ЦЕНТРАЛЬНЫЙ УНИВЕРСИТЕТ

УТВЕРЖДЕНА

Решением Ученого совета
АНО ВО «Центральный университет»
«24» июня 2025 г.
Протокол № 2

Рабочая программа дисциплины (модуля)
«Разработка на Java»

Направление подготовки: *02.03.01 Математика и компьютерные науки*

Направленность (профиль) подготовки: *Разработка*

Квалификация (степень) выпускника: *бакалавр*

Форма обучения: *очная*

Срок освоения программы: *4 года*

Год набора: *2025*

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ

Сертификат: 0215B0C000D1B3FEB44A37EE631E0C27D4
Владелец: Ивашкевич Евгений Васильевич
Действителен: с 13.01.2026 до 13.04.2027

Электронный документ

Москва
2025

Содержание

1. **Краткая характеристика дисциплины (модуля)** 3
2. **Перечень планируемых результатов обучения** 5
3. **Тематический план** 7
4. **Содержание дисциплины (модуля)** 7
5. **Учебно-методическое обеспечение** 8
6. **Материально-техническое обеспечение** 8
7. **Методические и оценочные материалы** 10

1. Краткая характеристика дисциплины (модуля)

Рабочая программа дисциплины (модуля) «Разработка на Java» составлена в соответствии с федеральным государственным образовательным стандартом высшего образования – бакалавриат по направлению 02.03.01 «Математика и компьютерные науки», профиль «Разработка», утвержденный приказом Министерства науки и высшего образования Российской Федерации № 1015 от 13.08.2020 года.

Изучение дисциплины (модуля) «Разработка на Java» позволяет сформировать у обучающихся умения эффективно проектировать, мониторить и защищать информационные системы, отвечая современным требованиям к надёжности и кибербезопасности. Это критически важно для построения отказоустойчивой и безопасной сетевой инфраструктуры в любой организации.

Место дисциплины (модуля) в структуре образовательной программы

Настоящая дисциплина (модуль) включена в учебный план по программе подготовки бакалавриата по направлению 02.03.01 «Математика и компьютерные науки», профиль «Разработка» и входит в вариативную часть Блока 1, формируемую участниками образовательных отношений, как дисциплина по выбору.

Дисциплина (модуль) изучается на 3 курсе в 5 семестре, доступна для прохождения при условии успешного завершения дисциплины (модуля) «Основы промышленной разработки».

Цель изучения дисциплины (модуля): освоить разработку полноценных сетевых приложений и сервисов на Java, способных реализовывать, анализировать и защищать коммуникации в соответствии с современными протоколами и принципами сетевой безопасности.

Задачи изучения дисциплины (модуля):

— изучить модель взаимодействия открытых систем OSI и стек протоколов TCP/IP, используя Java-библиотеки для работы с сокетами;

— реализовать и протестировать основные прикладные протоколы (HTTP, HTTPS, DNS, FTP, SMTP) в Java-приложениях;

— смоделировать механизмы маршрутизации и протоколы RIP, OSPF, BGP с помощью Java-фреймворков и библиотек;

— создать и настроить защищённые соединения (SSL/TLS) и внедрить шифрование данных в Java-сервисах;

— интегрировать Java-клиенты с инструментами анализа трафика (Wireshark, tcpdump) для выявления и предотвращения типовых сетевых атак.

В результате освоения дисциплины (модуля) обучающийся должен:

знать:

— синтаксис языка Java и особенности объектно-ориентированного программирования;

— назначение и особенности классов, объектов, пакетов, статических членов, интерфейсов и абстрактных классов;

— принципы наследования, полиморфизма и инкапсуляции;

— основы обработки исключений и ввода-вывода данных;

— основы многопоточности, HTTP, REST API и работы с базами данных;

уметь:

— разрабатывать программы на языке Java с использованием классов и объектов;

— применять механизмы наследования, полиморфизма, интерфейсов и абстрактных классов;

— использовать Maven для подключения библиотек и управления зависимостями;

— разрабатывать многопоточные программы;

— писать модульные тесты;

владеть:

— навыками разработки программ на языке Java;

— навыками применения объектно-ориентированного подхода при разработке программ;

— навыками написания модульных тестов;

— навыками работы с HTTP, REST API и базами данных;

— навыками применения паттернов проектирования.

2. Перечень планируемых результатов обучения

Компетенции, формируемые в результате освоения дисциплины (модуля) при проведении учебных занятий в форме контактной работы обучающихся с педагогическими работниками Университета и в форме самостоятельной работы обучающихся:

Компетенция	Содержание компетенции	Индикатор компетенции	Перечень планируемых результатов обучения по дисциплине (модулю)
УК-1.	Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач	УК-1.1.	Знает методы поиска и анализа информации в области разработки, основные принципы критической оценки источников информации и их релевантности.
		УК-1.2.	Умеет критически оценивать источники информации и синтезировать данные из различных источников для решения задач, применять системный подход к анализу и решению комплексных проблем
		УК-1.3.	Имеет практический опыт работы с современными инструментами и технологиями для обработки информации, формулировании и структурировании задач на основе полученной информации
ОПК-1.	Способен консультировать и использовать фундаментальные знания в области математического анализа, комплексного и функционального анализа алгебры, аналитической геометрии, дифференциальной геометрии и топологии, дифференциальных уравнений, дискретной математики и математической логики, теории вероятностей, математической статистики и случайных процессов, численных методов, теоретической механики в профессиональной деятельности	ОПК-1.1.	Знает основные концепции и теории в области математического анализа и смежных дисциплин; методы и подходы, используемые в различных областях математики
		ОПК-1.2.	Умеет применять математические методы для решения профессиональных задач
		ОПК-1.3.	Имеет практический опыт разработки и реализация математических моделей в профессиональной деятельности
ОПК-4.	Способен находить, анализировать, реализовывать программно и использовать на практике математические алгоритмы, в том числе с применением современных вычислительных систем	ОПК-4.1.	Знает базовые основы современного математического аппарата, связанного с проектированием, разработкой, реализацией и оценкой качества программных продуктов и программных комплексов в различных областях человеческой деятельности

		ОПК-4.2.	Умеет использовать этот математический аппарат в профессиональной деятельности
		ОПК-4.3.	Имеет практический опыт применения современного математического аппарата, связанного с проектированием, разработкой, реализацией и оценкой качества программных продуктов и программных комплексов в различных областях человеческой деятельности
ОПК-6.	Способен разрабатывать алгоритмы и компьютерные программы, пригодные для практического применения	ОПК-6.1.	Знает алгоритмы разработки, компьютерные программы, а также алгоритмы вычислительной математики
		ОПК-6.2.	Умеет разрабатывать математические программные продукты и комплексы с использованием современных технологий программирования
		ОПК-6.3.	Имеет практический опыт разработки интеллектуальных информационных систем для визуализации результатов исследований
ПК-3	Способен использовать методы математического и алгоритмического моделирования при решении теоретических и прикладных задач	ПК-3.1.	Знает основные методы математического и алгоритмического моделирования, а также их применение для решения теоретических и прикладных задач
		ПК-3.2.	Умеет разрабатывать и применять математические модели и алгоритмы для решения различных задач, анализируя полученные результаты
		ПК-3.3.	Имеет практический опыт использования методов математического и алгоритмического моделирования в реальных проектах или исследованиях

3. Тематический план

№ п/п	Наименование раздела дисциплины (модуля)	Трудоемкость, академические часы				ТКУ (текущий контроль успеваемости)
		Очная форма				
		Аудиторная работа		Контроль	Самостоятельная работа	
		Лекции	Семинары (практические занятия)			
1.	Введение в Java: основные особенности языка + ООП	12	10	2	42	Домашнее задание, Контрольная работа
2.	Обработка данных и функциональный стиль	8	6	2	42	Домашнее задание, Контрольная работа
3.	Серверная разработка и финальный проект	10	6	4	42	Домашнее задание, Контрольная работа, Проект
	Зачет с оценкой			**4**		
	Итого:	***30***	***22***	***12***	***126***	
Объем дисциплины (модуля) (в ак. ч.)		***190***				
Объем дисциплины (модуля) (в зач. ед.)		***5***				

4. Содержание дисциплины (модуля)

№п/п	Наименование раздела дисциплины (модуля)	Содержание дисциплины (модуля) по темам
1.	Введение в Java: основные особенности языка + ООП	Введение в Java. Синтаксис для знающих C и Python. Классы и объекты. Память Инкапсуляция. Пакеты. Статические члены. Специальные виды классов Наследование и полиморфизм. Абстрактные классы и интерфейсы Maven. Подключение библиотек. Основы SOLID Исключения. Ввод-вывод
2.	Обработка данных и функциональный стиль	Обобщения Коллекции Лямбды и Stream API Юнит-тестирование. CI
3.	Серверная разработка и финальный проект	Многопоточность: основы Многопоточность: продолжение HTTP и REST API Работа с БД Паттерны проектирования: часть 1 Паттерны проектирования: часть 2

5. Учебно-методическое обеспечение

Университет располагает полным набором лицензионного и свободно распространяемого программного обеспечения, включая продукты отечественного производства.

Каждый студент в течение всего периода обучения получает индивидуальный неограниченный доступ к электронно-библиотечной системе и электронной информационно-образовательной среде университета. Эти системы предоставляют возможность доступа к ресурсам из любой точки, где есть подключение к сети Интернет, как на территории университета, так и за его пределами.

Студентам обеспечен удаленный доступ к современным профессиональным базам данных и информационным справочным системам.

Основная литература:

1. Долженко, А. И. Разработка приложений на платформе Java EE : учебное пособие / А. И. Долженко. - Ростов-на-Дону : Издательско-полиграфический комплекс Ростовского государственного экономического университета (РИНХ), 2024. - 98 с. - ISBN 978-5-7972-3315-2. - Текст : электронный. - URL: https://znanium.ru/catalog/product/2214527.

2. Бобырь, М. В. Программирование на языке Java. Практический курс : учебное пособие / М.В. Бобырь. — Москва : ИНФРА-М, 2025. — 189 с. — (Высшее образование). — DOI 10.12737/2160989. - ISBN 978-5-16-020136-8. - Текст : электронный. - URL: https://znanium.ru/catalog/product/2160989.

Дополнительная литература:

1. Леонард, А. Java. Решение практических задач : практическое руководство / А. Леонард. - Санкт-Петербург : БХВ-Петербург, 2021. - 720 с. - ISBN 978-5-9775-6719-0. - Текст : электронный. - URL: https://znanium.ru/catalog/product/2139141.

2. Урванов, Ф. В. Java. Состояние языка и его перспективы : практическое руководство / Ф. В. Урванов. - Санкт-Петербург : БХВ-Петербург, 2023. - 368 с. - ISBN 978-5-9775-1689-1. - Текст : электронный. - URL: https://znanium.com/catalog/product/2122950.

6. Материально-техническое обеспечение

Университет располагает материально-технической базой, соответствующей действующим противопожарным правилам и нормам и обеспечивающей проведение всех видов дисциплинарной и междисциплинарной подготовки, практической и научно-исследовательской работ обучающихся, предусмотренных учебным планом.

Помещения, которые представляют собой учебные аудитории для проведения занятий лекционного типа, занятий семинарского (практического) типа, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации, а также помещения для самостоятельной работы и помещения для хранения и профилактического обслуживания учебного оборудования. Помещения укомплектованы специализированной мебелью и техническими средствами обучения, служащими для представления учебной информации большой аудитории.

Изучение дисциплины (модуля) обеспечивается в учебных аудиториях, оснащенных:

— столами и стульями;

— компьютерной техникой;

— механическими калькуляторами;

— специализированным оборудованием, включая демонстрационное оборудование.

Помещения для самостоятельной работы обучающихся, в том числе приспособленные для использования инвалидами и лицами с ограниченными возможностями здоровья, оснащены компьютерной техникой с возможностью подключения к сети «Интернет» и

обеспечением доступа в электронную информационно-образовательную среду Университета.

Обучающимся предоставляется доступ (в том числе удаленный) к ресурсам информационно-телекоммуникационной сети «Интернет», электронным ресурсам (в том числе электронным библиотечным системам, современным профессиональным базам данных и информационным справочным системам):

№	Наименование портала (издания, курса, документа)	Ссылка
1.	Научная электронная библиотека elibrary.ru библиотека	https://elibrary.ru/defaultx.asp
2.	База данных для IT-специалистов	https://habr.com
3.	База данных ScienceDirect	https://www.sciencedirect.com
4.	Официальный сайт Министерства науки и высшего образования Российской Федерации	https://minobrnauki.gov.ru/
5.	Федеральный портал «Российское образование"	https://www.edu.ru/
6.	Информационная система "Единое окно доступа к образовательным ресурсам"	http://window.edu.ru/
7.	Единая коллекция цифровых образовательных ресурсов	http://school-collection.edu.ru/
8.	Федеральный центр информационно - образовательных ресурсов	http://fcior.edu.ru/

Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине (модулю), в том числе комплект лицензионного программного обеспечения, современные профессиональные базы данных и информационные справочные системы:

Наименование ПО	Производство	Лицензионное / свободно распространяемое
Операционные системы:		
Microsoft Imagine (Windows Client, Server)	зарубежное	лицензионное
Браузеры:		
Яндекс.Браузер	отечественное	свободно распространяемое
Google Chrome	зарубежное	свободно распространяемое
Офисные приложения:		
Microsoft Imagine (Visio, OneNote)	зарубежное	лицензионное
TeXstudio	зарубежное	свободно распространяемое
Adobe Acrobat Reader	зарубежное	свободно распространяемое
Программное обеспечение для планирования и учета времени:		
Toggle app	зарубежное	свободно распространяемое
Системы управления проектами:		
Microsoft Imagine (Project)	зарубежное	лицензионное
Системы управления базами данных:		

Microsoft Imagine (SQL Server)	зарубежное	лицензионное
Системы резервного копирования (backup):		
Acronis Backup Advanced for HyperV	зарубежное	лицензионное
Справочно-правовые системы:		
КонсультантПлюс: справочно-правовая система	отечественное	лицензионное
Средства антивирусной защиты:		
Kaspersky Endpoint Security для бизнеса Стандартный Russian Edition	отечественное	лицензионное
Среды разработки:		
Visual Studio Code	зарубежное	свободно распространяемое
Bash (Unix shell)	зарубежное	свободно распространяемое
Anaconda	зарубежное	свободно распространяемое
Robotic Operating System	зарубежное	свободно распространяемое
CopelliaSim	зарубежное	свободно распространяемое
Google Colaboratory	зарубежное	свободно распространяемое
Пакеты программных средств и библиотек:		
AutoPsy	зарубежное	свободно распространяемое
Interactive Disassembler (IDA)	зарубежное	свободно распространяемое
Системы управления библиографической информацией:		
Zotero	зарубежное	свободно распространяемое
Сервисы и службы:		
Bind	зарубежное	свободно распространяемое
Docker	зарубежное	свободно распространяемое

7. Методические и оценочные материалы

Методические указания для обучающихся по освоению дисциплины (модуля)

В процессе изучения дисциплины (модуля) «Разработка на Java» в рамках текущего контроля успеваемости используются такие виды учебной работы, как лекции, семинары, коллоквиум, проект, а также различные виды самостоятельной работы обучающихся по заданию преподавателя, направленные на развитие навыков профессиональной лексики, закрепление практических профессиональных компетенций, поощрение инициатив.

Лекция – систематическое, последовательное, монологическое изложение преподавателем учебного материала, как правило, теоретического характера.

В процессе лекций рекомендуется вести конспект лекций: кратко и схематично фиксировать основные идеи, выводы и обобщения лекции; выделять важные мысли,

ключевые слова и термины. Необходимо отметить вопросы или материалы, которые вызывают затруднения, и попытаться найти ответы в рекомендованной литературе. Если разобраться в материале не удается, следует сформулировать вопрос и задать его преподавателю на консультации или во время семинарского (практического) занятия.

Семинар — это форма учебной деятельности, проводимая в учебном заведении под руководством преподавателя, где студенты активно участвуют в обсуждениях, практических заданиях и других формах взаимодействия.

Для успешной подготовки к семинару рекомендуется заранее ознакомиться с темой занятия и основными материалами, чтобы иметь возможность активно участвовать в обсуждении. Также полезно подготовить вопросы и идеи для обсуждения, что поможет глубже понять материал и продемонстрировать заинтересованность.

Домашнее задание – набор задач по темам недели.

При работе над домашними заданиями важно внимательно ознакомиться с требованиями и сроками выполнения. Рекомендуется разбивать задания на этапы, чтобы избежать перегрузки и лучше усвоить материал, использовать различные источники информации, включая учебники и онлайн-ресурсы, для более глубокого понимания темы.

Контрольная работа – письменная работа с набором задач, которые нужно решить за ограниченное время.

Цель контрольной работы – получить специальные знания по одной или нескольким темам дисциплины и продемонстрировать навыки их практического применения.

Проект – исследовательская работа по дисциплине (модулю) и презентация результатов.

Для успешной подготовки к проекту рекомендуется: четко определить цели и задачи проекта; составить план работы, разбив проект на этапы с указанием сроков выполнения каждого из них; использовать разнообразные источники информации и инструменты для исследования темы; регулярно проверять прогресс и вносить коррективы в план, если это необходимо.

Самостоятельная работа – работа студентов, направленная на углубленное изучение отдельных тем и вопросов учебной дисциплины (модуля).

В процессе самостоятельной работы студенты взаимодействуют с рекомендованными материалами при минимальном участии преподавателя. Задачи студента включают работу с конспектами лекций (обработка текста), повторное изучение учебных материалов планов и тезисов ответов, изучение дополнительных тем, выполнение учебно-исследовательских заданий и другое.

Система оценивания результатов обучения по дисциплине (модулю)

Критерии получения уровня и оценивания сформированности компетенций по дисциплине (модулю) «Разработка на Java»

Оценивание уровня учебных достижений обучающихся по дисциплине (модулю) осуществляется в виде текущего контроля успеваемости и промежуточной аттестации.

Промежуточная аттестация по дисциплине (модулю) осуществляется в форме **зачета с оценкой**, при этом проводится оценка компетенций, сформированных по дисциплине.

Для оценивания текущего контроля успеваемости и промежуточной аттестации используется десятибалльная шкала оценивания, которая соотносится с традиционной пятибалльной шкалой следующим образом, если иное не указано в силлабусе дисциплины:

Десятибалльная оценка	Пятибалльная оценка	Общая характеристика результата обучения по дисциплине (модулю)
10	Отлично	Студент полностью владеет знаниями, изложенными в рабочей программе, и глубоко осмысляет дисциплину (модуль). Он самостоятельно и логически последовательно отвечает на все вопросы, акцентируя внимание на наиболее важном. Умеет анализировать, сравнивать, классифицировать, обобщать, конкретизировать и систематизировать изученный материал, выделяя ключевые моменты и устанавливая причинно-следственные связи. Четко формулирует ответы, уверенно интерпретирует результаты анализов и других исследований, а также решает сложные задачи. Студент хорошо знаком с методами исследования, необходимыми для практической деятельности, и умеет связывать теоретические аспекты дисциплины (модуля) с практическими задачами.
9	Отлично	
8	Отлично	
7	Хорошо	Студент обладает знаниями предмета почти в полном объеме рабочей программы и самостоятельно, логически последовательно и всесторонне отвечает на все вопросы, акцентируя внимание на наиболее значимых моментах. Он умеет анализировать, сравнивать, классифицировать, обобщать, конкретизировать и систематизировать изученный материал, выделяя его ключевые аспекты и устанавливая причинно-следственные связи. Формулирует свои ответы, уверенно интерпретирует результаты анализов и других исследований, а также решает сложные ситуационные задачи. Студент хорошо знаком с методами исследования, необходимыми для практической деятельности, и умеет связывать теоретические аспекты предмета с практическими задачами.
6	Хорошо	
5	Удовлетворительно	Студент обладает базовыми знаниями по дисциплине (модулю), но испытывает трудности при самостоятельных ответах и использует неточные формулировки. В ходе ответов он допускает ошибки, касающиеся сути вопросов. Студент способен решать только самые простые задачи и владеет лишь минимальным набором методов исследования.
4	Удовлетворительно	
3	Не сдан	

Электронный документ

Десятибалльная оценка	Пятибалльная оценка	Общая характеристика результата обучения по дисциплине (модулю)
2	Не сдан	Студент не овладел обязательным минимумом знаний по предмету и не может ответить на вопросы, даже если преподаватель задает дополнительные наводящие вопросы.
1	Не сдан	

Дисциплина (модуль) «Разработка на Java» оценивается следующим образом:

Активность	Вес	Количество	Описание
Домашнее задание	20%	6	Набор задач по темам недели
Контрольная работа	20%	3	Письменная работа с набором задач, которые нужно решить за ограниченное время
Проект	30%	1	Исследовательская работа по дисциплине (модулю) и презентация результатов
Зачет с оценкой	30%	1	Письменная или устная работа над заданием, направленным на проверку полученных знаний и навыков по дисциплине (модулю)

Формула расчёта итоговой оценки по дисциплине (модулю) «Разработка на Java»: «0,2 × среднее за домашние задания + 0,2 × среднее за контрольные работы + 0,3 × Проект + 0,3 × Зачет с оценкой».

Текущий контроль успеваемости обучающихся по дисциплине (модулю)

Примерные домашние задания

Домашнее задание по теме «Введение в Java: основные особенности языка»

Задание: Консольное приложение «Управление сотрудниками»

Цель: Закрепить основы синтаксиса, ООП, работу с исключениями и сборку проекта.

Требования к реализации:

1. Структура проекта: Проект должен собираться через Maven. Код разбит по пакетам (model, service, util).
2. Классы и объекты: Создать класс Employee с полями (id, name, salary). Реализовать геттеры/сеттеры, конструкторы, toString, equals, hashCode.
3. Наследование и полиморфизм: Создать подклассы Manager и Developer. Переопределить метод calculateBonus().
4. Интерфейсы и абстрактные классы: Создать интерфейс Promotable с методом promote(). Реализовать его в классах сотрудников.
5. Инкапсуляция и static: Все поля private. Использовать static счетчик для генерации уникальных ID.
6. Исключения: Создать собственное исключение InvalidSalaryException. Выбрасывать его при установке отрицательной зарплаты.
7. Ввод-вывод: Сохранять список сотрудников в текстовый файл и читать из него при запуске.

Домашнее задание по теме «Обработка данных и функциональный стиль»

Задание: Анализ данных «Статистика заказов»

Цель: Освоить Collections, Stream API, Generics и модульное тестирование.

Требования к реализации:

1. Обобщения (Generics): Реализовать универсальный класс Box<T> для хранения объектов любого типа.

2. Коллекции: Использовать List, Set и Map для хранения данных о заказах (Order: id, customer, amount, status).

3. Stream API:

— Отфильтровать заказы суммой больше 1000.

— Сгруппировать заказы по статусам.

— Найти общую сумму заказов через reduce.

— Использовать лямбда-выражения для компараторов сортировки.

4. Юнит-тестирование: Написать тесты на JUnit 5 для сервисного слоя (проверка фильтрации, подсчета суммы).

5. CI: Подготовить конфигурационный файл (например, .github/workflows/maven.yml) для автоматического запуска тестов при пуше в репозиторий.

Домашнее задание по теме «Серверная разработка и финальный проект»

Задание: REST-сервис «Менеджер задач» (Task Tracker)

Цель: Создать полноценное веб-приложение с БД, многопоточностью и паттернами.

Требования к реализации:

1. HTTP и REST API: Реализовать endpoints (GET, POST, PUT, DELETE) для управления задачами. Использовать Spring Boot или HttpServlet.

2. Работа с БД: Подключить базу данных (PostgreSQL/MySQL). Использовать JDBC или JPA/Hibernate для сохранения задач.

3. Многопоточность:

— Реализовать отправку email-уведомлений о новых задачах в отдельном потоке.

— Использовать ExecutorService для обработки фоновых задач.

— Обеспечить потокобезопасность общих ресурсов.

4. Паттерны проектирования:

— Использовать Singleton для подключения к БД.

— Использовать Factory или Builder для создания сложных объектов задач.

— Применить Repository pattern для доступа к данным.

5. Финальная сборка: Проект должен запускаться через docker-compose (приложение + БД).

Примерные задания для контрольной работы

1. В чем различие между размещением примитивных типов и объектов в памяти (Stack vs Heap)?

2. Какие существуют модификаторы доступа и какой из них обеспечивает максимальную скрытность?

3. В чем разница между перегрузкой (overloading) и переопределением (overriding) методов?

4. Когда следует использовать абстрактный класс, а когда интерфейс?

5. Чем отличаются проверяемые (checked) исключения от непроверяемых (unchecked)?

6. Что означают wildcard-символы <? extends T> и <? super T> в Generics?

7. В чем ключевое отличие между ArrayList, LinkedList и HashSet?

8. Для чего используются методы filter, map и reduce в потоках данных?

9. Как обеспечить потокобезопасность при доступе к общей переменной из нескольких потоков?

10. Какой паттерн гарантирует наличие только одного экземпляра класса и как его реализовать потокобезопасно?

Примерное описание задания и критерии оценивания к проекту

Тема проекта Защищённое клиент-серверное приложение с VPN-шифрованием

Цель: Создать полноценное безопасное веб-приложение с БД, многопоточностью, HTTPS и паттернами проектирования.

Требования к реализации:

1. HTTP и REST API: Реализовать endpoints (GET, POST, PUT, DELETE) для управления подключениями. Использовать Spring Boot или HttpServlet. Поддержка HTTPS (порт 443).
2. Работа с БД: Подключить базу данных (PostgreSQL/MySQL). Сохранение логов подключений, статистики трафика, пользователей. Использовать JDBC или JPA/Hibernate.
3. Многопоточность:
— Обработка множественных клиентских подключений через ExecutorService.
— Реализовать лимит запросов для защиты от DoS-атак.
— Обеспечить потокобезопасность общих ресурсов (мьютексы, ConcurrentHashMap).
4. Безопасность:
— Настройка IPsec-туннеля (например, strongSwan) для VPN-шифрования.
— Реализация TLS/SSL для HTTPS.
— Защита от MITM (сертификаты), DNS-спуфинг (валидация DNS).
5. Паттерны проектирования:
— Singleton для подключения к БД и менеджера соединений.
— Factory для создания типов клиентов.
— Observer для мониторинга трафика в реальном времени.
— Repository для доступа к данным.
6. Тестирование и верификация:
— Стресс-тест (10 000 запросов).
— Захват трафика в Wireshark/tcpdump.
— Подтверждение шифрования (порт 443, TLS-handshake).
7. Финальная сборка: Проект запускается через docker-compose (приложение + БД + VPN-шлюз).

Критерии оценивания проекта:

№	Критерий	Что проверяется
1	Корректность архитектуры и схемы сети	Соответствие выбранных портов/протоколов заявленным требованиям, правильность расположения VPN-туннеля, наличие всех необходимых компонентов (сервер, клиент, DNS, шлюз).
2	Качество программной реализации	Структурированность кода (разделение по пакетам), использование Java-библиотек (java.net, javax.net.ssl, библиотеки для IPsec), наличие комментариев, обработка исключений, покрытие критических участков тестами (JUnit).
3	Безопасность и устойчивость	Реализация HTTPS/TLS, шифрование трафика, защита от DoS (лимит запросов), описание мер против MITM и DNS-спуфинг, доказательство (лог-файлы, захваченный трафик).

Электронный документ

№	Критерий	Что проверяется
4	Тестирование и верификация	Проведённые нагрузочные и функциональные тесты, отчёт о результатах (время отклика, процент ошибок), наличие скриншотов/логов из Wireshark/tcpdump.
5	Документация	Полнота отчёта (введение, схема, код, результаты, выводы, рекомендации), качество оформления.

Задания для промежуточной аттестации по дисциплине (модулю)

№ п/п	Задание	Ответ (все варианты)	Компетенция
1	Укажите ключевое слово, используемое для наследования класса в Java.	extends	ОПК-6
2	Назовите механизм, позволяющий реализовать множественное наследование типов.	интерфейсы / interfaces	ОПК-6
3	Укажите конструкцию языка для автоматического закрытия ресурсов.	try-with-resources	ОПК-6
4	Как называется базовый класс для всех ошибок и исключений в Java?	Throwable	ОПК-6
5	Укажите инструмент сборки и управления зависимостями, изучаемый в курсе.	Maven	УК-1
6	Назовите принцип SOLID, запрещающий изменять поведение базового класса в наследниках.	Подстановка Лисков / Liskov Substitution	УК-1
7	Укажите интерфейс коллекции для хранения упорядоченной последовательности элементов.	List	ОПК-4
8	Назовите интерфейс коллекции для хранения пар «ключ-значение».	Map	ОПК-4
9	Укажите функциональный интерфейс, возвращающий булево значение.	Predicate	ОПК-4
10	Назовите метод Stream API для преобразования элементов потока.	map	ОПК-4
11	Укажите аннотацию, обозначающую метод модульного теста в JUnit.	@Test	УК-1
12	Назовите ключевое слово для обеспечения потокобезопасности через монитор.	synchronized	ОПК-6
13	Укажите класс для атомарных операций над целочисленным значением.	AtomicInteger	ПК-3

№ п/п	Задание	Ответ (все варианты)	Компетенция
14	Назовите HTTP-метод, используемый для создания нового ресурса.	POST	ОПК-6
15	Укажите интерфейс JDBC для представления соединения с базой данных.	Connection	ОПК-6
16	Назовите порождающий паттерн проектирования, гарантирующий единственный экземпляр класса.	Singleton / Одиночка	ПК-3
17	Укажите структурный паттерн для совместной работы несовместимых интерфейсов.	Adapter / Адаптер	ПК-3
18	Назовите интерфейс пула потоков для управления выполнением задач.	ExecutorService	ПК-3
19	Укажите асимптотическую сложность операции получения элемента в HashMap.	O(1)	ОПК-1
20	Назовите примитивный тип данных для представления логических значений.	boolean	ОПК-1