Министерство науки и образования Украины  
Днепропетровский национальный университет им. О. Гончара

**Курсовая работа**

по программированию

Исполнитель:  
студент гр. ПК-11-1з  
Шевцов Леонид Анатольевич

2012 г.

# Оглавление

Оглавление 2

Введение 3

Постановка задачи 4

Анализ требований и разработка алгоритма 4

Формат входных данных 5

Пример работы программы 6

Выводы 7

Использованная литература 8

Примечание 1. Исходный текст программы 9

# Введение

Целью данной работы является закрепление и проверка знаний, полученных на курсе «Основы программирования на С/С++».

В частности, работа уделяет внимание:

* Чтению и записи текстовых файлов.
* Структурным типам С.
* Строковому типу и строковым операторам.
* Динамическому выделению и освобождению памяти.
* Написанию алгоритмов с использованием условий и циклов.
* Модульному программированию.

# Постановка задачи

Написать программу, которая соответствует следующим требованиям:

1. Наличие описанного извне шаблона структуры name с двумя элементами: строкой для имени и строкой для фамилии.
2. Наличие описанного извне шаблона структуры «пассажир» с элементами: структурой name, ном\_рейса (строка), кол\_вещей(целый тип), общ\_вес(целый тип).
3. Функция main() должна объявлять массив структур «пассажир» в количестве S\_SIZE (первая строка входного файла) и выполнять инициализацию структуры «пассажир» из входного файла. Определить функции для выполнения заданий, описанных ниже, результаты поместить в выходной файл. Имена входного и выходного файла считывать с командной строки.
4. Определить:
   1. Рейс с наибольшим весом багажа.
   2. Пассажира с наибольшим количеством вещей.
   3. Список пассажиров и информацию про багаж, для заданного рейса.

# Формат входных данных

Условие задачи упоминает, что исходные данные о пассажирах должны храниться в текстовом файле, и в первой строчке этого файла должно быть записано число пассажиров.

Остаток файла занимают записи о пассажирах.

Для простоты, положим, что в каждой строчке файла записано по одному полю структуры «Пассажир», т.е. формат файла таков:

* Число пассажиров (целое число)
* Имя первого пассажира (строка)
* Фамилия –«- (строка)
* Номер рейса –«- (строка)
* Число предметов багажа –«- (целое число)
* Общий вес багажа –«- (целое число)
* Имя второго пассажира (строка)
* … и т.д.

Все строковые поля **должны** быть не пустыми.

Все числовые поля **должны** быть больше либо равны нулю.

Пробелы вокруг полей **не допускаются**.

Число строчек в файле **должно** быть не менее 1 + (число пассажиров)\*5.

Остаток строчек в файле могут игнорироваться программой.

# Анализ требований и разработка алгоритма

При вводе данных из файла необходимо убедиться, что файл имеет правильный формат.

В первую очередь, число пассажиров должно быть не меньше нуля для соответствия формату входного файла.

Кроме того, несмотря на то, что в общем случае входной файл с нулевым числом пассажиров имеет смысл, ни одну из поставленных задач нельзя решить

Чтобы выполнить пункты «a» и «с» условия, будет удобно разбить список пассажиров по рейсам.

Так или иначе, для выполнения первого пункта необходимо подсчитывать вес каждого рейса, т.е. заводить структуру данных для хранения соответствия рейса и веса; однако, поскольку в дальнейшем нам придется выбрать всех пассажиров рейса, почему бы не начать с построения соответствия «рейс – пассажиры»?

После того, как такое соответствие построено, остается:

* Для пункта «а» - обойти все рейсы, подсчитать вес и найти максимальный.
* Для пункта «с» - выбрать из карты соответствия нужный нам рейс.

Построение дополнительной структуры данных для соответствий требует её правильного освобождения в конце работы.

# Пример работы программы

# Выводы

В процессе работы были совершены попытки применить функциональный стиль программирования к языку C, а именно, карринг, и сделаны выводы, что это невозможно либо непрактично.

В процессе работы было обнаружено, что среда разработки Visual Studio никаких версий не поддерживает стандарт С99 (и не планирует), а ограничивается устаревшим С89/С90.

Одна из неприятно неподдерживаемых возможностей – это объявление переменных в любом месте функции, а не только в её начале.

Из этого сделан вывод, что при написании программ на С нужно пользоваться другими компиляторами, например, GCC или, на Windows, MinGW.

Другой сделанный вывод – не стоит вообще использовать С для программ, которые будут использоваться на Windows.

# Использованная литература

# Примечание 1. Исходный текст программы