
УТВЕРЖДЕНА

Решением Ученого совета
АНО ВО «Центральный университет»
«25» мая 2026 г.
Протокол № 2

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

Направление подготовки: 02.04.01 Математика и компьютерные науки

Направленность (профиль) подготовки: Программная инженерия

Квалификация (степень) выпускника: магистр

Форма обучения: очная

Срок освоения программы: 2 года

Год набора: 2026

**Москва
2026**

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	8
6. Материально-техническое обеспечение	8
7. Методические и оценочные материалы	10

1. Краткая характеристика дисциплины (модуля)

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

Изучение дисциплины (модуля) «Многопоточное программирование» позволяет понять принципы параллелизма, научиться создавать потокобезопасные и производительные приложения, а также овладеть навыками отладки и профилирования многопоточных систем.

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 Программная инженерия и входит в обязательную часть Блока 1.

Дисциплина (модуль) доступна к изучению на 2 курсе в 3 семестре.

Цель изучения дисциплины (модуля): формирование у студентов глубокого понимания принципов параллельного выполнения, навыков проектирования и реализации потокобезопасных, высокопроизводительных приложений, а также развитие компетенций в области отладки, тестирования и профилирования многопоточных систем.

Задачи изучения дисциплины (модуля):

- изучить модель многопоточности, жизненный цикл потоков и механизмы синхронизации;
- освоить принципы потокобезопасности, включая использование блокировок, атомарных операций и потокобезопасных структур данных;
- научиться управлять созданием, взаимодействием и завершением потоков в различных средах выполнения;
- освоить основы реактивного программирования и асинхронных паттернов;
- развить навыки диагностики и устранения типовых ошибок (гонки, взаимная блокировка, голодание) с помощью профилировщиков и отладчиков.

В результате освоения дисциплины (модуля) обучающийся должен:

знать:

- понятие многопоточности и жизненный цикл потоков;
- проблемы и принципы соблюдения потокобезопасности;
- способы управления потоками;
- принципы реактивного программирования.

уметь:

- создавать потоки и управлять их состоянием;
- делать фрагменты кода и приложения потокобезопасными;
- управлять множествами потоков;
- настраивать мониторинг и профилировать многопоточный код.

владеть:

- навыком реализации надежных и безопасных многопоточных приложений;
- навыком тестирования и отлаживания многопоточного кода.

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)			
1	Основы многопоточности. Блокировки и атомарные переменные.	6	6		25	Домашние задания
2	Модели памяти. Поддержка многопоточности на уровне процессора.	6	6		26	Домашние задания
3	Различные механизмы синхронизации в Java.	6	6		26	Домашние задания
4	Lock-free структуры данных и алгоритмы.	6	6		26	Домашние задания
5	Тестирование, отладка и продвинутые темы, связанные с многопоточностью.	6	6		25	
	Экзамен			2		
Итого:		30	30	2	128	
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	Основы многопоточности. Блокировки и атомарные переменные.	Параллельность и конкурентность. Поток и процессы. Взаимное исключение. Data race, race condition. Гарантия liveness, deadlocks. Атомарные переменные.
2	Модели памяти. Поддержка многопоточности на уровне процессора.	Store-load reorderings. Устройство памяти в x86, arm. Протокол когерентности кэшей. Модели памяти в языках программирования. Отношение happens-before. Последовательная согласованность. Модель памяти Java. Слабые модели памяти.
3	Различные механизмы синхронизации в Java.	Пулы потоков, условные переменные. Разные примитивы синхронизации в Java. Java virtual threads, фиберы. Задачи, колбэки, фьючи.
4	Lock-free структуры данных и алгоритмы.	Lock-free гарантии, реализации на Java. Проблема менеджмента памяти, hazard pointers. Lock-free. Проблема менеджмента памяти, Hazard Pointers. Конкурентные структуры данных.
5	Тестирование, отладка и продвинутое темы, связанные с	Тестирование, отладка, профилирование многопоточного кода. JIT компиляция.

	многопоточностью.	Scalable scheduler на примере ForkJoinPool. Structured concurrency.
--	-------------------	--

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. Уильямс, Э. С++. Практика многопоточного программирования : практическое руководство / Э. Уильямс. - Санкт-Петербург : Питер, 2020. - 640 с. - (Серия «Для профессионалов»). - ISBN 978-5-4461-0831-2. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1733519>.

2. Цукалос, М. Golang для профи: работа с сетью, многопоточность, структуры данных и машинное обучение с Go : практическое руководство / М. Цукалос. - Санкт-Петербург : Питер, 2021. - 720 с. - (Серия «Для профессионалов»). - ISBN 978-5-4461-1617-1. - Текст : электронный. - URL: <https://znanium.ru/catalog/product/1733707>.

Дополнительная литература:

1. Уильямс, Э. Параллельное программирование на С++ в действии. Практика разработки многопоточных программ : практическое руководство / Э. Уильямс ; пер. с англ. А. А. Слинкина. - 2-е изд. - Москва : ДМК Пресс, 2023. - 673 с. - ISBN 978-5-89818-319-6. - Текст : электронный. - URL: <https://znanium.ru/catalog/product/2102606>.

2. Клири, С. Конкурентность в C#. Асинхронное, параллельное и многопоточное программирование : практическое руководство / С. Клири. - 2-е межд. изд. - Санкт-Петербург : Питер, 2020. - 272 с. - (Серия «Для профессионалов»). - ISBN 978-5-4461-1572-3. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1756127>.

3. Maurice Herlihy, Nir Shavit. The Art of Multiprocessor Programming. 2008 - ISBN 978-0-12-370591-4.

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) «Многопоточное программирование» в рамках текущего контроля успеваемости используются такие виды учебной работы, как лекции, семинары, 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.

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.

Электронный документ

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, 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
4	Удовлетворительно	

Десятибалльная оценка	Пятибалльная оценка	Общая характеристика результата обучения по дисциплине (модулю)
		ответов он допускает ошибки, касающиеся сути вопросов. Студент способен решать только самые простые задачи и владеет лишь минимальным набором методов исследования.
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	Количество	Описание
Домашние задания	50%	6	Набор задач по темам недели
Экзамен	50%	1	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»: $\langle 0,5 \times \text{среднее за домашние задания} + 0,5 \times \text{экзамен} \rangle$.

Текущий контроль успеваемости обучающихся по дисциплине (модулю)

Примерные домашние задания

Домашнее задание 1

1. Создайте несколько потоков, которые одновременно увеличивают общую счётчиковую переменную. Продемонстрируйте возникновение состояния гонки (data race).
2. Исправьте код с использованием synchronized и ReentrantLock, сравните поведение.
3. Реализуйте ту же операцию с помощью AtomicInteger. Объясните, в чём разница по производительности и безопасности.
4. Напишите пример взаимной блокировки (deadlock) из двух потоков и предложите способ её предотвращения.
5. Оформите отчёт с анализом состояний потоков и использованных примитивов синхронизации.

Домашнее задание 2

1. Напишите программу, демонстрирующую переупорядочивание операций чтения/записи (reordering) без синхронизации.
2. Исправьте поведение с помощью volatile и объясните, как работает отношение happens-before в Java.
3. Реализуйте потокобезопасный флаг завершения работы, устойчивый к оптимизациям JIT.
4. Сравните поведение программы на x86 и в теоретической слабой модели памяти (например, ARM).
5. Опишите, как модель памяти Java обеспечивает согласованность кэшей через протоколы когерентности.

Домашнее задание 3

1. Реализуйте lock-free односвязный список на Java с использованием

- AtomicReference и compareAndSet.
2. Напишите тесты для проверки корректности вставки и удаления элементов из нескольких потоков.
 3. Создайте пул потоков с помощью ExecutorService и выполните на нём набор асинхронных задач, возвращающих Future.
 4. Перепишите ту же логику с использованием виртуальных потоков (virtual threads) в Java 21+. Сравните потребление памяти и производительность.
 5. Проанализируйте выполнение с помощью профилировщика (например, VisualVM или async-profiler), выявите узкие места и предложите оптимизации.

Задания для промежуточной аттестации по дисциплине (модулю)

№ п/п	Задание	Ответ	Компетенция
1.	Укажите, какая проблема возникает при одновременном доступе нескольких потоков к одной переменной без синхронизации	data race	ПК-2
2.	Напишите, как называется состояние, при котором два потока ждут блокировок друг у друга	deadlock	ПК-3
3.	Укажите, какой интерфейс в Java используется для создания пула потоков	ExecutorService	ОПК-3
4.	Напишите, какая аннотация или ключевое слово в Java гарантирует видимость изменений между потоками	volatile	ПК-2
5.	Укажите, какая операция используется в lock-free алгоритмах для атомарного обновления ссылки	compareAndSet (CAS)	ПК-3
6.	Напишите, как называется модель, при которой все потоки видят одинаковый порядок операций	последовательная согласованность	ОПК-3
7.	Укажите, какой метод останавливает выполнение потока до завершения другого	join()	ПК-2
8.	Напишите, как называется пул, оптимизированный для рекурсивных задач в Java	ForkJoinPool	ПК-3
9.	Укажите, какая структура данных в Java является потокобезопасной и не использует блокировки	ConcurrentLinkedQueue	ОПК-3
10.	Напишите, как называется подход, при котором потоки создаются легковесными и управляются средой выполнения	виртуальные потоки (virtual threads)	ПК-2
11.	Укажите, какой принцип гарантирует, что одно событие происходит раньше другого в многопоточной среде	happens-before	ПК-3
12.	Напишите, как называется ошибка, при которой потоки бесконечно ждут условия из-за отсутствия сигнала	голодание (starvation)	ОПК-3
13.	Укажите, какой класс в Java предоставляет атомарные операции с целыми числами	AtomicInteger	ПК-2

14.	Напишите, как называется механизм, предотвращающий переупорядочивание компилятором и процессором	memory barrier / fence	ПК-3
15.	Укажите, какая проблема возникает при удалении узла в lock-free структуре из-за повторного использования памяти	ABA-проблема	ОПК-3
16.	Напишите, как называется подход к написанию многопоточного кода, при котором область видимости потоков структурирована	structured concurrency	ПК-2
17.	Укажите, какой инструмент позволяет визуализировать загрузку потоков и задержки выполнения	профилировщик (например, VisualVM)	ПК-3
18.	Напишите, как называется паттерн, при котором задачи разделяются и объединяются в пуле потоков	fork/join	ОПК-3
19.	Укажите, какой тип блокировки не использует системные вызовы ядра при ожидании	спин-блокировка (spinlock)	ПК-2
20.	Напишите, как называется указатель, защищающий узел от преждевременного освобождения в lock-free коде	hazard pointer	ПК-3