
УТВЕРЖДЕНА

Решением Ученого совета
АНО ВО «Центральный университет»
«07» марта 2024 г.
Протокол №1

**Рабочая программа дисциплины (модуля)
«Многопоточное программирование»**

Направление подготовки: 02.04.01 Математика и компьютерные науки

Направленность (профиль) подготовки: Backend-разработка

Квалификация (степень) выпускника: магистр

Форма обучения: очная

Срок освоения программы: 2 года

Год набора: 2024

**Москва
2024**

Содержание

1. Краткая характеристика дисциплины (модуля)	3
2. Перечень планируемых результатов обучения.....	4
3. Тематический план.....	6
4. Содержание дисциплины (модуля).....	6
5. Учебно-методическое обеспечение	7
6. Материально-техническое обеспечение	7
7. Методические и оценочные материалы	9

1. Краткая характеристика дисциплины (модуля)

Рабочая программа дисциплины (модуля) «Многопоточное программирование» составлена в соответствии с федеральным государственным образовательным стандартом высшего образования – магистратура по специальности 02.04.01 Математика и компьютерные науки, профиль Backend-разработка, утвержденный приказом Министерства науки и высшего образования Российской Федерации № 810 от 23.08.2017 года.

Многопоточное программирование является ключевым навыком в современной разработке, поскольку позволяет эффективно использовать многоядерные процессоры и ресурсы системы. Это способствует созданию более быстрых, масштабируемых и отзывчивых приложений, что особенно важно для серверных систем, игр, обработки больших данных и пользовательских интерфейсов. Понимание принципов синхронизации и управления потоками помогает предотвращать ошибки, такие как гонки данных и взаимные блокировки, обеспечивая надежность и стабильность программ.

Место дисциплины (модуля) в структуре образовательной программы

Настоящая дисциплина (модуль) включена в учебный план по программе подготовки магистратуры по направлению 02.04.01 Математика и компьютерные науки, профиль Backend-разработка и входит в обязательную часть Блока 1.

Дисциплина (модуль) изучается на 2 курсе в 3 семестре.

Цель изучения дисциплины (модуля): формирование у студентов знаний о методах и средствах разработки программного обеспечения, способного эффективно выполнять несколько потоков выполнения для повышения производительности и отзывчивости приложений.

Задачи изучения дисциплины (модуля):

- формирование знаний о распространенных проблемах гонки данных;
- формирование знаний устройства примитивов синхронизации;
- формирование умения применять механизмы синхронизации;
- формирование умения реализовывать структуры данных и алгоритмы, используемые в многопоточном программировании;
- формирование умения реализовывать надежные многопоточные программы.

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)
ОПК-3.	Способен самостоятельно создавать прикладные программные средства на основе современных информационных технологий и сетевых ресурсов, в том числе отечественного производства	ОПК-3.1.	Знает основные принципы программирования, архитектуры программного обеспечения и современные языки программирования, а также особенности отечественных информационных технологий и сетевых ресурсов
		ОПК-3.2.	Умеет разрабатывать прикладные программные средства, используя современные инструменты и технологии, а также интегрировать их с сетевыми ресурсами для решения конкретных задач
		ОПК-3.3.	Имеет практический опыт разработки программных средств, используемых при построении математических моделей в естественных науках
ПК-2.	Способен математически корректно ставить естественнонаучные и прикладные задачи	ПК-2.1.	Знает основные методы и подходы к математическому моделированию, а также теоретические основы естественных и прикладных наук, необходимые для корректной формулировки задач
		ПК-2.2.	Умеет анализировать практические ситуации и формулировать на их основе математические модели, включая выбор адекватных методов решения и формулировку условий задачи
		ПК-2.3.	Имеет практический опыт в разработке и решении математических задач в рамках проектов или научных исследований, где были успешно поставлены и решены естественнонаучные и прикладные задачи

ПК-3.	Способен решать задачи профессиональной деятельности, формулировать результат, увидеть следствия полученного результата	ПК-3.1.	Знает основные принципы и методы решения задач профессиональной деятельности, а также способы формулирования и представления результатов, включая анализ последствий и их значимость в контексте проекта
		ПК-3.2.	Умеет применять математические и компьютерные методы для решения конкретных задач, формулировать четкие и обоснованные результаты, а также анализировать их последствия для дальнейших действий и решений
		ПК-3.3.	Имеет практический опыт в решении профессиональных задач, включая участие в проектах, где были получены результаты и проанализированы их следствия, что способствовало принятию обоснованных решений

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)	Консультации					
1	Основы многопоточности	3	3	5		32	Домашнее задание Подготовка к семинару
2	Синхронизация и управление потоками	3	3	5		32	Домашнее задание Подготовка к семинару
3	Продвинутое структуры данных и алгоритмы	4	4	6		34	Домашнее задание Подготовка к семинару
4	Асинхронные и реактивные системы	3	3	6		34	Домашнее задание Подготовка к семинару
5	Оптимизация и практика	3	3	6		32	Домашнее задание Проект
	<i>Экзамен</i>				4		
Итого:		16	16	28	4	164	
Объем дисциплины (модуля) (в ак. ч.)		228					
Объем дисциплины (модуля) (в зач. ед.)		6					

4. Содержание дисциплины (модуля)

№п/п	Наименование раздела дисциплины (модуля)	Содержание дисциплины (модуля) по темам
1	Основы многопоточности	Введение, закон Мура, race condition, модель памяти. Класс Thread, Runnable, жизненный цикл потока. Проблемы видимости (volatile), атомарность (Atomic-классы).
2	Синхронизация и управление потоками	synchronized, мониторы, deadlock. Пулы потоков (ExecutorService), Future. CompletableFuture. Интерфейс Lock, ReadWriteLock, Condition
3	Продвинутое структуры данных и алгоритмы	ConcurrentHashMap, BlockingQueue, неблокирующие структуры. Синхронизаторы (CountDownLatch, CyclicBarrier, Semaphore). ForkJoinPool, параллельные стримы
4	Асинхронные и реактивные системы	Реактивное программирование (Reactor/RxJava), очереди событий. Виртуальные потоки. Тестирование многопоточного кода (нагрузочные тесты, детектирование гонок)
5	Оптимизация и практика	Мониторинг (VisualVM, JFR), профилирование, оптимизация блокировок

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. Зыков, С. В. Программирование : учебник и практикум для вузов / С. В. Зыков. — 2-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2025. — 285 с. — (Высшее образование). — ISBN 978-5-534-16031-4. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/560815>.

2. Компьютерные сети : учебник и практикум для вузов / под научной редакцией А. М. Нечаева, А. Е. Трубина, А. Ю. Анисимова. — Москва : Издательство Юрайт, 2025. — 515 с. — (Высшее образование). — ISBN 978-5-534-21452-9. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/572239>.

Дополнительная литература:

1. Гниденко, И. Г. Технологии и методы программирования : учебник для вузов / И. Г. Гниденко, Ф. Ф. Павлов, Д. Ю. Федоров. — 2-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2025. — 241 с. — (Высшее образование). — ISBN 978-5-534-18130-2. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/581329>.

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;

— 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 к 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) «Многопоточное программирование» в рамках текущего контроля успеваемости используются такие виды учебной работы, как лекции, семинары, консультации, проект, домашние задания, 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 удастся, следует сформулировать вопрос и задать его преподавателю на консультации или во время семинарского (практического) занятия.

Участие в семинаре (аудиторная работа) – активная работа студента на семинаре, его ответы на вопросы преподавателя и участие в дискуссии.

Для успешного участия в семинаре студентам рекомендуется заранее ознакомиться с темой обсуждения, прочитать необходимые материалы и подготовить вопросы. Важно активно слушать и вовлекаться в дискуссию, высказывая свои мнения и аргументируя их. При ответах на вопросы преподавателя стоит быть уверенным, четким и логичным, опираясь на изученный материал. Также полезно поддерживать диалог с однокурсниками, чтобы обогатить обсуждение и расширить свои знания.

Консультации – структурированные встречи, на которых преподаватели предоставляют индивидуальную или групповую помощь в освоении учебного материала, обсуждении вопросов и решении проблем, возникающих в процессе обучения.

Консультации могут включать разъяснение сложных тем, подготовку к экзаменам и помощь в выполнении проектных работ, что способствует более глубокому пониманию предмета и улучшению академической успеваемости.

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.

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

Для успешной подготовки к проекту: четко определите цели и задачи проекта, распределите роли и обязанности между участниками, а также установите сроки выполнения каждой части работы. Регулярно проводите встречи для обсуждения прогресса и решения возникающих вопросов.

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) «Многопоточное программирование»

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 *экзамена*, 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:

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
6	Хорошо	

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

Дисциплина (модуль) «Многопоточное программирование» оценивается следующим образом:

Активность	Вес	Описание
Домашние задания	50%	Набор заданий по темам. За каждое из заданий можно набрать 10 баллов
Проект	20%	Исследовательская работа по курсу и презентация результатов
Экзамен	30%	Письменная или устная работа над заданием, направленным на проверку полученных знаний и навыков по дисциплине (модулю)

Формула расчёта итоговой оценки по дисциплине (модулю) «Многопоточное программирование»: $\langle 0,5 \times \text{среднее за домашние задания} + 0,2 \times \text{проект} + 0,3 \times \text{экзамен} \rangle$.

Текущий контроль успеваемости обучающихся по дисциплине (модулю)

Примерные темы семинарских занятий

Основы многопоточности

- 1. Введение в многопоточность:** Преимущества и недостатки многопоточного программирования.
- 2. Закон Мура и его влияние на производительность:** Как закон Мура влияет на многопоточность и параллелизм.
- 3. Race Condition:** Определение, примеры и способы предотвращения гонки данных.
- 4. Модель памяти в Java:** Как работает модель памяти и её влияние на многопоточность.

5. **Классы Thread и Runnable:** Жизненный цикл потока и их использование в многопоточном программировании.

Синхронизация и управление потоками

1. **Ключевое слово synchronized и мониторы:** Как работают синхронизация и блокировка потоков.

2. **Deadlock:** Определение, примеры и методы предотвращения взаимной блокировки.

3. **Пулы потоков и ExecutorService:** Как использовать ExecutorService для управления потоками.

4. **Future и CompletableFuture:** Асинхронное программирование с использованием Future и CompletableFuture.

5. **Интерфейсы Lock и ReadWriteLock:** Различия, применение и примеры использования.

Продвинутые структуры данных и алгоритмы

1. **ConcurrentHashMap:** Преимущества и применение в многопоточных приложениях.

2. **BlockingQueue:** Использование BlockingQueue для синхронизации потоков и обработки данных.

3. **Неблокирующие структуры данных:** Примеры и преимущества неблокирующих структур.

4. **Синхронизаторы (CountDownLatch, CyclicBarrier, Semaphore):** Как и когда использовать эти синхронизаторы.

5. **ForkJoinPool и параллельные стримы:** Применение ForkJoinPool и использование параллельных стримов для обработки данных.

Асинхронные и реактивные системы

1. **Реактивное программирование:** Основы реактивного программирования и его применение (Reactor/RxJava).

2. **Очереди событий:** Как работают очереди событий и их роль в асинхронных системах.

3. **Виртуальные потоки:** Преимущества и использование виртуальных потоков в Java.

4. **Тестирование многопоточного кода:** Методы нагрузочного тестирования и детектирования гонок.

5. **Сравнение асинхронного и реактивного программирования:** Преимущества и недостатки каждого подхода.

Примерные домашние задания

Домашнее задание: Основы многопоточности

1. Объясните закон Мура и его влияние на развитие многопоточных приложений.

2. Что такое race condition? Приведите пример ситуации, когда она может возникнуть.

3. Опишите жизненный цикл потока в Java с использованием классов Thread и Runnable.

4. Что такое проблема видимости в многопоточности? Как ключевое слово volatile помогает её решить?

5. Чем отличаются атомарные операции от неатомарных? Приведите пример использования Atomic-классов в Java.

Домашнее задание: Синхронизация и управление потоками

1. Объясните принцип работы ключевого слова `synchronized` и роль мониторов в Java.
2. Что такое `deadlock`? Приведите пример кода, который может привести к взаимной блокировке.
3. Опишите назначение и использование пула потоков `ExecutorService`. Как работает интерфейс `Future`?
4. Что такое `CompletableFuture` и как он расширяет возможности работы с асинхронными задачами?
5. В чем разница между интерфейсами `Lock` и `ReadWriteLock`? Как используется `Condition`?

Домашнее задание: Продвинутые структуры данных и алгоритмы

1. Опишите структуру данных `ConcurrentHashMap` и её преимущества по сравнению с обычным `HashMap` в многопоточном окружении.
2. Что такое `BlockingQueue`? Приведите пример её использования для организации взаимодействия потоков.
3. Объясните назначение синхронизаторов `CountDownLatch`, `CyclicBarrier` и `Semaphore`. В каких ситуациях они применяются?
4. Что такое `ForkJoinPool`? Как он помогает в реализации параллельных вычислений?
5. Как работают параллельные стримы в Java? Приведите пример их использования для обработки коллекции.

Примерное задание для проекта

Проект по теме: Мониторинг, профилирование и оптимизация блокировок в многопоточном приложении

Цель проекта:

Научиться использовать инструменты мониторинга (`VisualVM`, `Java Flight Recorder`) для анализа производительности многопоточного приложения, выявлять узкие места, связанные с блокировками, и оптимизировать код для повышения эффективности.

Задание:

1. **Выбрать или разработать многопоточное Java-приложение**, в котором присутствуют блокировки (например, с использованием `synchronized`, `Lock`, `BlockingQueue` и т.п.). Можно взять готовый пример с гонками или с избыточной синхронизацией.
2. **Собрать исходные показатели производительности** приложения:
 - Время выполнения ключевых операций.
 - Использование CPU.
 - Количество и длительность блокировок.
3. **Провести мониторинг и профилирование** с помощью:
 - `VisualVM` (сбор данных о потоках, блокировках, памяти).
 - `Java Flight Recorder (JFR)` — анализ событий и блокировок.
4. **Проанализировать полученные данные**, выявить узкие места, связанные с блокировками и синхронизацией.
5. **Внести изменения в код для оптимизации блокировок**, например:
 - Снизить область синхронизации.
 - Заменить `synchronized` на более эффективные примитивы (`Lock`, `ReadWriteLock`).

- Использовать неблокирующие структуры данных.
- 6. **Повторно провести мониторинг и профилирование** после оптимизации.
- 7. **Сравнить результаты до и после оптимизации**, сделать выводы о достигнутом улучшении.
- 8. **Подготовить отчет**, включающий:
 - Описание приложения и его многопоточной части.
 - Методики мониторинга и инструменты.
 - Анализ исходных данных.
 - Описание внесенных изменений.
 - Сравнительный анализ результатов.
 - Выводы и рекомендации.

Этапы выполнения:

Этап	Срок	Описание
1. Выбор/создание приложения	1 неделя	Подготовка исходного многопоточного кода
2. Сбор исходных данных	1 неделя	Мониторинг и профилирование до оптимизации
3. Анализ и оптимизация	1-2 недели	Поиск узких мест и внесение изменений
4. Повторный мониторинг	1 неделя	Сбор данных после оптимизации
5. Подготовка отчета	1 неделя	Оформление результатов и выводов

Критерии оценивания:

Критерий	Максимальный балл	Описание
Корректность и полнота мониторинга	25	Использование VisualVM и JFR, полнота собранных данных
Анализ и выявление узких мест	20	Глубина и качество анализа проблем с блокировками
Качество и эффективность оптимизации	25	Внесённые изменения, их влияние на производительность
Сравнительный анализ результатов	15	Чёткое сравнение показателей до и после оптимизации
Оформление и полнота отчёта	15	Структура, ясность изложения, наличие всех разделов

Форма сдачи проекта:

- Исходный код проекта (до и после оптимизации).
- Файлы с профилями/снимками из VisualVM и JFR.
- Отчет в формате PDF (10-15 страниц), включающий все пункты из задания.
- Краткая презентация (5-7 слайдов) с основными результатами и выводами.

Задания для промежуточной аттестации по дисциплине (модулю)

№ п/п	Задание	Ответ	Компетенция
1.	Какой закон описывает увеличение количества транзисторов на чипах с течением времени?	Закон Мура	ОПК-3
2.	Какой ключевое слово используется для синхронизации блоков кода в Java?	Synchronized synchronized	ПК-2
3.	Состояние гонки, когда несколько потоков одновременно обращаются к общему ресурсу, что может привести к непредсказуемым результатам – это?	Race condition race condition	ПК-3
4.	Какой класс в Java используется для создания потоков?	Thread	ПК-2
5.	Какой класс используется для создания неблокирующих структур данных в Java?	ConcurrentHashMap	ОПК-3
6.	Как называется модель, которая описывает, как потоки взаимодействуют с памятью?	Модель памяти	ОПК-3
7.	Какой интерфейс позволяет управлять состоянием блокировки в многопоточном приложении?	Lock	ПК-3
8.	Какой класс в Java позволяет реализовать пул потоков?	ExecutorService	ПК-2
9.	Какой синхронизатор используется для ожидания завершения группы потоков?	CountDownLatch	ОПК-3
10.	Какой интерфейс используется для создания асинхронных задач в Java?	CompletableFuture	ПК-2
11.	Какой класс используется для создания очередей с блокировкой?	BlockingQueue	ОПК-3
12.	Какой метод используется для запуска потока в Java?	start()	ПК-2
13.	Какой класс используется для реализации параллельных стримов в Java?	ForkJoinPool	ОПК-3
14.	Какой инструмент используется для профилирования Java-приложений?	VisualVM	ПК-2
15.	Какой механизм позволяет избежать взаимных блокировок?	Deadlock prevention	ПК-3