

**РП СФОРМИРОВАНА,
СОГЛАСОВАНА
И УТВЕРЖДЕНА В ЭИОС**

УТВЕРЖДАЮ /А.Н. Дедов/
(Ф.И.О. декана (директора института))

Б.1.2.15 Параллельные вычисления

12.03.04 Биотехнические системы и технологии

Бакалавр

Биомедицинские интеллектуальные системы и комплексы

Распределение учебного времени

(ГОД)

Программа составлена в соответствии с требованиями ФГОС ВО направления подготовки (специальности) 12.03.04 Биотехнические системы и технологии

Программу составили:

доцент	РТиМБС	СОГЛАСОВАНО	К.О. Иванов
(должность)	(кафедра)		(И.О. Фамилия)

РАССМОТРЕНА и ОДОБРЕНА на заседании кафедры, за которой закреплена дисциплина
Кафедра радиотехнических и медико-биологических систем

(наименование кафедры)		
20.01.2025	протокол №	6
(дата)		
Заведующий кафедрой	СОГЛАСОВАНО	Р.Г. Хафизов
		(И.О. Фамилия)

Рабочая программа СОГЛАСОВАНА с факультетом (институтом), выпускающей(ими) кафедрой(ами).

СООТВЕТСТВУЕТ действующей ОП.

Заведующий кафедрой	СОГЛАСОВАНО	Р.Г. Хафизов
		(И.О. Фамилия)

Председатель методической комиссии факультета (института), в который входит выпускающая кафедра

СОГЛАСОВАНО	А.Н. Дедов
	(И.О. Фамилия)

Эксперт(ы): Мухин Игорь Павлович, зав. научной лаборатории ООО "НПФ Мета-хром"

Рабочая программа проверена и зарегистрирована в УМЦ 12.03.2025 г.

Специалист учебно-методического центра СОГЛАСОВАНО /Т.А. Смирнова/

Раздел 1. ЦЕЛЬ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Целью освоения дисциплины является достижение планируемых результатов обучения, соответствующих установленным в ОПОП индикаторам достижения компетенций:

Код и наименование компетенции	Код и наименование индикатора достижения компетенции	Результаты обучения
1. ПК-2 Способность к математическому моделированию элементов и процессов биотехнических систем, их исследованию на базе профессиональных пакетов автоматизированного проектирования и самостоятельно разработанных программных продуктов	ПК-2.1 Разрабатывает алгоритмы и реализует математические и компьютерные модели элементы и процессы биотехнических систем с использованием объектно-ориентированных технологий	знания: Знание высшей математики, включая дифференциальные уравнения, линейную алгебру, математическую статистику и численные умения: Умение обработки и анализа данных, особенно в контексте биотехнологических приложений. навыки: Навыки программирования на языках, таких как Python, Java, C++ или других, используемых в биотехнических приложениях.
	ПК-2.2 Разрабатывает, реализует и применяет в профессиональной деятельности технологии искусственного интеллекта и различные численные методы, в том числе реализованные в готовых библиотеках при решении задач проектирования биотехнических систем	знания: Знание математических моделей, используемых в биотехнических системах. умения: Умение решать задачи оптимизации с учетом ограничений и специфических требований. навыки: Навыки применения различных численных методов, таких как метод конечных элементов, метод конечных разностей, метод Монте-Карло и другие, к решению задач в биотехнической области.
	ПК-2.3 Разрабатывает библиотеки и подпрограммы (макросы) для решения различных задач проектирования и конструирования, исследования и контроля биотехнических систем	знания: Знание работы с базами данных, если это необходимо для проекта. умения: Умение оптимизировать код для повышения производительности. навыки: Навыки разработки на высокоуровневых языках программирования, таких как Python, C++, Java, MATLAB и других, в зависимости от требований проекта.

2. ПК-4 Способность к созданию интегрированных биотехнических систем и медицинских систем и комплексов для решения сложных задач диагностики, лечения, мониторинга здоровья человека	ПК-4.1 Разрабатывает структуру интегрированной биотехнической системы комплексной диагностики, лечения, мониторинга и реабилитации здоровья человека на основе анализа информационных процессов, протекающих в биотехнической системе.	знания: Знание основных принципов работы медицинских устройств и систем. умения: Умение разрабатывать и оптимизировать технологические процессы производства. навыки: Навыки обработки и анализа данных, которые могут быть получены из биотехнических экспериментов.
	ПК-4.2 Выполняет сборку, юстировку и контроль медицинских изделий и биотехнических систем, а также наладку оборудования и поверку средств измерений	знания: Знание технологий изготовления, сборки, юстировки и контроля медицинских изделий и биотехнических систем умения: Умение разрабатывать и внедрять процессы юстировки для обеспечения точности измерений. навыки: Навыки эффективной отладки и исправления ошибок.
	ПК-4.3 Выполняет настройку программных средств, используемых для производства биотехнических систем медицинского назначения	знания: Знание основных стандартов документирования кода и процессов разработки программного обеспечения для решения медико-биологических задач умения: Умение использовать системы автоматического документирования программ в области биомедицинской инженерии навыки: Навыки документирования кода и создания пользовательской документации для библиотек и подпрограмм.

Раздел 2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОПОП

Дисциплина относится к элективным дисциплинам (модулям) ОПОП.

Дисциплина является элективной

Для продолжения формирования заявленных компетенций необходимы знания предшествующих дисциплин: Цифровые устройства и микропроцессоры (ПК-2), Цифровая обработка сигналов в биомедицинских системах (ПК-2); практик: Производственная практика. Технологическая (проектно-технологическая) практика (распределенная) (ПК-4) Изучаемая дисциплина является основой для продолжения формирования указанных компетенций в следующих дисциплинах: Интеллектуальные диагностические методы исследований в медицине (ПК-2); практиках: Преддипломная практика (ПК-2), Преддипломная практика (ПК-4); государственной итоговой аттестации в форме: Подготовка к процедуре защиты и защита выпускной квалификационной работы (ПК-2), Подготовка к процедуре защиты и защита выпускной квалификационной работы (ПК-4)

Раздел 3. ОПИСАНИЕ ОБРАЗОВАТЕЛЬНЫХ ТЕХНОЛОГИЙ

Для формирования заявленных компетенций используются методологические технологии, реализующие деятельностный, личностно-ориентированный, практико-ориентированный подходы.

Основными стратегическими технологиями являются: лекционные занятия, практические занятия, процедуры самообучения

На достижение конкретных целей обучения направлены применяемые тактические технологии: задания, классическая лекция, мини-проекты, проблемная лекция

Раздел 4. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

7 семестр

Виды и темы занятий	Количество часов	Формируемые компетенции
Параллельные вычисления	108	ПК-2, ПК-4
Лекция. Введение в параллельные вычисления: Понятие параллелизма и его применение в современных вычислениях. Типы параллельных архитектур и их особенности.	2	
Лекция. Основы многозадачности на C++: Потоки выполнения (threads) и их создание. Синхронизация потоков и обеспечение безопасности данных.	2	
Лекция. Стандартная библиотека шаблонов (STL) и параллелизм: Обзор STL и его компонентов, поддерживающих параллельные вычисления. Использование контейнеров и алгоритмов STL в параллельном режиме.	2	
Лекция. Параллельные паттерны программирования: Распределенная память и обмен данными между потоками. Параллельные циклы и разделение задач.	2	
Лекция. Параллельные вычислительные парадигмы: Модель MapReduce и ее применение в C++. Параллельные вычисления на основе задач (Task-based parallelism).	2	
Лекция. Оптимизация производительности: Профилирование и оптимизация параллельных программ. Инструменты для анализа и отладки параллельного кода.	2	
Лекция. Продвинутое тематическое программирование: Атомарные операции и их роль в обеспечении безопасности. Параллельное программирование с использованием GPU и фреймворков, таких как CUDA и OpenCL.	2	
Лекция. Практические примеры параллельных алгоритмов на C++: Реализация параллельных сортировок, поиска и других алгоритмов. Примеры использования параллельных вычислений в реальных проектах.	2	
Практическое занятие. Основы многозадачности: Создание и управление потоками выполнения в C++.	4	

Синхронизация потоков с использованием мьютексов и условных переменных.		
Практическое занятие. Параллельные алгоритмы с использованием STL: Работа с параллельными алгоритмами из библиотеки STL, такими как <code>std::for_each</code> , <code>std::transform</code> , <code>std::reduce</code> . Применение алгоритмов к различным структурам данных.	4	
Практическое занятие. Распределенная память и обмен данными: Использование структур данных для безопасного обмена данными между потоками. Работа с атомарными операциями для обеспечения атомарности операций.	4	
Практическое занятие. Параллельные циклы и разделение задач: Применение параллельных циклов для эффективной обработки данных. Разделение задач между потоками с использованием пулов задач.	4	
Практическое занятие. MapReduce в C++: Реализация простых MapReduce задач на C++ с использованием STL. Анализ и оптимизация производительности MapReduce-алгоритмов.	4	
Практическое занятие. Оптимизация производительности: Профилирование параллельных программ. Использование инструментов для выявления узких мест и оптимизации кода.	4	
Практическое занятие. Параллельное программирование с использованием GPU: Реализация простых вычислительных задач на GPU с помощью библиотеки CUDA. Сравнение производительности CPU и GPU реализаций.	4	
Практическое занятие. Практические примеры в реальных проектах: Работа с параллельными алгоритмами в реальных сценариях, таких как обработка данных, графики и т.д. Реализация эффективных параллельных решений для конкретных задач.	4	
Задания для самостоятельной работы, в том числе выполнение Самостоятельное изучение следующих разделов дисциплины: параллельное программирование на уровне задач; использование библиотек и фреймворков для управления задачами; создание и выполнение параллельных задач; использование библиотек, таких как Intel TBB (Threading Building Blocks) или OpenMP для параллельного программирования; разработка собственных компонентов для решения конкретных задач.	60	
Иная контактная работа:	0	

Раздел 5. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ

Изучение дисциплины рекомендуется начать с ознакомления с рабочей программой, ее структурой и содержанием разделов. Учебный материал структурирован, изучение дисциплины осуществляется в тематической последовательности. **Занятия лекционного типа** дают систематизированные знания по дисциплине, концентрируют внимание на наиболее сложных и важных вопросах. Во время лекционных занятий рекомендуется вести конспектирование учебного материала; обращать внимание на формулировки и категории, раскрывающие суть проблемы, явления или процесса; зафиксировать выводы и практические рекомендации. Подготовка к **занятиям семинарского типа** включает ознакомление с планом практического занятия; работу с конспектом лекций, выполнение домашнего задания, работу с учебной и учебно-методической литературой, научными изданиями и электронными образовательными ресурсами, рекомендованными рабочей программой дисциплины. Содержание **самостоятельной работы** определяется рабочей программой дисциплины, оценочными и методическими материалами, заданиями и указаниями преподавателя. Самостоятельная работа может осуществляться в аудиторной и внеаудиторной формах. Эффективным средством осуществления самостоятельной работы является электронная информационно-образовательная среда университета, которая обеспечивает доступ к образовательной программе, рабочей программе дисциплины, к электронным библиотечным системам, профессиональным базам данных и информационным справочным системам. Изучение дисциплины включает выполнение практических работ. Периодичность проведения, формы текущего контроля успеваемости, система оценивания хода освоения дисциплин представлены в рабочей программе. Формой промежуточной аттестации по дисциплине является зачет.

Раздел 6. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ И УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

6.1. Учебно-методическое обеспечение

№№ п/п	Список используемой литературы	Количество экземпляров печатных изданий, имеющих в библиотеке, или электронный адрес издания (ресурса) в сети Интернет
УЧЕБНЫЕ, УЧЕБНО-МЕТОДИЧЕСКИЕ И НАУЧНЫЕ ИЗДАНИЯ		
1.	Туганбаев, А. А. Системы линейных уравнений и линейное программирование [Электронный ресурс] : учебник для вузов / Туганбаев А. А. Санкт-Петербург: Лань, 2023. - 244 с. ISBN 978-5-507-48082-1.	https://e.lanbook.com/book/380684
2.	Новожилов, Олег Петрович. Архитектура ЭВМ и вычислительных систем [Текст : Электронный ресурс] : учебник для вузов / О. П. Новожилов. 2-е изд. Москва: Юрайт, 2024. - 505 с ISBN 978-5-534-20365-3.	https://urait.ru/bcode/558011
3.	Гвоздев, Владимир Ефимович. Управление программными проектами [Текст : Электронный ресурс] : учебное пособие для вузов / В. Е. Гвоздев [и др.] ; под редакцией Р. Ф. Маликова. Москва: Юрайт, 2024. - 167 с ISBN 978-5-534-14329-4.	https://urait.ru/bcode/543929
4.	Конова, Е. А. Алгоритмы и программы. Язык C++ [Электронный ресурс] : учебное пособие для вузов / Конова Е. А., Поллак Г. А.; Конова Е. А. 7-е изд., стер.	https://e.lanbook.com/book/297002

	Санкт-Петербург: Лань, 2023. - 384 с. ISBN 978-5-507-46070-0.	
5.	Иванов, Б. Н. Дискретная математика. Алгоритмы и программы. Расширенный курс [Электронный ресурс] : учебное пособие для вузов / Иванов Б. Н. Санкт-Петербург: Лань, 2023. - 668 с. ISBN 978-5-507-45685-7.	https://e.lanbook.com/book/356132
6.	Нагаева, И. А. Основы математического моделирования и численные методы [Электронный ресурс] / Нагаева И. А., Кузнецов И. А. Санкт-Петербург: Лань, 2024. - 204 с. ISBN 978-5-8114-9462-0.	https://e.lanbook.com/book/362324
ЭЛЕКТРОННЫЕ ОБРАЗОВАТЕЛЬНЫЕ РЕСУРСЫ		
1.	Научная электронная библиотека eLIBRARY.RU	http://elibrary.ru
2.	Научная электронная библиотека «Киберленинка»	http://cyberleninka.ru
ПРОФЕССИОНАЛЬНЫЕ БАЗЫ ДАННЫХ И ИНФОРМАЦИОННЫЕ СПРАВОЧНЫЕ СИСТЕМЫ		
1.	Справочно-правовая система Консультант+	http://www.consultant.ru
2.	Информационно-правовой портал Гарант	http://www.garant.ru
3.	Профессиональные справочные системы Техэксперт	http://www.cntd.ru

6.2. Материально-техническая база и программное обеспечение

№№ п/п	Аудитории для проведения учебных занятий, самостоятельной работы и проведения государственной итоговой аттестации	Перечень основного оборудования	Программное обеспечение
1.	531 (III)	ПК B112,2 420W/Intel Celeron Dual-Core E3300/кл,мышь,филт,мон. VA1931 (5)	Microsoft Windows Enterprise, Справочная правовая система "Консультант Плюс", Microsoft Office Standard, Агент Dr.Web, Комплект ГАРАНТ-Мастер, Microsoft Access, Microsoft Visio Professional, Microsoft Project Professional, Microsoft Visual Studio Enterprise, Комплект ПО для решения основных пользовательских задач

Раздел 7. ФОРМЫ КОНТРОЛЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ/ ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

Критерии оценивания индикаторов достижения компетенций направлены на:

- усвоение теоретического материала (объем знаний, глубина усвоения), предусмотренного рабочей программой;
- умение излагать материал (четкость, грамотность изложения материала, точность и полнота воспроизведения учебного материала);
- умение применять теоретические знания при решении практических заданий.

Шкала оценивания представлена ниже.

Уровень сформированности элементов компетенции	Критерии оценивания	Шкала оценивания
Пороговый уровень	Обучающийся имеет знания основного материала, проявляет умение логично его излагать, но может допускать неточности в изложении материала, недостаточно правильные формулировки, испытывает затруднения в выполнении практических заданий	Зачтено

7.1. Текущий контроль успеваемости

Текущий контроль успеваемости обеспечивает оценивание хода освоения дисциплины (модуля) и производится с применением технологии рейтингового контроля в соответствии с технологической картой дисциплины. Порядок составления технологической карты и алгоритм проведения процедуры оценивания видов деятельности обучающихся, направленных на освоение знаний, умений, навыков и/или опыта деятельности, по накопительной системе в баллах устанавливается положением о системе РИТМ в ФГБОУ ВО «ПГТУ»

7.2. Промежуточная аттестация обучающихся

Промежуточная аттестация обучающихся направлена на оценивание результатов обучения по дисциплине (модулю) и проводится с использованием фондов оценочных средств.

Примеры типовых контрольных заданий из базы фонда оценочных средств по образовательной программе.

Перечень вопросов для проведения промежуточной аттестации

1. Как создать поток выполнения (thread) в C++ с использованием STL?
2. Что такое мьютексы и зачем они нужны в параллельном программировании?
3. Какие алгоритмы из STL поддерживают параллельное выполнение, и как их использовать?
4. Как обеспечить безопасность данных при параллельном доступе к общим ресурсам?
5. Какие структуры данных можно использовать для обмена данными между потоками?
6. Какие атомарные операции предоставляет STL, и в каких случаях их следует использовать?
7. Как реализовать параллельный цикл с использованием STL?
8. Что такое пул задач (task pool), и как он может быть применен в параллельном программировании?
9. Как реализовать простую задачу MapReduce на C++ с использованием STL?
10. Как провести профилирование параллельной программы и выявить узкие места?
11. Какие инструменты можно использовать для отладки параллельного кода?
12. Как осуществляется параллельное программирование с использованием GPU в C++ с помощью CUDA?
13. Какие существуют оптимизации для улучшения производительности параллельных программ?
14. Какие библиотеки и фреймворки поддерживают параллельное программирование на C++?
15. Как оценить производительность параллельной программы и провести сравнение с последовательной версией?
16. Какие практические примеры применения параллельных вычислений в реальных проектах на C++ вы можете привести?
17. Какие преимущества и ограничения существуют при использовании параллельных вычислений в

различных типах задач?

18. Какие аспекты следует учитывать при разработке параллельных алгоритмов для многопроцессорных систем?

19. Какие существуют подходы к реализации параллельных алгоритмов с использованием сторонних библиотек и фреймворков?

20. Как можно оптимизировать параллельный код для повышения эффективности его выполнения?