

Факультет компьютерных технологий и прикладной математики

Направление и код подготовки/специальности (профиль): 01.03.02 Прикладная математика и информатика (Программирование и информационные технологии) / ОФО

Наименование и код дисциплины: Б1.О.08 Методы программирования	
Количество академических часов (аудиторные/внеаудиторные): 118/55,5	Количество зачетных единиц: 5
Предварительные требования для изучения дисциплины: нет	Уровень подготовки: бакалавриат
Язык обучения: русский	Вид занятий по дисциплине: лекции – 50 ак.час., лабораторные занятия – 68 ак.час., самостоятельная работа – 19,8 ак.час
Курс/семестр: 1/весенний	Вид аттестации: весенний семестр – зачет, экзамен
Образовательные технологии: коммуникативного обучения, разноуровневого (дифференцированного) обучения, модульного обучения, информационно-коммуникационные технологии, использования компьютерных программ, Интернет-технологии, проектная технология, игровая технология, развития критического мышления.	
Краткая аннотация к содержанию дисциплины: изучение фундаментальных структур данных (линейных, двунаправленных и кольцевых списков, двоичных деревьев, графов) и методов их обработки, а также приобретение практических навыков их реализации на языке программирования. Дисциплина формирует основы для применения современных средств вычислительной техники и программных технологий при решении задач в естественнонаучных, общепрофессиональных и специальных дисциплинах. Освоение теоретического материала и практическое применение структур данных, алгоритмов, файловых систем и контейнеров на базе современных ПК подготавливает студентов к использованию информационных систем и языков программирования в профессиональной деятельности.	
Темы лекционных и семинарских занятий: 1. Линейные динамические информационные структуры 2. Линейные двунаправленные связные списки 3. Кольцевые списки 4. Двоичные деревья 5. Файлы 6. Контейнеры 7. Обработка графов	
Полученные компетенции: Знать: • Математические основы алгоритмов (сложность, оптимизация). • Принципы работы структур данных (графы, деревья, STL-контейнеры). • Методы управления памятью (указатели, динамические структуры). Уметь: • Анализировать и выбирать оптимальные алгоритмы и структуры данных. • Реализовывать сложные структуры (графы, деревья) и алгоритмы на C++. • Проводить отладку, тестирование и оптимизацию кода. Владеть: • Навыками программирования на C++ (STL, работа с памятью, файлами). • Методами оценки и оптимизации производительности алгоритмов. • Техникой документирования и презентации решений.	