

АННОТАЦИЯ РАБОЧЕЙ ПРОГРАММЫ УЧЕБНОЙ ДИСЦИПЛИНЫ

Б.3.В.03 Программирование

Семестр: 5, 6

Количество часов: 144

Количество зачетных единиц: 4

Промежуточная аттестация: зачет, экзамен

Место дисциплины в структуре ООП:

Дисциплина «Программирование» относится к вариативной части профессионального цикла Б.3.В.03 учебного плана направления 230700 *Прикладная информатика* профиль «Прикладная информатика в информационной сфере».

Изучение дисциплины «Программирование» является основой для дальнейшего изучения дисциплин «Проектирование информационных систем», «Программная инженерия».

Цель и задачи освоения дисциплины:

Целью освоения дисциплины «Программирование» является освоение базовых алгоритмов обработки информации и концепций программирования на языке высокого уровня.

Для достижения поставленной цели необходимо решить следующие основные задачи:

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

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. Использование 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 последовательностях, 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 учебной дисциплины обучающийся должен обладать следующими общекультурными (ОК) и профессиональными (ПК) компетенциями:

ОК-10 способен использовать методы и средства для укрепления здоровья и обеспечения полноценной социальной и профессиональной деятельности (*знает* основные методы и средства укрепления здоровья и обеспечения полноценной социальной и профессиональной деятельности; *умеет* целенаправленно применять методы и средства укрепления здоровья и обеспечения полноценной социальной и профессиональной деятельности; *владеет* навыками организации мероприятий по укреплению здоровья и обеспечения полноценной социальной и профессиональной деятельности).

ОК-13 способен понимать сущность и значение информации в развитии современного информационного общества, сознавать опасности и угрозы, возникающие в этом процессе, соблюдать основные требования информационной безопасности, в том числе защиты государственной тайны (*знает* основные требования, предъявляемые к информационной безопасности; *умеет* применять на практике основные приемы по защите информации; *владеет* навыками защиты информации, составляющей государственную и коммерческую тайну).

ПК-3 способен использовать основные законы естественнонаучных дисциплин в профессиональной деятельности и эксплуатации современного электронного оборудования и информационно-коммуникационных технологий в соответствии с целями образовательной программы бакалавра (*знает* основные законы естественнонаучных дисциплин; цели образовательной программы бакалавра по избранной специальности; основы эксплуатации современного электронного оборудования и информационно-коммуникационных технологий; *умеет* использовать основные законы естественнонаучных дисциплин в профессиональной деятельности; *владеет* навыками эксплуатации электронного оборудования и информационных систем).

ПК-4 способен ставить и решать задачи с использованием современных информационно-коммуникационных технологий (*знает* современные информационно-коммуникационные технологии; методику использования современных информационно-коммуникационных технологий; *умеет* использовать современные информационно-коммуникационные технологии для решения практических задач; *владеет* навыками использования современных информационно-коммуникационных технологий в профессиональной деятельности).

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: . Г. Гомбоев, канд. физ.-мат. наук, доцент кафедры прикладной информатики