

АННОТАЦИЯ рабочей программы дисциплины **Б1.О.27«Методы программирования»**

Направление подготовки 02.03.02 Фундаментальная информатика и информационные технологии

Объем трудоемкости: 6 зач.ед.

Цель дисциплины:

Целью освоения дисциплины является приобретение знаний о базовых принципах объектно-ориентированного программирования и получение практических навыков программирования на языке высокого уровня.

Формирование компетенции обучающегося в области использования компьютера как средства управления информацией; изучение методов программирования для овладения знаниями в области технологии программирования; подготовка обучающихся к осознанному применению, как языков программирования, так и методов программирования.

Создание необходимой основы для использования современных средств вычислительной техники и прикладных программ при изучении студентами естественнонаучных, общепрофессиональных и специальных дисциплин.

Освоение, предусмотренного программой теоретического материала и приобретение практических навыков использования информационных систем, языков программирования и технологий на базе современных ПК.

Задачи дисциплины:

Основные задачи курса на основе системного подхода:

- знакомство с методами структурного и объектно-ориентированного программирования как наиболее распространенными и эффективными методами разработки программных продуктов;
- обучение разработке алгоритмов на основе структурного и объектно-ориентированного подхода;
- закрепление навыков алгоритмизации и программирования на основе изучения современных языков программирования;
- знакомство с основными структурами данных и типовыми методами обработки этих структур.

Место дисциплины в структуре ООП ВО

Дисциплина «Методы программирования» относится к «Обязательная часть» Блока 1 «Дисциплины (модули)» учебного плана.

Дисциплина «Методы программирования» является логически и содержательно-методически связана с такими дисциплинами как «Дискретная математика», «Теория графов и ее приложения», «Основы программирования». Данная дисциплина позволяет заложить основу для изучения программистских дисциплин профессионального цикла, предшествует таким дисциплинам как «Конструирование алгоритмов и структур данных». Является логически связанной с математическими дисциплинами, рассматривает объекты таких дисциплин как «Дискретная математика» с точки зрения программирования.

Для успешного освоения дисциплины студент должен иметь удовлетворительный уровень подготовки по дисциплине «основы программирования».

Обучающийся должен:

- знать основные конструкции языка программирования C++;
- уметь строить линейные и циклические алгоритмы, рекурсивные алгоритмы;

- владеть базовыми схемами использования структур данных: одномерные массивы, матрицы;
- уметь конструировать функции.

Требования к уровню освоения дисциплины

Изучение данной учебной дисциплины направлено на формирование у обучающихся следующих компетенций:

- УК-1** Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач
- ОПК-2** Способен применять компьютерные/суперкомпьютерные методы, современное программное обеспечение, в том числе отечественного происхождения, для решения задач профессиональной деятельности

Основные разделы дисциплины:

№	Наименование разделов (тем)
1	2
1.	Линейные динамические информационные структуры
2.	Стек. Очередь. Дек.
3.	Моделирование стека средствами языка C++.
4.	Моделирование очереди средствами языка C++.
5.	Кольцевой буфер.
6.	Реализация двух однотипных стеков.
7.	Однонаправленные списки. Операция вставки и удаления элемента.
8.	Двунаправленные списки. Построение, операция удаления и вставки элемента.
9.	Деревья. Дерево двоичного поиска. Обходы ДДП.
10.	Обработка дерева двоичного поиска.
11.	Система управления вводом-выводом. Буферизация ввода-вывода
12.	Общие операции над файлами
13.	Ориентированные графы. Представления ориентированных графов.
14.	Задача нахождения кратчайшего пути. Алгоритм Дейкстры.
15.	Нахождение кратчайших путей между парами вершин. Алгоритм Флойда.
16.	Поиск центра ориентированного графа.
17.	Обход ориентированных графов. Поиск в глубину.
18.	Алгоритм нахождения сильно связанных компонент.
19.	Представление неориентированных графов. Остовные деревья минимальной стоимости.
20.	Алгоритм Прима. Алгоритм Крускала.
21.	Контейнерный класс vector.
22.	Контейнерный класс list.
23.	Ассоциативные контейнеры set и map.
24.	Общая схема компиляции.

Курсовые работы: не предусмотрена

Форма проведения аттестации по дисциплине: зачет, экзамен

Автор Добровольская Н.Ю. доцент, канд. пед. наук, доцент