Оценочные материалы для текущего контроля и промежуточной аттестации по дисциплине

Алгоритмы и структуры данных, 3, 4 курс

Код, направление подготовки	09.03.04 Программная инженерия
Направленность (профиль)	Программное обеспечение компьютерных систем
Форма обучения	заочная
Кафедра-разработчик	автоматики и компьютерных систем
Выпускающая кафедра	автоматики и компьютерных систем

Темы курсового проекта:

1. Класс «Многоугольник»
2. Класс «Хеш-таблица»
3. Класс «Бинарное дерево»
4. Класс «Черно-красное дерево»
5. Класс «AVL-дерево»
6. Класс «Граф»
7. Класс «Двудольный граф»
8. Класс «Список»
9. Класс «Двусвязный циклический список»
10. Класс «Стек»
11. Класс «Очередь»
12. Класс «Дек»
13. Класс «Последовательность»
14. Класс «Множество»
15. Класс «Битовое множество»
16. Класс «Словарь»
17. Класс «Мультисловарь»
18. Класс «Очередь с приоритетами»
19. Класс «Битовый вектор»
20. Класс «Вектор»

Задание на выполнение курсового проекта

В соответствии с выбранным вариантом необходимо:

1. Провести анализ предметной области возможного применения проектируемого класса*.
2. Провести анализ функциональности проектируемого класса.
3. Разработать интерфейс класса.
4. В соответствии с разработанным интерфейсом спроектировать тестовое приложение.
5. Выполнить проектирование класса**, обоснованно выбирая и реализуя: необходимые поля класса; методы класса, включив в обязательном порядке операции добавления, удаления, изменения, поиска отдельных элементов, входящих в класс, соответствующие предметной области класса.
6. Провести проектирование алгоритмов, лежащих в основе разрабатываемых методов.
7. Реализовать полученное проектное решение.

8. Реализовать тестовое приложение и провести тестирование разработанного и реализованного класса.
9. Провести исследование одной из операций (вставка, удаление, изменение, поиск) над элементами. Построить зависимость времени выполнения операции от числа элементов, над которыми она выполняется.
10. Оценить асимптотическую сложность реализованных алгоритмов вставки, удаления и поиска элементов класса.

* В ряде случаев для получения эффективного решения потребуется проектирование и реализация более одного класса.

** Для вариантов 8, 10, 11, 12 требуется провести проектирование классов, используя два подхода к реализации хранения элементов: с использованием связных структур и с использованием массивов.

Типовые вопросы к экзамену:

1. Структуры и алгоритмы обработки данных. Связь со структурным и объектно-ориентированным программированием. Абстрактные типы данных.
2. Сложность алгоритмов. Асимптотическая сложность.
3. Структура данных «Список» («Динамический массив», «Стек», «Очередь», «Дерево», «Множество», «Словарь», «Граф»). Операции. Способы реализации.
4. Алгоритм сортировки выбором (вставками, Шелла, подсчетом, поразрядной, Хоара, слиянием, пирамидальной). Основные свойства.
5. Хеширование. Хеш-функции. Коллизии.
6. Графы. Способы представления.
7. Алгоритмы на графах.
8. Поиск. Алгоритмы поиска в линейных структурах.
9. Алгоритмы поиска строк.
10. Деревья поиска. Сбалансированные деревья. Алгоритмы построения, добавления и удаления элемента.
11. Градиентный метод. Область применения.
12. Динамическое программирование.
13. P и NP задачи. Сводимость задач. Классы задач.
14. Методы решения NP задач.

Примерные практические задания промежуточной аттестации:

Формулировка задания: заданы некоторые исходные данные, требуется сформулировать ответ, содержащий конкретные сведения, связанные с решением вычислительной задачи (свойством структуры, алгоритма).

Решаемая задача направлена на выявление следующих знаний, умений, навыков (по отдельности или в сочетаниях):

– формализации вычислительной задачи;
– умение выполнить сравнительный анализ;
– определить асимптотические характеристики алгоритма/структуры данных;
– выделить ключевые алгоритмические свойства программы;
– показать знания определенных структур, алгоритмов обработки данных.

Примеры некоторых задач промежуточной аттестации:

– построить сбалансированное дерево (АВЛ-дерево, красно-черное дерево, B-дерево и др.), которое получится в результате удаления (добавления) узла с ключом X;
– некоторый абстрактный тип данных (АТД) определяет структуру данных «очередь» («стек»). Функции Push, Pop и Count позволяют поместить узел в контейнер, извлечь узел и узнать

количество узлов соответственно. Построить блок-схему алгоритма обхода бинарного дерева (графа) в ширину (глубину);

– дано дерево. Показать результат, который будет отображен на экране при работе функции прямого (обратного, симметричного) обхода;
– задан граф. Построить промежуточные матрицы смежности при работе алгоритма Флойда (Дейкстры), либо другие внутренние структуры при работе алгоритмов Крускала (Борувки, Прима и др.);
– указаны несколько способов реализации некоторого АТД, необходимо сравнить асимптотические характеристики ключевых алгоритмов работы с данным АТД;
– задано некоторое условие задачи, необходимо построить алгоритм (псевдокод, написать программу), которая реализует эту задачу.