолчанию  
    public User() {}  
  
    // Геттеры и сеттеры для работы с приватными свойствами  
    public int getId() { return id; }  
    public void setId(int id) { this.id = id; }  
  
    // остальные геттеры и сеттеры  
}
```

Ниже приведен полный список классов, которые необходимо описать

Создать классы:

- User (id, name, email, password, role),
- Category (id, name),

- Product (id, name, description, price, categoryId),
- CartItem (userId, productId, quantity),
- Order (id, userId, orderDate, status),
- OrderItem (orderId, productId, quantity, price).

Этап 3. Реализация DAO

1. Настройка подключения к БД

Для работы с БД нужно настроить подключение. Создайте класс DBConnection для соединения с БД. Его нужно разместить в каталоге: src/main/java/com/example/dao/. Для подключения к БД реализуйте метод DBConnection, который будет возвращать объект подключения к базе данных.

2. Описание интерфейсов

Далее нужно создать интерфейсы, в которых будет указано, какие методы должны быть реализованы в классах. Интерфейсы нужны для того, чтобы **определять контракт**, который классы должны выполнять, не указывая, **как именно** это будет сделано. Создайте интерфейсы согласно коду ниже, если есть изменения по структуре данных, внесите их самостоятельно

Общая структура DAO

- **Интерфейс:** определяет методы для работы с сущностью.
- **Класс реализации:** использует JDBC для выполнения SQL-запросов.
- **Используемые технологии:** JDBC, MySQL Connector.

Создайте интерфейсы, указанные в списке ниже:

1. UserDAO

Интерфейс должен содержать методы:

- **boolean** register(User user);
- User findByEmail(String email);
- **boolean** updateUserRole(**int** userId, String role);

2. CategoryDAO

Интерфейс должен содержать методы:

- **boolean** addCategory(Category category);
- **boolean** deleteCategory(**int** categoryId);
- List<Category> getAllCategories();

3. ProductDAO

Интерфейс должен содержать методы:

- **boolean** addProduct(Product product);
- **boolean** deleteProduct(**int** productId);
- List<Product> getProductsByCategoryId(**int** categoryId);

4. CartDAO

Интерфейс должен содержать методы:

- **boolean** addToCart(CartItem item);
- **boolean** removeFromCart(**int** userId, **int** productId);
- List<CartItem> getCartItems(**int** userId);

5. OrderDAO

Интерфейс должен содержать методы:

- **boolean** createOrder(Order order, List<CartItem> cartItems);
- List<Order> getOrderHistory(int userId);

3. Создание классов DAO для работы с БД

На этом шаге необходимо написать реализацию вышеописанных интерфейсов. В каждом интерфейсе указаны методы, которые должны быть реализованы в классах. Для каждого интерфейса нужно написать свой класс, которые реализует конкретный интерфейс и методы, которые в интерфейсе указаны. Каждый метод без исключения, указанный в интерфейсе, должен быть реализован в соответствующем классе.

Создайте классы, указанные в списке ниже, классы должны реализовывать соответствующие интерфейсы:

- UserDaoImpl
- CategoryDaoImpl
- ProductDaoImpl
- CartDaoImpl
- OrderDaoImpl

Этап 4. Реализация сервлетов (контроллеров)

Задача - создать сервлеты в директории: src/main/java/com/example/servlets/

Ниже указан список сервлетов, которые необходимо реализовать и url, которые они должны обрабатывать.

Методы *doGet* используются для отображения страниц, *doPost* — для обработки действий (добавление, удаление, отправка форм).

1. AuthServlet - Обрабатывает аутентификацию и авторизацию.

Методы:

- doPost(HttpServletRequest, HttpServletResponse)
 - Маршруты:
 - /register → регистрация пользователя (создание записи в БД).
 - /login → вход в аккаунт (проверка email и пароля).
- doGet(HttpServletRequest, HttpServletResponse)
 - Маршрут: /logout → выход из аккаунта.

2. ProductServlet

Отображает товары и категории.

Методы:

- doGet(HttpServletRequest, HttpServletResponse)
 - Маршруты:
 - /products → список всех товаров.
 - /products?categoryId=1 → товары по категории.

- `/product?id=1` → детальная информация о товаре.

3. CartServlet

Управляет корзиной пользователя.

Методы:

- `doPost(HttpServletRequest, HttpServletResponse)`
 - Маршруты:
 - `/cart/add` → добавление товара в корзину.
 - `/cart/remove` → удаление товара из корзины.
- `doGet(HttpServletRequest, HttpServletResponse)`
 - Маршрут: `/cart` → просмотр корзины.

4. OrderServlet

Оформляет заказы и показывает историю.

Методы:

- `doPost(HttpServletRequest, HttpServletResponse)`
 - Маршрут: `/order/create` → оформление заказа.
- `doGet(HttpServletRequest, HttpServletResponse)`
 - Маршрут: `/orders` → история заказов.

5. AdminServlet

Управление товарами и категориями (только для администраторов).

Методы:

- `doPost(HttpServletRequest, HttpServletResponse)`
 - Маршруты:
 - `/admin/category/add` → добавление категории.
 - `/admin/category/delete` → удаление категории.
 - `/admin/product/add` → добавление товара.
 - `/admin/product/delete` → удаление товара.
- `doGet(HttpServletRequest, HttpServletResponse)`
 - Маршруты:
 - `/admin/categories` → список категорий для управления.
 - `/admin/products` → список товаров для управления.

Этап 5. Создание JSP-страниц (Views)

Задача - создать представления (JSP-страницы) создавать в директории: `src/main/webapp/`

Ниже указан список представлений, которые необходимо создать для реализации frontend-части приложения.

1. Общие элементы

Эти элементы должны быть во всех страницах для единообразия.

- Хедер - шапка сайта:
 - Меню навигации: "Главная", "Корзина", "История заказов".
 - Условные блоки:

- Для неавторизованных: "Войти" / "Зарегистрироваться".
- Для пользователей: "Выйти".
- Для администраторов: "Управление товарами", "Управление категориями".
- Футер - подвал: Контактная информация, копирайт.

2. Страницы пользователя

2.1. login.jsp - страница для входа на сайт, должна содержать форму для входа.

- Назначение: Форма входа.
- Содержание:
 - Поля: Email, Пароль.
 - Кнопка "Войти".
 - Ссылка на регистрацию.
 - Блок для вывода ошибок (если неверный логин/пароль).

2.2. register.jsp - страница для регистрации, должна содержать форму регистрации.

- Назначение: Форма регистрации.
- Содержание:
 - Поля: Имя, Email, Пароль.
 - Кнопка "Зарегистрироваться".
 - Блок для ошибок (если email уже занят).

2.3. products.jsp - страница просмотра списка товаров.

- Назначение: Список товаров с фильтрацией по категориям.
- Содержание:
 - Список категорий (боковая панель или сверху).
 - Карточки товаров: название, цена, кнопка "В корзину".
 - Ссылка на детали товара.

2.4. product-details.jsp - страница для детального просмотра товара.

- Назначение: Детальная информация о товаре.
- Содержание:
 - Название, описание, цена, категория.
 - Кнопка "Добавить в корзину".

2.5. cart.jsp - страница просмотра корзины покупателя и оформления заказа.

- Назначение: Просмотр и редактирование корзины.
- Содержание:
 - Таблица с товарами: название, цена, количество, сумма.
 - Кнопки: "Удалить", "Оформить заказ".
 - Общая сумма заказа.

2.6. orders.jsp - страница просмотра истории заказов пользователя

- Назначение: История заказов пользователя.
- Содержание:
 - Список заказов: дата, статус, сумма.
 - Детали заказа при клике (список товаров).

3. Страницы администратора

3.1. admin/categories.jsp - страница для управления категориями для администратора.

- Назначение: Управление категориями.
- Содержание:
 - Форма добавления категории.
 - Список существующих категорий с кнопкой "Удалить".

3.2. admin/products.jsp - страница для управления товарами для администратора.

- Назначение: Управление товарами.
- Содержание:
 - Форма добавления товара (с выбором категории).
 - Список товаров с кнопкой "Удалить".

Важные замечания по реализации:

1. **Безопасность:**

- На страницах администратора необходимо добавить проверку роли, чтобы другие пользователи не смогли редактировать товары и категории.

2. **Стили:**

- Для более приятного визуального оформления необходимо создать CSS-файл style.css для всех страниц и написать для каждой страницы стили.

3. **Навигация:**

- В хедере необходимо проверять атрибут сессии user, для того чтобы отображать соответствующие ссылки вход/регистрация или выход.

Этап 6. Деплой на Apache Tomcat

- Собрать проект в WAR-файл через Maven (команда mvn clean package).
- Разместить WAR-файл в папке webapps Tomcat.
- Запустить веб-сервер, произвести развертывание проекта и проверить на работоспособность.

Этап 7. Финальное тестирование и презентация

- Проверка всех функциональных требований.
- Демонстрация работы приложения.
- Запись видео (скринкаст) с обзором функционала.

6. Кадровое обеспечение образовательного процесса

ФИО	Место работы, должность	Ученая степень/звание	Высшее и (или) дополнительное образование в ИТ сфере	Модули (темы, лекции)	Стаж педагогической работы в образовательной организации высшего	Дополнительная информация (реализованные проекты и (или))

					о образо вания/ работы в компан ии ИТ сферы	исследо вания, и (или) патент ы, связанн ые с разрабо ткой, внедрен ием, развити ем цифров ых техноло гий
Апурин Николай Андреевич	генеральный директор ООО «АРТВЕЛЛ»		2009 - МИФИ факультет Б, комплексная информационная безопасность автоматизируем ых систем, специалитет	Производственн ая практика (8 часов)	-/ 17 лет	
Вареница Михаил	Android разработчик, Ozon Tech		2024 — Московский инженерно- физический институт, Институт Интеллектуальн ых Кибернетически х Систем, Кафедра 12 ИВТ (информатика и вычислительная техника), магистратура 2022 — Московский государственны й технический университет имени Н.Э. Баумана, Информатики и систем управления; ИУ, Кафедра ИУ9 (прикладная	Производственн ая практика (8 часов)	-/4 года	

			математика и информатика), бакалавриат			
--	--	--	--	--	--	--

7. Материально-техническое обеспечение реализации рабочей программы

Наименование специализированных аудиторий, кабинетов, лабораторий	Вид занятий	Наименование оборудования, программного обеспечения
Система дистанционного обучения	Лекция	Слушателю необходимо наличие доступа в сеть интернет, компьютер. Преподавателю курса необходимо наличие доступа администратора курса и оборудование для проведения дистанционных семинаров (вебинаров), качественный отказоустойчивый доступ в сеть интернет.
Система дистанционного обучения	Практическая работа	Слушателю необходимо наличие доступа в сеть интернет, компьютер. Преподавателю необходимо оборудование для проведения дистанционных семинаров (вебинаров), качественный отказоустойчивый доступ в сеть интернет.
Система дистанционного обучения	Самостоятельная работа	Слушателю необходимо наличие доступа в сеть интернет, компьютер.
Система дистанционного обучения	Промежуточная (итоговая) аттестация	Слушателю необходимо наличие доступа в сеть интернет, компьютер. Преподавателю необходимо оборудование для проведения дистанционных семинаров (вебинаров), качественный отказоустойчивый доступ в сеть интернет.

8. Информационное обеспечение реализации рабочей программы

Основная литература

1. Бек К. Экстремальное программирование: разработка через тестирование. Издательство “Питер”, 2020. - 224 с.
2. Блох, Д. Java Эффективное программирование / Д. Блох. - М.: Лори, 2016. - 440 с.
3. Java: руководство для начинающих, 9-е изд./Герберт Шилдт; пер. с англ. Ю.Н. Артеменко. — Киев. : “Науковий Світ”, 2023. — 752 с.

4. Хорстманн Кей С. Java. Библиотека профессионала. Том 1. Основы / Хорстманн Кей С. - М.: Диалектика / Вильямс, 2020 г. - 864 с.
5. Хорстманн Кей С. Java. Библиотека профессионала. Том 2. Расширенные средства программирования, 11-е изд.: Пер с англ. - СПб.: / Диалектика, 2020 г. - 1008 с.
6. Эккель К. Философия Java. 4-е полное изд. Издательство "Питер", 2024. - 1168 с.

Дополнительная литература

1. Доуни А. Алгоритмы и структуры данных. Извлечение информации на языке Java. Издательство "Питер", 2018. - 240 с.
2. Дубаков А.А. Введение в объектно-ориентированное программирование на Java: учебное пособие [Электронный ресурс]. – СПб, Университет ИТМО, 2016. – 250 с. – Режим доступа: <http://books.ifmo.ru/file/pdf/2139.pdf>
3. Дэвид, Флэнаган JavaScript. Подробное руководство / Флэнаган Дэвид. - М.: Символ-плюс, 2017. - 555 с.
4. Перри, Брюс У. Java сервлеты и JSP: сб. рецептов, 2009 г. - 768 с.
5. Кожомбердиева, Г. И. Программирование на языке Java: многопоточные приложения : учебное пособие / Г. И. Кожомбердиева. — Санкт-Петербург: ПГУПС, 2012. — 44 с.
6. Хорстманн, Кей С. Java SE 8. Вводный курс / Хорстманн Кей С. - М.: Диалектика / Вильямс, 2014. - 898 с.

Ресурсы информационно-телекоммуникационной сети "Интернет"

1. Руководство по языку программирования Java <https://metanit.com/java/tutorial/>
2. Электронный ресурс <https://www.codewars.com/>
3. Java и Вы [Электронный ресурс]. – Режим доступа: <https://www.java.com/ru/>

VI. Хакатон

Хакатон проходит в формате соревнования с участием представителей профильных организаций работодателей. Хакатон проходит в форме защиты проекта, в ходе которого обучающиеся используют и демонстрируют цифровые компетенции, предусмотренные в процессе освоения ДПП ПП.

Защита проекта включает презентацию и доклад о ключевых этапах его реализации, а также демонстрацию разработанного прототипа.

Оценку и подведение итогов осуществляют индустриальные партнеры.

Пример проекта для выполнения и представления на хакатоне

Описание задачи

Вам предстоит разработать веб-приложение "Интернет-магазин". Оно будет включать клиентскую и серверную часть, а также базу данных. Основной функционал приложения охватывает управление пользователями, товарами, категориями, корзиной и заказами.

Вы должны использовать технологии, изученные в предыдущем модуле, включая Java, JDBC, работу с базами данных, а также закрепить полученные знания в модуле 4: сервлеты, JSP, HTML/CSS.

По итогу проекта должно получиться работоспособное веб-приложение, способное обрабатывать HTTP-запросы и взаимодействовать с базой данных.

Функциональные требования

1. Роли.

В приложении должны быть реализованы роли для пользователей и администраторов. Пользователи могут просматривать товары, оформлять заказы. Администраторы управляют содержимым магазина.

2. Функционал пользователя.

- Регистрация на сайте
- Вход на сайт
- Выход из аккаунта
- Просмотр списка товаров
- Просмотр списка категорий товаров
- Просмотр товаров по категориям
- Просмотр конкретного товара детально
- Добавление товара в корзину
- Просмотр корзины
- Удаление товара из корзины
- Оформление заказа
- Просмотр истории заказов

3. Функционал администратора.

- Доступы для администраторов прописываются вручную. Из интерфейса сайта зарегистрироваться с ролью администратора нельзя.

- Вход на сайт пользователя с правами администратора.

- Выход из аккаунта

- Добавление категорий товаров

- Добавление товаров

- Удаление товаров

Нефункциональные требования

1. Используемые технологии: Java, mysql, maven, html, css.

2. Приложение должно использовать паттерн MVC(DAO, Bean, Controller, JSP)

3. Под каждую сущность должна быть создана таблица в БД.

4. Приложение должно быть написано для запуска с помощью веб-сервера Apache tomcat.

План реализации

Для разработки проекта необходимо заранее настроить/установить JDK, IDE, Maven.

Критерии оценки

- 1) Корректность работы функционала
- 2) Качество кода и архитектуры проекта
- 3) Полнота реализации (все обязательные модули включены)
- 4) Оформление и читаемость кода

VII. Аттестация по Программе

Оценка качества освоения программы включает текущую, промежуточную и итоговую аттестацию обучающихся.

Текущая аттестация служит для оценки объёма и уровня усвоения слушателем учебного материала одного модуля программы и применяется в рамках текущего контроля успеваемости слушателя. Текущая аттестация включает прохождение студентом входного, промежуточного и итогового ассесментов Университета Иннополис.

Промежуточная аттестация слушателей включает сдачу зачетов по модулям, предусмотренным учебно-тематическим планом.

Итоговая аттестация выполняется в виде защиты обучающимся проекта.

Контроль качества знаний осуществляется преподавателем.

Текущий контроль успеваемости слушателей проводится на платформе дистанционного обучения и на платформе Университета Иннополис.

Объектом оценивания выступают: формируемые знания и умения.

Итоговая аттестация слушателей проводится в сроки, предусмотренные учебным планом и графиком учебного процесса.

Форма и условия аттестационных испытаний определяются настоящей программой.

Защита итогового проекта проводится слушателями после получения допуска к итоговой аттестации. Доступ к прохождению итоговой аттестации предоставляется слушателям, успешно прошедшим промежуточную аттестацию по всем модулям программы.

Результаты итоговых аттестационных испытаний определяются отметками «отлично», «хорошо», «удовлетворительно», «неудовлетворительно» и объявляются после оформления в установленном порядке протоколов заседаний итоговой аттестационной комиссии.

Оценка «отлично (5)» выставляется обучающемуся, если он показал всесторонние, систематизированные, глубокие знания учебной программы дисциплины и умение уверенно применять их на практике при решении конкретных задач, свободное и правильное обоснование принятых решений;

оценка «хорошо (4)» выставляется обучающемуся, если он твердо знает материал, грамотно и по существу излагает его, умеет применять полученные знания на практике, но допускает неточности в ответе или делает несущественные ошибки при решении задач;

оценка «удовлетворительно (3)» выставляется обучающемуся, показавшему фрагментарный, разрозненный характер знаний, неточные формулировки базовых понятий, нарушения логической последовательности в изложении программного материала, если при этом он владеет основными разделами учебной программы, необходимыми для дальнейшего обучения и может применять полученные знания по образцу в стандартной ситуации;

оценка «неудовлетворительно (2)» выставляется обучающемуся, который не знает большей части основного содержания учебной программы дисциплины, допускает грубые ошибки в формулировках основных понятий дисциплины и не умеет использовать полученные знания при решении типовых практических задач.

В случае несогласия с результатами итоговой аттестации слушатель имеет право подать апелляцию в аттестационную комиссию на имя председателя.

Апелляция принимается в день сдачи итогового экзамена сразу после объявления оценок.

В ходе рассмотрения апелляции проверяется правильность и аргументированность выставленной оценки. При необходимости ИАК принимает решение о возможности пересдачи итогового экзамена или повторной защиты выпускной квалификационной

работы. Итоги рассмотрения апелляции вносятся в соответствующий протокол в раздел «Особое мнение комиссии».

ИАК рассматривает апелляцию по итоговым экзаменам и сообщает свое решение не позднее, чем через сутки с момента подачи апелляции. При принятии решения о возможности пересдачи итогового экзамена комиссией анализируется лист ответа слушателя и при необходимости ведется беседа по дополнительным вопросам. Решение ИАК по апелляции является окончательным.

Составляющие процесса обучения, которые оцениваются в ходе обучения, и их вклад в итоговую оценку представлены в таблице ниже.

Таблица 91

	Основные показатели оценки	Вклад в итоговую оценку, %
1	Промежуточная аттестация по модулю 1	16
2	Промежуточная аттестация по модулю 2	16
3	Промежуточная аттестация по модулю 3	16
4	Промежуточная аттестация по модулю 4	16
5	Промежуточная аттестация по модулю 5	16
6	Промежуточная аттестация по модулю 6	16
8	Итоговая аттестация	20

Самостоятельная работа слушателей подразделяется на аудиторную и внеаудиторную. Аудиторную самостоятельную работу составляют практические задания, которые выполняются слушателями во время учебных занятий, результаты ее выполнения проверяются и оцениваются преподавателем в учебном процессе.

Внеаудиторная самостоятельная работа включает формы: изучение дополнительной литературы, подготовка итоговых проектов по модулям, подготовка проекта к итоговой аттестации.

Основными критериями качества организации самостоятельной работы служит наличие контроля результатов самостоятельной работы.

Основными современными формами организации самостоятельной работы слушателей являются творческие работы и работа с информационными компьютерными технологиями.

VII.1. Темы и задания для демонстрационного экзамена

Итоговая аттестация обучающихся проходит в формате демонстрационного экзамена с участием представителей профильных организаций работодателей. Итоговая аттестация проходит в форме защиты проекта, в ходе которого обучающиеся используют и демонстрируют цифровые компетенции, предусмотренные в процессе освоения ДПП ПП.

Демонстрационный экзамен – аттестационное испытание, предусматривающее выполнение профессиональных задач и оценку результатов и (или) процесса выполнения профессиональных задач для подтверждения применения обучающимися цифровых компетенций на практике.

Задания демонстрационного экзамена разрабатываются с участием организаций-работодателей, отраслевых партнеров и профессиональных сообществ.

Защита проекта сопровождается презентацией и докладом об основных этапах реализации проекта, а также демонстрацией прототипа проектного решения.

Перечень тем и показатели оценки

Примеры тем	Основные показатели оценки	Формы и методы контроля и оценки	Вес задания, %
Работа с базами данных	Компетенции 32	Защита проекта	15
Работа на сервере	Компетенции 333, 335	Защита проекта	20
Работа с фреймворками (Spring)	Компетенции 335, 32	Защита проекта	30
Java Core	Компетенции 335, 32	Защита проекта	30
Микросервисы и паттерны проектирования ПО	Компетенции 335, 32	Защита проекта	30

Формат представления результатов

Необходимо подготовить презентацию решения одного из проектов (в формате MS PowerPoint, Google Презентаций или другом аналогичном), после чего записать видеопрезентацию своего решения с использованием программы записи демонстрации экрана.

Критерии оценивания

1. Работоспособность и стабильность функционирования.

Продукт демонстрирует стабильную работу и функциональность в рамках установленных условий эксплуатации.

Показатель: отсутствие критичных ошибок и сбоев в процессе тестирования и демонстрации функционала.

2. Обоснованность технических решений.

Предложенное техническое решение имеет ясное обоснование выбора архитектуры, технологий и инструментов разработки.

Показатель: четкое изложение аргументов, почему были выбраны конкретные технологии и подходы, с указанием преимуществ и ограничений каждого из них.

Пример проектов для выполнения и представления на демонстративном экзамене

Проектная задача 1

Название проекта: Разработка новостного блога на Java с использованием MySQL или PostgreSQL и работы с файлами.

Цель проекта: Создать приложение для управления и публикации новостного блога, взаимодействующее с пользователем через текстовые файлы. Новости и категории хранятся в базе данных PostgreSQL

Краткое описание: Разработать приложение, управляющее новостным порталом.

Пример задания проекта.

Этап 1: Настройка окружения

1. **Установка окружения разработки:**
 - Установите **JDK 21** или более новую версию с официального сайта Oracle.
 - Установите текстовый редактор или IDE, например **IntelliJ IDEA**, Eclipse, или Visual Studio Code.
2. **Установка и настройка базы данных:**
 - Установите PostgreSQL или MySQL:
 - PostgreSQL:
 - Скачайте и установите PostgreSQL с официального сайта.
 - Во время установки настройте учетную запись администратора и запомните пароль.
 - MySQL:
 - Установите сервер MySQL.
 - Настройте учетную запись администратора.
 - Создайте базу данных для проекта:
 - Назовите базу данных `news_blog`.
3. **Создание таблиц в базе данных:**
 - Подключитесь к СУБД с помощью командной строки или интерфейса управления (pgAdmin для PostgreSQL, MySQL Workbench для MySQL).
 - Выполните следующие SQL-запросы:

```
CREATE TABLE Category (
    id SERIAL PRIMARY KEY,
    name VARCHAR(255) NOT NULL UNIQUE
);

CREATE TABLE Article (
    id SERIAL PRIMARY KEY,
    title VARCHAR(255) NOT NULL,
    summary TEXT,
    content TEXT,
    publication_date DATE NOT NULL,
    category_id INT REFERENCES Category(id) ON DELETE SET NULL
);
```

4. **Подключение к базе данных из Java:**
 - Загрузите JDBC-драйвер для выбранной базы данных:
 - PostgreSQL: скачайте `postgresql-<версия>.jar` с сайта PostgreSQL.
 - MySQL: скачайте `mysql-connector-java-<версия>.jar` с сайта MySQL.
 - Сохраните `.jar` файл в папке проекта, например, в `lib`.
5. **Создание базовой структуры проекта:**
 - Создайте папку проекта, например `NewsBlogProject`.
 - Внутри папки создайте следующую структуру:

```
NewsBlogProject/
├── lib/           # Для библиотек (JDBC-драйвер)
├── src/          # Исходный код
├── input/        # Входные файлы
├── output/       # Выходные файлы
└── Main.java     # Точка входа в приложение
```

Этап 2: Реализация модели (Model)

1. **Создание сущностей:**
 - Создайте папку `src/model` для классов модели.

- Реализуйте два класса: Category и Article.

2. Класс Category:

- Создайте файл src/model/Category.java и напишите следующий код:

```
package model;

public class Category {
    private int id;
    private String name;

    // Конструкторы
    public Category() {}

    public Category(String name) {
        this.name = name;
    }

    public Category(int id, String name) {
        this.id = id;
        this.name = name;
    }

    // Геттеры и сеттеры
    public int getId() { return id; }
    public void setId(int id) { this.id = id; }

    public String getName() { return name; }
    public void setName(String name) { this.name = name; }
}
```

3. Класс Article:

- Создайте файл src/model/Article.java и напишите следующий код:

```
package model;

import java.time.LocalDate;

public class Article {
    private int id;
    private String title;
    private String summary;
    private String content;
    private LocalDate publicationDate;
    private int categoryId;

    // Конструкторы
    public Article() {}

    public Article(String title, String summary, String content,
        LocalDate publicationDate, int categoryId) {
        this.title = title;
        this.summary = summary;
        this.content = content;
        this.publicationDate = publicationDate;
        this.categoryId = categoryId;
    }

    public Article(int id, String title, String summary, String
        content, LocalDate publicationDate, int categoryId) {
        this.id = id;
        this.title = title;
        this.summary = summary;
    }
}
```

```

        this.content = content;
        this.publicationDate = publicationDate;
        this.categoryId = categoryId;
    }

    // Геттеры и сеттеры
    public int getId() { return id; }
    public void setId(int id) { this.id = id; }

    public String getTitle() { return title; }
    public void setTitle(String title) { this.title = title; }

    public String getSummary() { return summary; }
    public void setSummary(String summary) { this.summary =
summary; }

    public String getContent() { return content; }
    public void setContent(String content) { this.content =
content; }

    public LocalDate getPublicationDate() { return
publicationDate; }
    public void setPublicationDate(LocalDate publicationDate) {
this.publicationDate = publicationDate; }

    public int getCategoryId() { return categoryId; }
    public void setCategoryId(int categoryId) { this.categoryId =
categoryId; }
}

```

Этап 3: Реализация слоя DAO

1. Создайте интерфейсы DAO:

- Создайте папку src/dao.
- Напишите интерфейсы CategoryDAO и ArticleDAO.

2. Интерфейс CategoryDAO:

- Создайте файл src/dao/CategoryDAO.java:

```

package dao;

import model.Category;
import java.util.List;

public interface CategoryDAO {
    void addCategory(Category category);
    void updateCategory(int id, String newName);
    void deleteCategory(int id);
    List<Category> getAllCategories();
    Category getCategoryById(int id);
}

```

3. Интерфейс ArticleDAO:

- Создайте файл src/dao/ArticleDAO.java:

```

package dao;

import model.Article;
import java.util.List;

public interface ArticleDAO {
    void addArticle(Article article);
}

```

```

        void updateArticle(Article article);
        void deleteArticle(int id);
        List<Article> getAllArticles();
        Article getArticleById(int id);
    }

```

4. Реализация DAO-классов:

- Создайте папку `src/dao/impl` для реализации DAO.
- Реализуйте `CategoryDAOImpl` и `ArticleDAOImpl`, используя JDBC.

5. Класс `DatabaseConnection`:

- Создайте файл `src/dao/DatabaseConnection.java` для управления подключением:

```

package dao;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;

public class DatabaseConnection {
    private static final String URL =
        "jdbc:postgresql://localhost:5432/news_blog";
    private static final String USER = "username";
    private static final String PASSWORD = "password";

    public static Connection getConnection() throws SQLException {
        return DriverManager.getConnection(URL, USER, PASSWORD);
    }
}

```

Этап 4: Работа с файлами

1. Создание пакета для обработки файлов:

- Создайте папку `src/service`.

2. Реализация обработки статей:

- Напишите класс `ArticleFileProcessor` для работы с файлом `input_articles.txt`:
 - Чтение данных из файла.
 - Вызов DAO-методов для выполнения операций.

3. Реализация обработки категорий:

- Напишите класс `CategoryFileProcessor` для работы с файлом `input_categories.txt`.

4. Вывод результатов:

- Реализуйте запись итогов операций в файлы `output_articles.txt` и `output_categories.txt`.

Этап 5: Главный класс приложения

1. Создайте `Main.java`:

- Напишите меню для выбора операций:
 - Обработка категорий.
 - Обработка статей.

2. Пример вызова:

```

public static void main(String[] args) {
    // Вызов файловых процессоров
}

```

```

        new CategoryFileProcessor().process();
        new ArticleFileProcessor().process();
    }

```

Рекомендации по выполнению

1. Проверьте настройку окружения и убедитесь, что все компоненты настроены и работают корректно.
2. Убедитесь, что правильно выбран и настроен драйвер.
3. Проверьте корректность реализации интерфейсов DAO.
4. Убедитесь в качестве кода: чистота и поддерживаемость кода, наличие комментариев.
5. Проверьте исключения и обработку ошибок: правильность обработки исключений во всех ситуациях, когда это необходимо (работа с файлами, с БД, с внешними серверами).

Проектная задача 2

Название проекта: Лямбда-выражения и Stream API

Цель проекта: закрепить полученные знания на практике

Этап 1: Основы лямбда-выражений

Создайте функциональный интерфейс Operation, который будет представлять одну операцию, принимающую два целых числа и возвращающую результат в виде целого числа. Реализуйте в методе main три операции с использованием лямбда-выражений:

1. Сложение двух чисел.
2. Вычитание двух чисел.
3. Умножение двух чисел.

Программа должна запрашивать у пользователя ввод двух целых чисел, выполнять операции и выводить результаты в консоль.

Требования:

- Создайте функциональный интерфейс Operation с одним методом apply(int a, int b).
- В методе main запросите у пользователя ввод двух целых чисел.
- Используйте лямбда-выражения для реализации операций сложения, вычитания и умножения.
- Выведите результаты операций в консоль в формате:

Ввод	Результат
5 10	Сумма: 15 Разность: -5 Произведение: 50
30 66	Сумма: 96 Разность: -36 Произведение: 1980
100 60	Сумма: 160 Разность: 40

	Произведение: 6000
--	--------------------

Этап 2: Использование функциональных интерфейсов с коллекциями

Создайте список строк с именами пяти людей, данные для списка нужно ввести с клавиатуры. Используйте методы `forEach`, `removeIf` и `map` с лямбда-выражениями для выполнения следующих задач:

1. Выведите все имена в верхнем регистре.
2. Удалите из списка все имена, длина которых меньше 4 символов.
3. Преобразуйте оставшиеся имена в их длину и выведите результат на экран.

Требования:

1. Программа должна запрашивать у пользователя ввод пяти имен.
2. Используйте метод `forEach` и лямбда-выражение для вывода всех имен в верхнем регистре.
3. Используйте метод `removeIf` и лямбда-выражение для удаления имен, длина которых меньше 4 символов.
4. Используйте метод `map` и лямбда-выражение для преобразования имен в их длину и вывода результата.

Ввод	Результат
<i>John Alice Bob Jane Tom</i>	Имена в верхнем регистре: JOHN ALICE BOB JANE TOM Длина оставшихся имен: 4 5 4
<i>Инна Николай Владимир Ольга Игорь</i>	Имена в верхнем регистре: ИННА НИКОЛАЙ ВЛАДИМИР ОЛЬГА ИГОРЬ Длина оставшихся имен: 4 7 8 5 5
<i>Bob Sam Bill Sarah Jane</i>	Имена в верхнем регистре: BOB SAM BILL SARAH

	JANE Длина оставшихся имен: 4 5 4
--	---

Этап 3: Использование Predicate, Function, Consumer

Создайте список из пяти целых чисел, данные для списка нужно ввести с клавиатуры. Используйте функциональные интерфейсы Predicate, Function и Consumer для выполнения следующих задач:

1. Используйте Predicate<Integer> для фильтрации всех четных чисел и выведите их на экран.
2. Используйте Function<Integer, String> для преобразования чисел в строки, добавляя префикс "Number: " перед каждым числом.
3. Используйте Consumer<Integer> для вывода каждого числа в консоль в квадрате (например, 4 → 16).

Требования:

1. Программа должна запрашивать у пользователя ввод пяти целых чисел.
2. Используйте Predicate<Integer> для фильтрации четных чисел.
3. Используйте Function<Integer, String> для преобразования чисел в строки с префиксом.
4. Используйте Consumer<Integer> для вывода квадратов чисел.

Ввод	Результат
1 2 3 4 5	2 4 Number: 1 Number: 2 Number: 3 Number: 4 Number: 5 1 4 9 16 25
6 7 8 9 10	6 8 10 Number: 6 Number: 7 Number: 8 Number: 9 Number: 10 36 49

	64 81 100
43 56 51 43 67	56 Number: 43 Number: 56 Number: 51 Number: 43 Number: 67 1849 3136 2601 1849 4489

Этап 4: Реализация пользовательского функционального интерфейса

Создайте функциональный интерфейс StringTransformer, который будет принимать строку и возвращать преобразованную строку. Реализуйте несколько лямбда-выражений для:

1. Преобразования строки в верхний регистр.
2. Добавления суффикса "_SUFFIX" к строке.
3. Удаления пробелов в строке.

Напишите метод applyTransformation(String input, StringTransformer transformer), который применяет одну из этих операций. Программа должна запрашивать у пользователя ввод трех строк и применять к каждой из них соответствующую операцию.

Требования:

1. Создайте функциональный интерфейс StringTransformer с методом transform(String input).
2. Реализуйте три лямбда-выражения для:
 - Преобразования строки в верхний регистр.
 - Добавления суффикса "_SUFFIX".
 - Удаления пробелов в строке.
3. Напишите метод applyTransformation, который принимает строку и объект StringTransformer, и возвращает преобразованную строку.
4. Программа должна запрашивать у пользователя ввод трех строк и применять к каждой из них соответствующую операцию.

Ввод	Результат
<i>"hello world" "hello world" "hello world"</i>	HELLO WORLD hello world_SUFFIX helloworld
<i>я программирую "на java"</i>	Я программирую_SUFFIX наjava
<i>Однажды "в студеную" "зимнюю пору"</i>	ОДНАЖДЫ в студеную_SUFFIX

	зимнюю пору
--	-------------

Рекомендации по выполнению

- Проверьте настройку окружения и убедитесь, что все компоненты настроены и работают корректно.
- Убедитесь, что правильно выбран и настроен драйвер.
- Проверьте корректность реализации интерфейсов.
- Убедитесь в качестве кода: чистота и поддерживаемость кода, наличие комментариев.
- Проверьте исключения и обработку ошибок: правильность обработки исключений во всех ситуациях, когда это необходимо (работа с файлами, с БД, с внешними серверами).

Проектная задача 3

Название проекта: Введение в работу с сетью: Класс InetAddress и работа с IP-адресами

Цель проекта: закрепить полученные знания на практике

Этапы работы

1. Поиск IP-адресов по имени хоста:

Напишите программу, которая принимает имя хоста (например, "www.google.com") и выводит его IP-адреса.

- Используйте класс InetAddress для получения всех IP-адресов для данного имени хоста.
- Реализуйте обработку ошибок для случаев, когда имя хоста не существует или не доступно.

2. Проверка доступности хоста:

Создайте программу, которая проверяет, доступен ли хост по его IP-адресу с использованием метода isReachable() из класса InetAddress.

- Программа должна отправлять пинг на адрес и выводить, доступен ли он, с указанием времени отклика.

3. Решение проблемы с DNS:

Напишите программу, которая проверяет существование DNS-записи для заданного домена. Программа должна:

- Запросить имя хоста.
- Использовать InetAddress.getByName() для получения информации о DNS-записи.
- Обработать исключения, если DNS-запись отсутствует.

Рекомендации по выполнению

- Проверьте настройку окружения и убедитесь, что все компоненты настроены и работают корректно.
- Убедитесь, что правильно выбран и настроен драйвер.
- Проверьте корректность реализации интерфейсов.
- Убедитесь в качестве кода: чистота и поддерживаемость кода, наличие комментариев.

- Проверьте исключения и обработку ошибок: правильность обработки исключений во всех ситуациях, когда это необходимо (работа с файлами, с БД, с внешними серверами).

Проектная задача 4

Название проекта: Введение в работу с сетью: Протокол TCP и работа с сокетами

Цель проекта: закрепить полученные знания на практике

Этапы работы

1. Создание TCP-сервера и клиента:

Напишите программу, которая создает TCP-сервер и клиент:

- Сервер должен слушать на порту 8080 и принимать соединения.
- Клиент должен подключаться к серверу и отправлять строку.
- Сервер должен принимать сообщение и отправлять обратно клиенту.

2. Многопоточный TCP-сервер:

Реализуйте многопоточный сервер, который может обрабатывать несколько клиентов одновременно:

- Сервер должен запускать новый поток для каждого подключающегося клиента, чтобы обрабатывать его запросы независимо.
- Каждый клиент должен отправлять серверу строку, а сервер возвращает строку в ответ.

3. Чтение и запись через сокеты:

Напишите программу, которая позволяет клиенту отправить несколько строк на сервер через TCP-соединение. Сервер должен читать строки, изменять их (например, перевести в верхний регистр) и отправить обратно клиенту.

Рекомендации по выполнению

- Проверьте настройку окружения и убедитесь, что все компоненты настроены и работают корректно.
- Убедитесь, что правильно выбран и настроен драйвер.
- Проверьте корректность реализации интерфейсов.
- Убедитесь в качестве кода: чистота и поддерживаемость кода, наличие комментариев.
- Проверьте исключения и обработку ошибок: правильность обработки исключений во всех ситуациях, когда это необходимо (работа с файлами, с БД, с внешними серверами).

Проектная задача 5

Название проекта: Введение в работу с сетью: Протокол UDP и работа с датаграммами

Цель проекта: закрепить полученные знания на практике

Этапы работы

1. UDP-сервер и клиент:

Напишите простую программу, использующую UDP для обмена сообщениями:

- Сервер должен слушать на порту 9876 и принимать сообщения от клиента.
- Клиент должен отправить сообщение на сервер, а сервер возвращает его обратно.

2. Многократные сообщения через UDP:

Реализуйте UDP-сервер и клиент, которые отправляют и принимают несколько сообщений.

- Сервер должен принимать сообщения от клиента и отвечать на каждое сообщение.
- Клиент должен отправлять несколько сообщений подряд и выводить ответы от сервера.

3. Обработка ошибок в UDP:

Создайте UDP-сервер, который обрабатывает пакеты с ошибками:

- Сервер должен проверять, если сообщение содержит ошибку (например, неправильный формат данных) и отправлять ответ с ошибкой.
- Клиент должен отправить сообщение с возможными ошибками и проверить корректность ответа.

Рекомендации по выполнению

- Проверьте настройку окружения и убедитесь, что все компоненты настроены и работают корректно.
- Убедитесь, что правильно выбран и настроен драйвер.
- Проверьте корректность реализации интерфейсов.
- Убедитесь в качестве кода: чистота и поддерживаемость кода, наличие комментариев.
- Проверьте исключения и обработку ошибок: правильность обработки исключений во всех ситуациях, когда это необходимо (работа с файлами, с БД, с внешними серверами).

VIII. Завершение обучения по Программе

Лицам, завершившим обучение по Программе и достигших целевого уровня сформированности цифровых компетенций по результатам итоговой оценки и прошедших итоговую аттестацию, присваивается дополнительная ИТ-квалификация, установленная Программой. При освоении Программы параллельно с получением высшего образования диплом о профессиональной переподготовке выдается не ранее получения соответствующего документа об образовании и о квалификации (за исключением лиц, имеющих среднее профессиональное или высшее образование). Лицам, не прошедшим итоговую аттестацию или получившим на итоговой аттестации неудовлетворительные результаты, а также лицам, освоившим часть Программы и (или) отчисленным из образовательной организации высшего образования, реализующей Программу, выдается справка об обучении или о периоде обучения по образцу, самостоятельно устанавливаемому образовательной организацией высшего образования.