

АННОТАЦИЯ РАБОЧЕЙ ПРОГРАММЫ ДИСЦИПЛИНЫ

Дисциплина Б1.В.ОД.10 Программирование

Семестр: 5, 6, 7

Количество часов: 396

Количество зачетных единиц: 11

Курсовая работа: -

Промежуточная аттестация: зачет, зачет, экзамен

Место дисциплины в структуре ООП:

Дисциплина «Программирование» относится к дисциплинам вариативной части блока Дисциплины (модули) Б1.В.ОД.10 направления 09.03.03 *Прикладная информатика* направленность «Прикладная информатика в информационной сфере».

Дисциплина «Программирование» предполагает предварительное изучение обучающимися дисциплин: «Теория систем и системный анализ», «Компьютерная оптимизация», «Математика», «Теория вероятностей и математическая статистика», «Дискретная математика», «Физика», «Алгебра и геометрия», «Компьютерное решение задач имитационным методом», «Математическая логика», «Вычислительная математика», «Введение в теорию алгоритмов и алгоритмические языки», «Экономико-математические модели и методы», «Информационное общество и бизнес», «Проблемы и перспективы информационного общества», «Информационные системы и технологии», «Интеллектуальные информационные системы», «Информатика и программирование», имеет преемственную связь с дисциплинами: «Базы данных», «Информационная безопасность», «Защита информации», «Компьютерные справочно-правовые системы», «Автоматизированные бухгалтерские информационные системы», «Программирование в среде 1С: Предприятие», «Программирование в среде 1С: Бухгалтерия», «Мировые информационные ресурсы», «Организационное и правовое обеспечение информационной безопасности», «Проектирование информационных систем», «Проектный практикум», «Инженерия знаний».

Изучение дисциплины «Программирование» является основой для дальнейшего изучения дисциплин: «Сети и телекоммуникации», «Программная инженерия».

Цель дисциплины: освоение обучающимися базовых алгоритмов обработки информации и концепций программирования на языке высокого уровня.

Задачи:

- освоение основных алгоритмов обработки данных;
- ознакомление обучающихся с конструкциями языка программирования

- ния высокого уровня и технологией разработки программ на языке высокого уровня;
- изучение основных структур данных и алгоритмов их обработки;
 - изучение базовых концепций парадигм объектно-ориентированного и параллельного программирования;
 - применение современных программных средств при разработке программного обеспечения.

Содержание дисциплины:

Общее понятие алгоритма и краткий обзор существующих алгоритмических языков. Неформальный алгоритмический язык – псевдокод, максимально приближенный к естественному языку. Основные конструкции алгоритмического языка – алгоритм, ветвление, цикл; приводятся простейшие примеры программ на псевдокоде. Понятие переменной.

Алфавит, синтаксис и семантика языка программирования. Средства определения синтаксиса: расширенные формулы Бэкуса-Наура (РБНФ), синтаксические диаграммы. Классификация языков программирования по уровню абстракции. Обзор основных элементов языка программирования высокого уровня.

Интуитивное определение алгоритма. Основные свойства алгоритмов. Понятие исполнителя алгоритма. Основные методы разработки алгоритмов. Формальное определение алгоритма. Машина Тьюринга. Операции над машинами Тьюринга. Тезис Тьюринга. Нормальные алгорифмы Маркова. Принцип нормализации Маркова. Понятия самоприменимости алгоритма и алгоритмической неразрешимости.

Рекурсивные алгоритмы. Понятие рекурсивного алгоритма. Виды рекурсии. Структура рекурсивного алгоритма. Примеры рекурсивных алгоритмов. Реализация перебора с возвратом с помощью рекурсии. Реализация механизма рекурсии. Сравнение рекурсии и итерации. Рекурсивные данные. Бинарные деревья. Обходы бинарного дерева. Основные алгоритмы обработки бинарных деревьев (уничтожение дерева, вставки информации в дерево, удаление вершины из дерева, сравнения двух деревьев и др.).

Понятие абстрактного типа данных. Классификация структур данных. Последовательные списки: стек, очередь, дек. Связные списки: однонаправленный список, двунаправленный список, циклический список. Графы. Методы реализации графов: матрица смежности, матрица инцидентности, списки смежных вершин, список ребер.

Внутренняя сортировка (сортировка массивов). Понятие сложности алгоритма сортировки. Основные алгоритмы внутренней сортировки: пузырьковая сортировка, шейкерная сортировка, сортировка простым выбором, сортировка простыми вставками, сортировка Шелла, сортировка слиянием, быстрая сортировка. Внешняя сортировка (сортировка файлов). Отрезки файла. Операции разделения файла и слияния файлов. Некоторые алгоритмы внешней сортировки: многофазная сортировка, сбалансированное слияние. Ис-

пользование алгоритмов внутренней сортировки в сортировке последовательных файлов.

Поиск в массиве. Линейный поиск. Бинарный поиск. Поиск в таблице. Хеширование. Выбор хеш-функции. Разрешение коллизий: метод открытой адресации, метод цепочек.

Парадигма объектно-ориентированного программирования (ООП): концепции объекта и класса, инкапсуляции, наследования и полиморфизма. Спецификация видимости атрибутов и методов объекта. Механизмы раннего и позднего связывания. Статические и виртуальные методы. Иерархии классов. Шаблоны классов.

Событийно-управляемое программирование. Пользовательские и системные события в программе. Методы обработки и распространение событий. Управление параллелизмом с помощью механизма обработки событий. Прикладной программный интерфейс (Application Programming Interface), API-программирование. Методы обработки данных, основанные на компонентных технологиях. Промежуточное программное обеспечение (middleware).

Параллельная вычислительная система. Примеры больших задач. Режимы выполнения задач: последовательный, псевдопараллельный (разделение времени) и параллельный. Виды параллелизма: многопроцессорная, векторная и конвейерная обработка. Методика разработки параллельных алгоритмов. Модель «процессы-каналы» параллельной программы. Разделение вычислений на независимые части: параллелизм по данным и функциональный параллелизм. Выделение информационных зависимостей: локальные и глобальные, статические и динамические схемы передачи данных, структурные и произвольные, синхронные и асинхронные способы взаимодействия. Масштабирование подзадач. Распределение подзадач по процессорам вычислительной системы.

Основные конструкции алгоритмического языка Компонетный Паскаль – алгоритм, ветвление, цикл; простейшие примеры программ на псевдокоде. Понятие переменной.

Понятие типа переменной как множества значений, которые она может принимать, и набора операций, которые можно совершать со значениями. Наиболее важные базовые типы алгоритмического языка – целые и вещественные числа. Особенность представления целых чисел в компьютере как элементов кольца вычетов, интерпретация элементов кольца вычетов как неотрицательных чисел или чисел со знаком. Представление вещественных чисел в компьютере в плавающей форме, особенности арифметики чисел с плавающей запятой.

Символьные переменные и способы кодирования символов. Логический тип и логические выражения, отличие логических выражений от арифметических: сокращенное вычисление результата. Конструкция массива. Возможные способы представления текстовых строк.

Вычисление функции на последовательности элементов как фрагмент реальных программ. Общая схема вычисления функций на последовательно-

стях, основанная на понятии индуктивной функции и индуктивного расширения. Применение общей схемы иллюстрируется на примерах – вычисление суммы и максимума последовательности, схема Горнера вычисления значения многочлена и его производной и т.п.

Рассматривается схема построения цикла «пока» с помощью инварианта, т.е. утверждения, которое сохраняется при каждом выполнении тела цикла. Применение этой схемы дает возможность сознательно строить алгоритм и доказывать правильность его работы по тексту, не прибегая к тестированию. Применение схемы иллюстрируется на примерах: алгоритм Евклида вычисления наибольшего общего делителя, алгоритм быстрого возведения в степень, расширенный алгоритм Евклида, приближенное вычисление логарифма без использования разложения в ряд.

Рассматривается устройство компьютера, построенного по фон-Неймановской архитектуре. Приводятся основные составные части компьютера: процессор, оперативная память, шина, внешние устройства. Рассматриваются общие принципы построения и работы процессора, указываются важнейшие регистры процессора и алгоритм его работы. Дается классификация CISC и RISC-процессоров. Рассматривается аппаратный стек и его использование в командах вызова подпрограмм и для размещения локальных переменных.

Рассматривается способ записи программ на языке RTL (Register Transfer Language), представляющем собой Ассемблер, не зависящий от команд конкретного процессора. Приводятся примеры записи программ на RTL и на Ассемблере процессора Intel 80386. Кратко рассматриваются более сложные принципы работы компьютера: взаимодействие с внешними устройствами, асинхронные и синхронные прерывания, использование виртуальной памяти для поддержки параллельных задач, процессы, нити и их синхронизация.

Общие принципы построения Си-программы: разбиение проекта на h- и c-файлы, разделение интерфейса и реализации, использование препроцессора. Базовые типы языка Си, конструкции массива и указателя, позволяющие строить новые типы, а также модификаторы типов. Операции и выражения языка Си.

Основные типы данных. Описание различных типов, переменные и константы. Символьные строки. Препроцессор языка Си.

Основные операции. Дополнительные операции. Перечень операций языка Си.

Выражения. Простейшие выражения. Операторы. Составные операторы.

Эквивалентность типов. Преобразование типов. Неявное преобразование типа. Арифметические преобразования. Явные преобразования типов. Синтаксис типов.

Ввод и вывод одного символа. Буферы. Чтение одной строки. Чтение файла. Переключение и работа с файлами.

Оператор if. Расширение оператора «if». Операции отношения. Логические операции. Операция условия: «?:». Множественный выбор.

Цикл с предусловием. Цикл со счетчиком. Цикл с постусловием. Другие управляющие операторы. Структурное программирование.

Создание и использование функций. Аргументы функции. Возвращение значений. Локальные переменные. Нахождение адресов. Указатели, первое знакомство. Функции с переменным количеством аргументов.

Классы памяти и область действия. Автоматические переменные. Внешние переменные. Статические переменные. Внешние статические переменные. Регистровые переменные.

Общие сведения. Символические константы: #define. Использование аргументов с #define. Макроопределение и функция. Включение файла #include. Условная компиляция.

Указатели и массивы. Массивы. Указатели. Динамические объекты. Создание динамических объектов. Доступ к динамическим объектам. Строки – дополнительные сведения о связи между указателями и массивами. Инициализация массивов и классы памяти. Функции. Операции с указателями.

Строковые константы. Массивы символьных строк и их инициализация. Массив и указатель: различия. Указатели и строки. Ввод-вывод строк. Обработка строк.

Определение структурных переменных. Доступ к компонентам структуры. Поля битов в структурах. Объединения. Перечисления. Переменные структуры. Указатели структуры. Массив структур. Переименования типов.

Библиотека языка Си содержит множество функций и макроопределений. Библиотеки меняются от системы к системе, но есть ядро функций (стандартная библиотека). Распределение памяти.

Формальные и фактические параметры. Передача параметров по ссылке. Области видимости. Реализация стандартных операций поиска и сортировки с помощью функций.

В результате освоения дисциплины обучающийся должен обладать следующими компетенциями:

ОПК-3: способностью использовать основные законы естественнонаучных дисциплин и современные информационно-коммуникационные технологии в профессиональной деятельности (*знать* основные законы естественнонаучных дисциплин и современные информационно-коммуникационные технологии в профессиональной деятельности; *уметь* использовать основные законы естественнонаучных дисциплин и современные информационно-коммуникационные технологии в профессиональной деятельности; *владеть навыками* использования основных законов естественнонаучных дисциплин и современные информационно-коммуникационных технологий в профессиональной деятельности);

ОПК-4: способностью решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности (*знать* информационно-коммуникационные технологии; методы поиска, анализа, способы обра-

ботки и передачи информации; принципы обработки данных с помощью информационно-коммуникационных технологий; основные требования к информационной безопасности информационных систем; *уметь* применять информационные технологии для решения профессиональных задач с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности; *владеть навыками* работы с компьютером как средством управления информацией и решения стандартных задач профессиональной деятельности);

ПК-2: способностью разрабатывать, внедрять и адаптировать прикладное программное обеспечение (*знать* современные инструментальные средства и технологии программирования; *уметь* разрабатывать компоненты программных комплексов и баз данных; *владеть навыками* работы с сетевыми программными средствами общего назначения);

ПК-8: способностью программировать приложения и создавать программные прототипы решения прикладных задач (*знать* базовые алгоритмы для работы с информацией; технология разработки алгоритмов и программ; *уметь* работать с современными системами программирования; строить алгоритмы решения прикладных задач и оценивать их сложность; *владеть навыками* разработки пользовательских интерфейсов; программирования и тестирования программ).

Образовательные технологии:

В преподавании дисциплины «Программирование» применяются разнообразные интерактивные образовательные технологии в зависимости от вида и цели учебного занятия.

Теоретический материал излагается на лекционных занятиях в следующих формах:

- проблемно-ориентированные лекции;
- лекции-дискуссии.

Лабораторные занятия по дисциплине «Программирование» ориентированы на закрепление теоретического материала, изложенного на лекционных занятиях, а также на приобретение дополнительных знаний, умений и практических навыков осуществления аналитической и профессиональной деятельности с применением интерактивных форм обучения (моделирования деловых ситуаций, подготовка презентаций, работа в команде и др.).

Составитель: Л. Г. Гомбоев, канд. физ.-мат. наук, доцент, кафедра информатики и естественнонаучных дисциплин.