Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (ТУСУР)

Кафедра экономической математики, информатики и статистики (ЭМИС)

СТРУКТУРЫ И УКАЗАТЕЛИ

Отчет по лабораторной работе по дисциплине «Объектно-ориентированное программирование»

Студент группы 549  Баулин С.К.  
«\_\_» \_\_\_\_\_\_\_\_\_\_\_ 2020 г.  
Кандидат физико-математических наук, доцент кафедры ЭМИС \_\_\_\_\_\_\_ Шельмина Е. А. «\_\_» \_\_\_\_\_\_\_\_\_\_\_ 2020 г.

\_\_\_\_\_\_

оценка

Томск 2020

Лабораторная работа №2

Структуры и указатели

Цель работы

Повторить работу со структурами и указателями в С++.

Краткий теоретический материал

Языки программирования C/C++ поддерживает определяемые пользователем структуры - структурированный тип данных. Он является собранием одного или более объектов (переменных, массивов, указателей, других структур и т.д.), которые для удобства работы с ними сгруппированы под одним именем.

Структуры:

- облегчают написание и понимание программ.

- помогают сгруппировать данные, объединяемые каким-либо общим понятием.

- позволяют группу связанных между собой переменных использовать как множество отдельных элементов, а также как единое целое.

Как и массив, структура представляет собой совокупность данных. но отличается от него тем, что к ее элементам (компонентам) необходимо обращаться по имени и ее элементы могуч быть различного типа. Структуры целесообразно использовать там. где необходимо объединить данные, относящиеся к одному объекту.

Определение структуры состоит из двух шагов:

- объявление шаблона структуры (задание нового типа данных, определенного пользователем);

- определение переменных типа объявленного шаблона.

Объявление шаблонов структур. Общий синтаксис объявления шаблона структуры: struct имя\_шаблона

{

тип1 имя\_переменной1;

тип1 имя\_переменной1;

//другие члены данных;

};

Имена шаблонов должны быть уникальными в пределах их области определения для того, чтобы компилятор мог различать различные типы шаблонов. Задание шаблона осуществляется с помощью ключевого слова struct, за которым следует имя шаблона структуры и список элементов, заключенных в фигурные скобки.

Имена элементов в одном шаблоне также должны быть уникальными. Однако в разных шаблонах можно использовать одинаковые имена элементов.

Задание только шаблона не влечет резервирования памяти компилятором. Шаблон представляет компилятору необходимую информацию об элементах структурной переменной для резервирования места в оперативной памяти и организации доступа к ней при определении структурной переменой и использовании отдельных элементов структурной переменной.

Определение структуры-переменной ничем не отличается от объявления обычной переменной с предопределенным типом. Общий синтаксис:

struct имя\_шаблона имя\_переменной;

Доступ к компонентам структуры. Доступ к полям осуществляется с помощью оператора «.» при непосредственной работе со структурой или «->» - при использовании указателей на структуру Эти операторы называются селекторами членов класса. Общий синтаксис для доступа к компонентам структуры следующий:

имя\_переменной\_структуры.член\_данных;

имя\_указателя->имя\_поля;

(\*имя\_указателя).имя\_поля;

При определении структурных переменных можно инициализировать пх поля. Эта возможность подобна инициализации массива и следует тем же правилами:

имя\_шаблона имя\_переменной\_структуры = {значение1, значение2,...};

Компилятор присваивает значение1 первой переменной в структуре, значение2 - второй переменной структуры п т.д., и тут необходимо следовать некоторым правилам:

- присваиваемые значения должны совпадать по типу с соответствующими полями структуры:

- можно объявлять меньшее количество присваиваемых значений, чем количество полей. Компилятор присвоит нули остальными полями структуры:

- список инициализации последовательно присваивает значения полям структуры, вложенных структур и массивов.

Копирование структур-переменных. Язык С(С++) позволяет оператору присваивания копировать

значения одной структуры-переменной в другую переменную. при условии, что обе структуры-переменные относятся к одному7 и тому же типу. Таким образом, единственный оператор может скопировать несколько членов данных, которые включают массивы и вложенные структуры.

Однако следует учитывать, что оператор присваивания выполняет то, что называется поверхностной копией в применении к структурам-переменных. Поверхностная копия представляет собой копирование бит за битом значений полей переменной-источника в соответствующие поля переменной-цели. При этом может возникнуть проблема с такими членами данных, как указатели, поэтому использовать поверхностное копирование структур надо осторожно.

Формулировка заданий с полученными результатами.

Задание 1. Реализовать решение задачи двумя способами: с помощью массива структур и с помощью указателя на структуру.

Ввести массив структур в соответствии с вариантом. Рассортировать массив в алфавитном порядке по первому полю, входящему в структуру. В программе реализовать меню:

1) Ввод массива структур;

2) Сортировка массива структур;

3) Поиск в массиве структур по заданном}' параметру;

4) Изменение заданной структуры;

5) Удаление структуры из массива:

6) Вывод на экран массива структур;

7) Выход.

Структура «Сотрудник»: фамилия, имя, отчество; должность; год рождения; заработная плата.

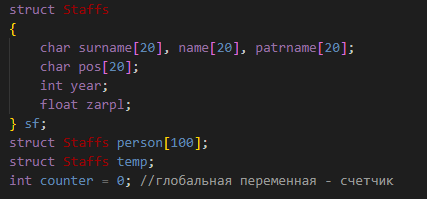
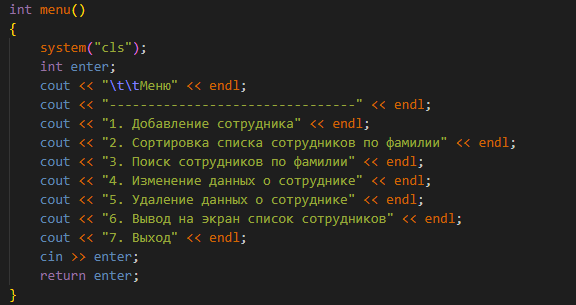
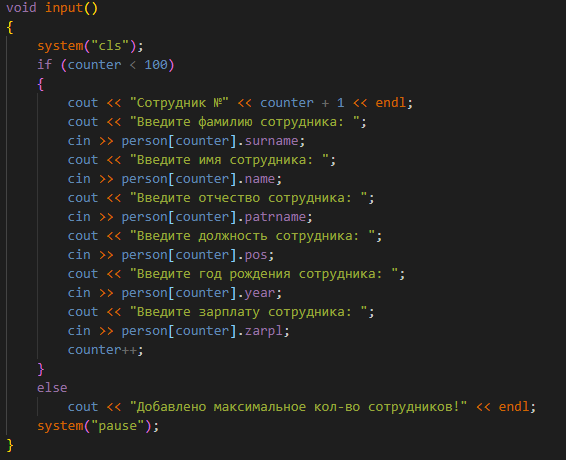
Код и результат выполненного задания представлены на рисунках 1-16.

Рисунок 1 – Скриншот структуры «Сотрудник»

Вывод

Повторена работа со структурами и указателями

Рисунок 2 – Скриншот функции меню

Рисунок 3 – Скриншот функции ввода

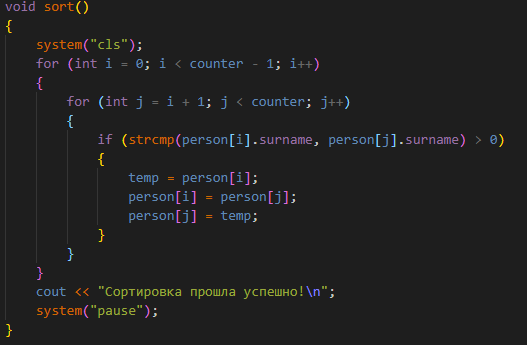
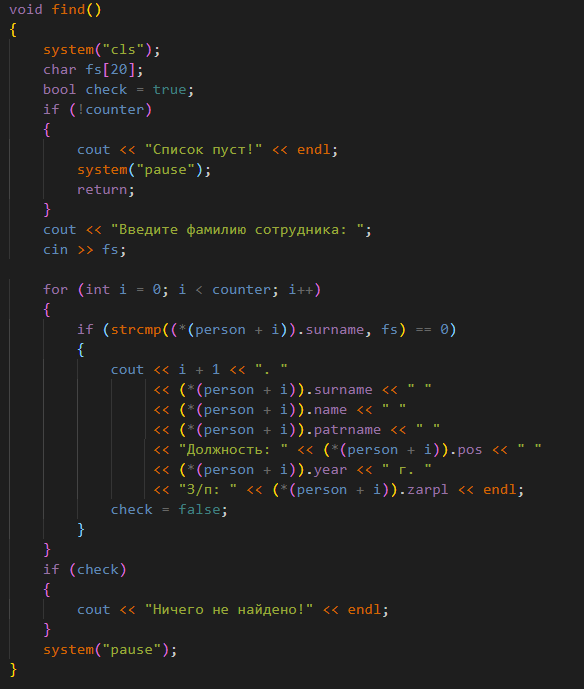
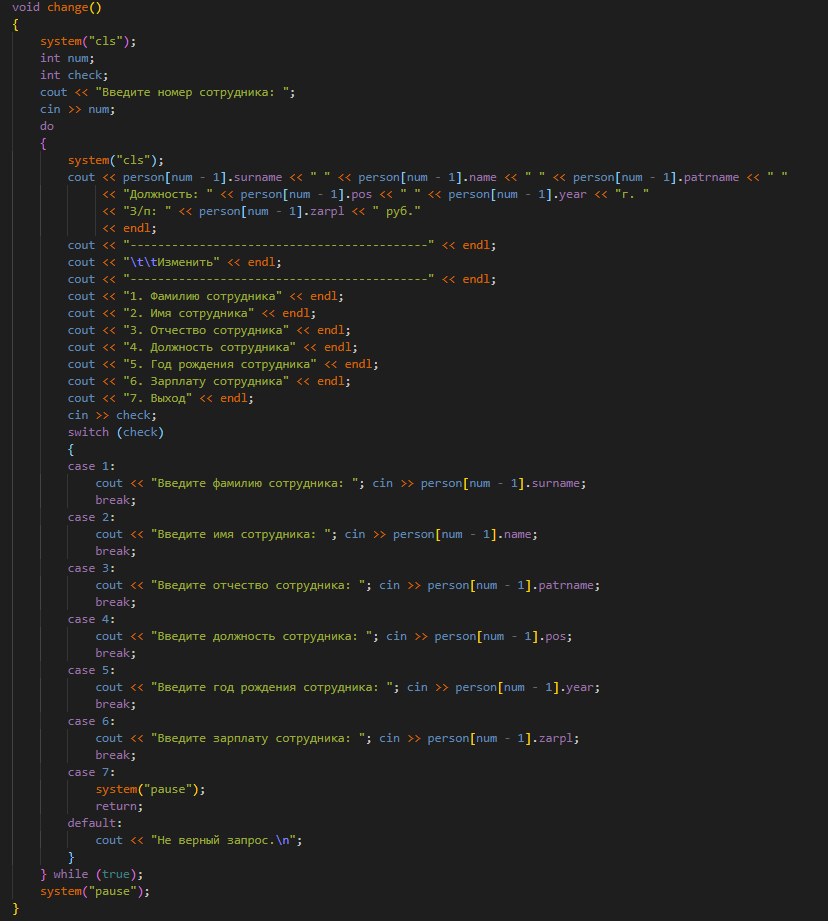
Рисунок 4 – Скриншот функции сортировки

Рисунок 5 – Скриншот функции поиска

Рисунок 6 – Скриншот функции изменения данных

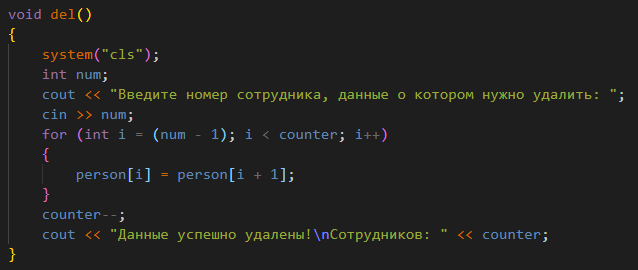
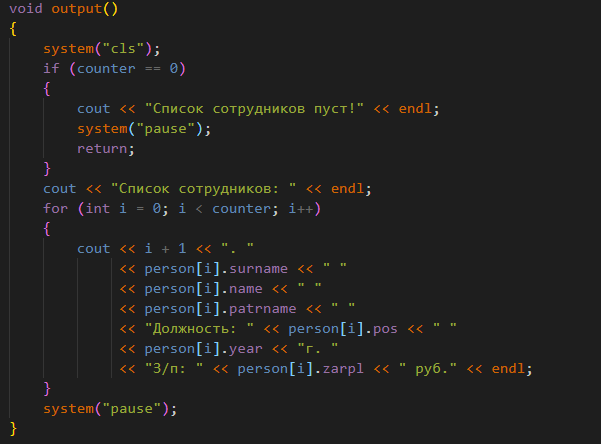
Рисунок 7 – Скриншот функции удаления

Рисунок 8 – Скриншот функции вывода

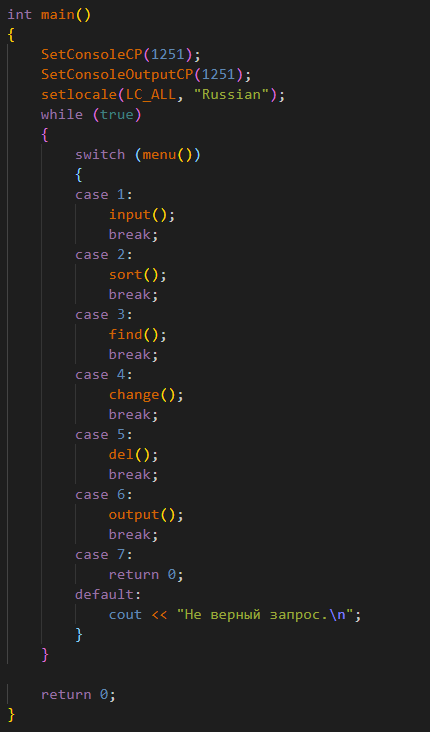
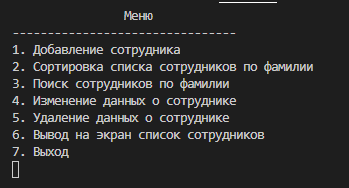
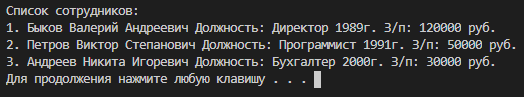
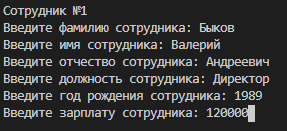
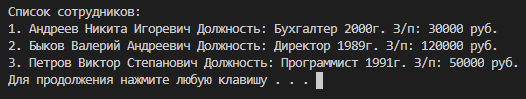
Рисунок 9 – Скриншот функции main

Рисунок 10 – Скриншот результата работы функции меню

Рисунок 11 – Скриншот результата работы функции ввода

Рисунок 12 – Скриншот результата работы функции вывода

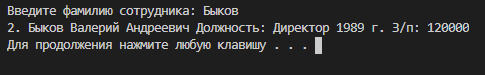
Рисунок 13 – Скриншот результата работы функции сортировки

Рисунок 14 – Скриншот результата работы функции поиска

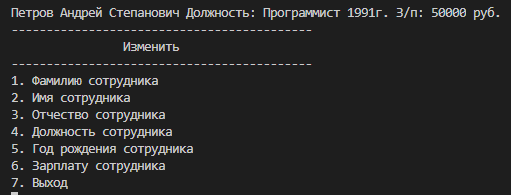
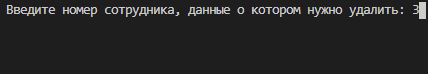
Рисунок 15 – Скриншот результата работы функции изменения

Рисунок 16 – Скриншот результата работы функции удаления