

АННОТАЦИЯ ДИСЦИПЛИН

Направление подготовки– *«01.04.02. Прикладная математики и информатика» (МОП: Системное программирование)*

Год начала подготовки: 2025г.

№	Наименование дисциплины	Краткое описание	Код компетенции, код индикатора достижения компетенции
1	Иностранный язык	Программа курса иностранного языка носит коммуникативно-ориентированный характер. Его задачи определяются коммуникативными и профессиональными потребностями обучаемых. Цель курса - приобретение общей, коммуникативной и профессиональной компетенции. Коммуникативная компетенция включает лингвистический, социокультурный и прагматический компоненты. Соответственно, надо уметь соотносить языковые средства с конкретными сферами, ситуациями, условиями и задачами общения. Достижение профессиональных целей предполагает расширение кругозора студентов, повышение уровня специального образования, а также культуры мышления, общения и речи	УК-1 (УК-1.1, УК-1.2, УК-1.3) УК-2(УК-2.1, УК-2.2, УК-2.3) УК-3 (УК-3.1, УК-3.1, УК-3.3) УК-4(УК-4.1, УК-4.1, УК-4.3) УК-5(УК-5.1, УК-5.1, УК-5.3)

2	Основы ускоренных вычислений с использованием CUDA C/C ++	Использование графических процессоров для рендеринга хорошо известно, но их возможности для общих параллельных вычислений были изучены лишь недавно. Параллельные алгоритмы, работающие на графических процессорах, часто могут достигать 100-кратного ускорения по сравнению с аналогичными алгоритмами ЦП, причем во многих существующих приложениях для физического моделирования, обработки сигналов, финансового моделирования, нейронных сетей и множества других областей. В этом курсе рассматриваются методы программирования для графического процессора. На курсе будет представлен язык параллельных вычислений NVIDIA CUDA. Помимо рассмотрения модели и синтаксиса программирования CUDA, в курсе также будут обсуждаться архитектура графического процессора, высокопроизводительные вычисления на графических процессорах, параллельные алгоритмы, библиотеки CUDA и приложения вычислений на графическом процессоре. Наборы задач охватывают оптимизацию производительности и несколько конкретных примеров применения графических процессоров, таких как численная математика, медицинская визуализация, финансы и другие области	ПК-2 (ПК-2.1, ПК-2.2, ПК-2.3) ОПК-1(ОПК-1.1,ОПК-1.2, ОПК-1.3)
3	Стохастические системы	В курсе студенты осваивают основы стохастических систем и процессов, включая цепи Маркова, оптимальное управление, фильтрацию Калмана, методы обучения с подкреплением и	ПК-4 (ПК-4.1, ПК-4.2, ПК-4.3) ОПК-4(ОПК-4.1,ОПК-4.2, ОПК-4.3)

		аппроксимацию функций. Также будет введение в теорию игр и её применение.	
4	Параллельное программирование	Курс "Параллельное программирование" посвящен изучению теории и методов практической разработки параллельных программ для современных архитектур компьютеров. Особенностью курса является комплексное рассмотрение проблем параллелизма, как на уровне библиотек, предоставляемых прикладным программистам, так и на уровне близком к архитектуре микропроцессоров и графических ускорителей, что более важно для системных программистов.	УК-3 (ПК-4.1, ПК-4.2, ПК-4.3) ПК-7 (ПК-7.1, ПК-7.2, ПК-7.3)
5	Сложные структуры данных	Актуальность программы заключается в том, что слушатели знакомятся с многими продвинутыми структурами данных и алгоритмами, которые позволяют эффективным образом решать множество практических задач.	УК-2 (УК-2.1, УК-2.2, УК-2.3) ОПК-1(ОПК-1.1, ОПК-1.2, ОПК-1.3)
6	Научный семинар	Научный семинар является основой образовательной программы, поскольку помогает выстроить студенту индивидуальную траекторию обучения. В рамках курса происходит подбор темы индивидуального обучения, подбор проектных групп и консультантов. Семинар рассматривает основные проблемы и задачи разработки системного программного обеспечения, а также вопросы стандартизации процессов разработки и обеспечения качества программных систем. Основная форма работы семинара, это доклады его участников, на которых происходит обсуждение рассматриваемых научных и практических вопросов разработки системного программного обеспечения.	УК-2 (УК-2.1, УК-2.2, УК-2.3) УК-6 (УК-6.1, УК-6.2, УК-6.3) ОПК-2(ОПК-2.1, ОПК-2.2, ОПК-2.3) ПК-2 (ПК-2.1, ПК-2.2, ПК-2.3)

7	Численные методы и оптимизация	Численные методы являются основной составляющей частью вычислительной математики, на основе которых строятся алгоритмы численного решения задач алгебры и анализа, дифференциальных уравнений и др. Цель предмета «Численные методы и оптимизация» - изучение современных разделов теории и некоторых ее приложений.	ПК-2 (ПК-2.1, ПК-2.2, ПК-2.3)
8	Философия разработки ПО	Курс "Философия разработки ПО" исследует основные идеи и принципы, лежащие в основе создания программного обеспечения. Студенты изучают различные философские концепции, которые влияют на процессы разработки, такие как гибкость методологии, обеспечение качества программного продукта, этика в программировании, а также социальные и культурные аспекты технологий. Курс предоставляет студентам глубокое понимание основополагающих принципов, необходимых для эффективного применения их знаний и навыков в сфере информационных технологий.	ПК-7 (ПК-7.1, ПК-7.2, ПК-7.3)
9	Построение и анализ алгоритмов дискретной оптимизации	Алгоритмы и теоремы для типичных задач дискретной оптимизации. Задача нахождения максимального потока в сетях. Алгоритм Форда-Фалкерсона, анализ алгоритма. Модификация Карпа–Эдмонса. Теорема Кенига и алгоритм построения максимального паросочетания в двудольных графах. Теорема Дилворта и алгоритм раскраски графа интервалов. Теорема Гейла о спросе и предложении. Теорема Райзера о существовании 0-1 матриц. Венгерский алгоритм для задачи о назначениях. Матроиды. Примеры матроидов. Эквивалентные системы аксиом.	ПК-7 (ПК-7.1, ПК-7.2, ПК-7.3) ПК-8 (ПК-8.1, ПК-8.2, ПК-8.3) ПК-10 (ПК-10.1, ПК-10.2, ПК-10.3)

		Оптимизационные задачи на матроидах. Матроиды и жадный алгоритм. Метод Крацубы для умножения целых чисел. Алгоритм Штрасса. Приближенные полиномиальные алгоритмы для NP-трудных задач. Поведение жадного алгоритма для задач покрытия множества и коммивояжера с неравенством треугольника. Алгоритм Кристофидеса для задач коммивояжера с неравенством треугольника. Приближенно полиномиальные схемы для задач коммивояжера на плоскости и о рюкзаке. Поведение жадного алгоритма для задачи покрытия множества в типичном случае.	
10	Практические методы анализа ПО	Курс "Практические методы анализа ПО" представляет собой обзор основных инструментов и приемов, применяемых для анализа программного обеспечения на практике. В рамках обучения студенты изучают разнообразные методики статического и динамического анализа кода, тестирования ПО, а также методы обеспечения безопасности и повышения производительности. Курс включает как теоретические основы данных методов, так и их практическое применение с использованием современных инструментов разработки. По завершении обучения студенты приобретают необходимые навыки для успешного анализа и улучшения программных продуктов в реальных проектах.	ОПК-3(ОПК-3.1, ОПК-3.2, ОПК-3.3) ОПК-4(ОПК-4.1,ОПК-4.2, ОПК-4.3) ПК-11 (ПК-11.1, ПК-11.2, ПК-11.3) ПК-12 (ПК-12.1, ПК-12.2, ПК-12.3)
11	Современные базы данных	Курс посвящён изучению современных подходов к построению и к использованию систем	ПК-1 (ПК-1.1, ПК-1.2, ПК-1.3)

		управления базами данных. Курс разделён на две части. В первой части рассказывается об исторических тенденциях развития ранних СУБД, о реляционной модели, о языке SQL для прикладных задач. Вторая часть курса посвящена появлению NoSQL, распределённых и облачных СУБД, а также сложностям, которые встречаются на этом пути. Цель курса: научить студентов ориентироваться семействах СУБД от самых ранних до современных, а также уметь понимать, какие из них оправданно применять для решения разных классов задач, а какие нет.	ПК-7 (ПК-7.1, ПК-7.2, ПК-7.3)
12	Безопасность и надежность программного обеспечения	В рамках курса студенты будут изучать основы безопасной разработки программ. Будут изучаться известные программные ошибки и способы защиты программ от таких ошибок. Для прохождения данного курса необходимо базовые знания по C/C++.	ПК-3 (ПК-3.1, ПК-3.2, ПК-3.3) ОПК-3 (ОПК-3.1, ОПК-3.2, ОПК-3.3)
13	Введение в машинное обучение	В курсе рассматриваются основные задачи обучения по прецедентам: классификация, кластеризация, регрессия, понижение размерности. Изучаются методы их решения, как классические, так и новые, созданные за последние 10–15 лет. Упор делается на глубокое понимание математических основ, взаимосвязей, достоинств и ограничений рассматриваемых методов. Отдельные теоремы приводятся с доказательствами. Все методы излагаются по единой схеме: - исходные идеи и эвристики; - их формализация и математическая теория;	УК-1 (УК-1.1, УК-1.2, УК-1.3) УК-2(УК-2.1, УК-2.2, УК-2.3) УК-3 (УК-3.1, УК-3.1, УК-3.3) УК-4(УК-4.1, УК-4.1, УК-4.3) УК-5(УК-5.1, УК-5.1, УК-5.3) ПК-2 (ПК-2.1, ПК-2.2, ПК-2.3) ОПК-1 (ОПК-1.1, ОПК-1.2, ППК-1.3)

		- описание алгоритма в виде слабо формализованного псевдокода; - анализ достоинств, недостатков и границ применимости; - пути устранения недостатков; - сравнение с другими методами; - примеры прикладных задач.	ОПК-2(ОПК-2.1, ОПК-2.2, ОПК-2.3) ОПК-3(ОПК-3.1, ОПК-3.2, ОПК-3.3) ОПК-4(ОПК-4.1,ОПК-4.2, ОПК-4.3)
14	Введение в компиляторы	В курсе «Введение в компиляторы» рассматриваются основные принципы построения современных оптимизирующих компиляторов, причем основное внимание уделяется именно оптимизации кода — начиная от генерации внутреннего представления программы, анализа потока данных, различных этапов анализа и оптимизации кода, и заканчивая распределением регистров для конкретной целевой архитектуры и генерации для нее машинного кода. В ходе курса подробно рассматриваются алгоритмы компиляторных оптимизаций, а также разбираются примеры их работы. Рассматриваемые в курсе алгоритмы составляют основу оптимизаций, применяемых в современных промышленных компиляторах, таких как Clang/LLVM и GCC.	УК-1 (УК-1.1, УК-1.2, УК-1.3) УК-2(УК-2.1, УК-2.2, УК-2.3) УК-3 (УК-3.1, УК-3.1, УК-3.3) УК-4(УК-4.1, УК-4.1, УК-4.3) УК-5(УК-5.1, УК-5.1, УК-5.3) ПК-2 (ПК-2.1, ПК-2.2, ПК-2.3) ОПК-1 (ОПК-1.1, ОПК-1.2, ППК-1.3) ОПК-2(ОПК-2.1, ОПК-2.2, ОПК-2.3) ОПК-3(ОПК-3.1, ОПК-3.2, ОПК-3.3) ОПК-4(ОПК-4.1,ОПК-4.2, ОПК-4.3)
15	Введение в большие языковые модели (LLM)	Развитие больших языковых моделей (LLM) представляет собой одну из наиболее значимых инноваций в области обработки естественного языка (NLP). Эти модели, такие как GPT-4 от OpenAI, способны генерировать текст, схожий с человеческим, выполнять задачи перевода,	ОПК-4(ОПК-4.1,ОПК-4.2, ОПК-4.3) ПК-12 (ПК-12.1, ПК-12.2, ПК-12.3)

		резюмирования, ответа на вопросы и многое другое. В данном курсе студенты познакомятся с основами LLM, их архитектурой, применением и этическими аспектами. Курс начнется с обзора истории и эволюции языковых моделей, начиная от первых экспериментов в области NLP до современных передовых разработок. Студенты изучат архитектуру трансформеров, лежащую в основе большинства современных LLM, включая концепции внимания и механизмов самообучения. Особое внимание будет уделено различиям между моделями различных поколений, таким как GPT, BERT и их модификациями. Практическая часть курса будет сосредоточена на применении LLM в различных областях, включая генерацию текста, диалоговые системы и тд. Студенты будут работать с реальными примерами и данными, а также научатся настраивать и дообучать модели под специфические задачи. По завершении курса студенты смогут: 1. Понимать основные концепции и архитектуру больших языковых моделей. 2. Применять LLM для решения практических задач в области NLP. 3. Дообучать LLM для специфических задач.	
16	Методы формальной верификации	Курс посвящен формальной верификации программ и моделей компьютерных систем. Цели — ознакомление студентов с базовыми принципами и методами формальной верификации и формирование у студентов навыков необходимых для практического	ОПК-3(ОПК-3.1, ОПК-3.2, ОПК-3.3) ПК-5 (ПК-5.1, ПК-5.2, ПК-5.3)

		использования рассмотренных методов. В основе лежат два класса методов: дедуктивная верификация и проверка моделей; рассматриваются методы формализации семантики программ (операционная и аксиоматическая семантика, структуры Крипке), методы формальной спецификации требований (программные контракты, темпоральная логика линейного времени), методы доказательства корректности программ (метод Флойда, теоретико-автоматный подход к проверке моделей в явной и символической формах). Для практики используются инструменты Frama-C/AstraVer/Why3 (дедуктивная верификация), а также Spin и NuSMV (проверка моделей).	
--	--	--	--