

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

Владимир Атаманов

22 сентября 2017 г.

1. Домашнее задание №1

Написать программу, которая в качестве аргументов командной строки принимает целое число N и имя текстового файла (документа). Документ состоит из абзацев.

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

Слова во входном файле разделяются символами, для которых библиотечные функции `isspace()` или `ispunct()` возвращают ненулевое значение. Считать, что абзацы в тексте разделяются символом `\n`.

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:

Мах-норма матрицы, находящейся во входном файле:

$$\max_i \sum_j |a_{ij}|$$

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 двумерный массив вещественных чисел.

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 аргумента командной строки принимает имя текстового файла, содержащего слова.

Необходимо вывести на стандартный выходной поток список всех различных слов в файле в порядке убывания количества вхождений слов (с указанием количества вхождений).

При реализации алгоритма необходимо использовать односвязный список, каждый узел которого должен содержать указатель на слово и количество вхождений этого слова.

Слова во входном файле разделяются символами, для которых библиотечные функции `isspace()` или `ispunct()` возвращают 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.

Account:

- Номер счёта
- Фамилия владельца
- Дата открытия
- Баланс
- Общая сумма операций со счётом

Создать массив объектов. Вывести:

1. список счетов, открытых позднее заданной даты;
2. список счетов данного владельца;
3. выбрать N счетов с самым большим балансом.

5. Домашнее задание №5

Реализовать класс «Дробь» – **Fraction**

В классе перегрузить операции сложения, вычитания, умножения, деления, присваивания, операции отношения и вывода на экран.

Создать массив объектов и передать его в функцию, которая изменяет каждый элемент массива с четным индексом путем добавления следующего за ним элемента массива, используя реализованные операции.

6. Домашнее задание №6

Реализовать базовый класс **Item** (единица хранения в библиотеке), содержащий данные-члены: `invNumber` – инвентарный номер и `taken` – взято на руки или имеется в наличии, а также методы:

```
virtual void Show(); //показать информацию о единице хранения
bool isAvailable(); // есть ли единица хранения в наличии ?
int GetinvNumber(); //возвращает инвентарный номер
void Take(); // операция «взять»
void Return(); // операция «вернуть»
```

Построить производные классы **Book** и **Magazin**. Класс **Book** содержит данные-члены: `author`, `title`, `publisher`, `year` и методы: `Author()`; `Title()`; `Publisher()`; `YearOfPublishing()`; `Show()`.

Класс **Magazin** включает данные-члены: `volume`; `number`; `year`; `title` и методы: `Volume()`; `Title()`; `Number()`; `Year()`; `Show()`.

Создать массив указателей на объекты базового класса и заполнить этот массив объектами производных классов. Вызвать метод `Show()` базового класса и просмотреть массив объектов.

7. Домашнее задание №7

«Очередь с двумя концами» – Deque. Написать несколько конструкторов, в том числе конструктор копирования. Элементы очереди хранятся в массиве. Если массив имеет фиксированную размерность, то предусмотреть контроль выхода за пределы массива. Если память выделяется динамически и ее не хватает, то увеличить размер выделенной памяти.

В классе реализовать методы добавления элементов в очередь (в начала и в конец очереди) и удаление элементов из очереди (из начала и из конца очереди). Перегрузить операции сложения, вычитания, присваивания и вывода на экран.

Создать массив объектов. Передавать объекты в функцию, которая удаляет из очереди первый (с головы очереди), третий, пятый и т. д. (нечётные) элементы.

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.