

Домашнее задание по курсу подготовительной программы по программированию на C/C++

Александр Буданин

22 сентября 2017 г.

1. Домашнее задание №1

Написать программу, которая в качестве аргументов командной строки принимает целое число N (первый аргумент) и имена текстовых файлов (документов).

Необходимо вывести все слова, которые встречаются более чем в N документах (с указанием количества вхождений каждого слова).

Слова во входных файлах разделяются символами, для которых библиотечные функции `isspace()` или `ispunct()` возвращают ненулевое значение.

2. Домашнее задание №2

В аргументе командной строки передаётся имена текстовых файлов в которых записаны двумерные матрицы вещественных чисел.

Необходимо напечатать на стандартный выходной поток результат следующего алгоритма:

Умножение матриц, хранящихся во входных файлах.

Формат хранения матриц в файлах – не разреженный – в первой строке файла записано количество строк матрицы; во второй – количество столбцов матрицы; в последующих строках файла записаны элементы строк самой матрицы (по одной строке матрицы на одной строке файла).

В качестве внутреннего представления матрицы в памяти программы использовать разреженное представление – одномерный массив структур (полями которых являются индексы элемента матрицы и само значение элемента), упорядоченный по значениям пар индексов элементов матрицы (для быстрого поиска в нём).

Целевой алгоритм не должен зависеть от представления матрицы в файле и в памяти, т.е. для работы с матрицей должны быть реализованы и использованы в алгоритме функции: `double get_elem(void *matr, int row, int col)`, которая возвращает значение элемента матрицы по его индексам, и `void set_elem(void *matr, int row, int col, double elem)`, которая устанавливает значение элемента матрицы по его индексам

3. Домашнее задание №3

Написать программу, которая в качестве аргументов командной строки принимает имена двух текстовых файлов. В первом файле содержится последовательность целых чисел.

Необходимо записать во второй файл числа, содержащиеся в первом файле, упорядоченные в порядке возрастания.

При реализации алгоритма необходимо использовать динамический массив. Числа в первом файле разделяются символами, для которых библиотечная функция `isspace()` возвращает ненулевое значение.

4. Домашнее задание №4

Разработать классы для описанных ниже объектов. Включить в класс конструкторы, конструктор копирования, деструктор, методы `set(...)`, `get(...)`, `show(...)`. Память под строковые поля необходимо выделять динамически. Определить другие необходимые методы.

Customer:

- Фамилия
- Имя
- Отчество
- Адрес
- Телефон
- Номер кредитной карточки
- Номер банковского счета
- Сумма покупок за последнюю неделю.

Создать массив объектов. Вывести:

1. список покупателей в алфавитном порядке;
2. список покупателей, номер кредитной карточки которых находится в заданном интервале;
3. список покупателей с суммой покупок за неделю, превышающих заданную.

5. Домашнее задание №5

Реализовать класс «Многочлен» – **Polynom**

В классе степени n перегрузить операции сложения, вычитания, умножения, инкремента, декремента, индексирования, присваивания, вывода на экран.

Создать массив объектов класса. Передать его в функцию, вычисляющую сумму полиномов массива и возвращающую полином-результат, который выводится на экран в основной программе (использовать при этом реализованные перегруженные операции).

6. Домашнее задание №6

Реализовать абстрактный базовый класс «Фигура», содержащий методы вычисления площади и метод `show()` для отображения информации об объекте. Построить производные классы Square (квадрат), Circle (окружность), Rectangle (прямоугольник), содержащие методы для вычисления площади фигур.

Создать массив указателей на объекты базового класса и заполнить этот массив объектами производных классов.

Продемонстрировать работу виртуальных методов базового класса при проходе по этому массиву.

7. Домашнее задание №7

«Упорядоченное бинарное дерево» – Tree. Тип хранимых в дереве ключей – параметр шаблона. Написать несколько конструкторов, в том числе конструктор копирования.

В классе реализовать методы для вставки элемента в дерево, удаления элемента по ключу, поиска элемента по ключу, вывода на экран. Перегрузить операции сложения, вычитания, присваивания и вывода на экран.

Сформировать дерево, вывести содержимое его узлов в порядке возрастания, найти узел по ключу, вывести содержимое листьев дерева (вершин, не имеющих потомков).

Реализовать предложенный шаблонный класс (тип элементов, которые хранит класс – параметр шаблона; каждый объект класса может хранить элементы только одного типа). Продемонстрировать реализованную функциональность класса для работы с различными типами данных.

Исключительные ситуации необходимо обрабатывать через механизм исключений.