

АННОТАЦИЯ РАБОЧЕЙ ПРОГРАММЫ УЧЕБНОЙ ДИСЦИПЛИНЫ

Б.2.В.02 Введение в теорию алгоритмов и алгоритмические языки

Семестр: 4

Количество часов: 144

Количество зачетных единиц: 4

Промежуточная аттестация: экзамен

Место дисциплины в структуре ООП:

Дисциплина «Введение в теорию алгоритмов и алгоритмические языки» относится к вариативной части математического и естественнонаучного цикла Б.2.В.02 направления 230700.62 *Прикладная информатика* профиль «Прикладная информатика в информационной сфере».

Изучение дисциплины «Введение в теорию алгоритмов и алгоритмические языки» является основой для дальнейшего изучения дисциплин «Программирование», «Операционные системы», «Базы данных», «Программная инженерия».

Цель и задачи освоения дисциплины:

Цель изучения дисциплины «Введение в теорию алгоритмов и алгоритмические языки» – освоение базовых понятий и терминов программирования как науки.

Основными задачами при изучении дисциплины «Введение в теорию алгоритмов и алгоритмические языки» являются:

- 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; 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 алгоритмы 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. Вы-

деление 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.

В результате освоения учебной дисциплины обучающийся должен обладать следующими общекультурными (ОК) и профессиональными (ПК) компетенциями:

ОК-1 способен использовать, обобщать и анализировать информацию, ставить цели и находить пути их достижения в условиях формирования и развития информационного общества (знать результаты, полученные отечественными и зарубежными исследователями; уметь анализировать исходные данные, необходимые для расчета экономических и социально-экономических показателей, характеризующих деятельность хозяйствующих субъектов; владеть навыками и методами анализа исходных данных, необходимые для расчета экономических и социально-экономических показателей, и составления программ исследований).

ОК-5 способен самостоятельно приобретать и использовать в практической деятельности новые знания и умения, стремиться к саморазвитию (знать современные источники получения информации; уметь осуществлять поиск требуемой информации; владеть методами современных информационных технологий для решения своих профессиональных задач).

ПК-5 способен осуществлять и обосновывать выбор проектных решений по видам обеспечения информационных систем (знать назначение и виды ИС; состав функциональных и обеспечивающих подсистем ИС; модели и процессы жизненного цикла ИС; стадии создания ИС; уметь проводить сравнительный анализ и выбор ИКТ для решения прикладных задач и создания ИС; владеть инструментальными средствами моделирования предметной области, прикладных и информационных процессов).

ПК-21 способен применять системный подход и математические методы в формализации решения прикладных задач (знать методы и модели теории систем и системного анализа; уметь выбирать методы моделирования систем, структурировать и анализировать цели и функции систем управления, проводить системный анализ прикладной области; владеть навыками работы с инструментами системного анализа; навыками программирования в современных средах).

Образовательные технологии:

В преподавании дисциплины «Введение в теорию алгоритмов и алгоритмические языки» применяются разнообразные интерактивные образовательные технологии в зависимости от вида и цели учебного занятия.

Теоретический материал излагается на лекционных занятиях в форме проблемно-ориентированных лекций.

Лабораторные занятия по дисциплине «Введение в теорию алгоритмов и алгоритмические языки» 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, создание базы данных и др.).

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