МИНОБРНАУКИ РОССИИ

Федеральное государственное автономное

образовательное учреждение высшего образования

«ЮЖНЫЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт компьютерных технологий и информационной безопасности

Кафедра информационно-аналитических систем безопасности

имени Л.С. Берштейна

**ОТЧЕТ**

**По лабораторной работе №5**

По курсу «Основы АиП»

**На тему**

«Массивы структур»

Вариант №4

Выполнил:

студент гр. КТбо1-8

Васильченко Н.С.

«\_\_\_» 2017 г.

Проверил:

Профессор ИКТИБ

кафедры ИАСБ

Беляков С.Л

«\_\_\_» 2017 г.

**Таганрог – 2017**

Оглавление

[Цель работы 3](#_Toc500510898)

[Вариант задания 4](#_Toc500510899)

[Алгоритм работы 5](#_Toc500510900)

[Примеры входных и выходных данных 6](#_Toc500510901)

[Заключение 7](#_Toc500510902)

[Код программы 8](#_Toc500510903)

# Цель работы

Цель данной лабораторной работы состоит в том, чтобынаучить студентов работать со сложными типами данных, каковыми являются структуры данных.

# Вариант задания

Вариант 4

«Футбольная команда»: название, город, количество сыгранных игр, количество очков (проигрыши, выигрыши, ничьи), количество игроков, фамилия тренера.

# Алгоритм работы

1. Подключаем библиотеки для ввода/вывода, задержки консоли, работы со строками.
2. Создаём структуру с полями разных типов.
3. В главной функции объявляем массив структур.
4. Далее создаем бесконечный цикл while.
5. Просим пользователя ввести цифру от 1 до 4 для определённого решения действий.
6. Если введён 0, то заполняем структуру.
7. Если введён 1, то выполняется упорядочение структуры, если она уже имеется.
8. Если введена 2, то выводит в файл структуру, если она уже имеется.
9. Если введена 3, то выводит на экран структуру, если она уже имеется.
10. Если введена 4, то осуществляется выход.
11. Пользователь сам заполняет структуру.
12. В зависимости от выбранного действия.
13. Эту структуру он может упорядочить по определённому элементу с помощью метода сортировки пузырьком.
14. Так же он может записать структуру в файл, используя цикл с параметрами.
15. Или вывести на экран, используя цикл с параметрами.
16. Также он может завершить программу.

# Примеры входных и выходных данных

|  |  |  |
| --- | --- | --- |
| Входные данные | Выходные данные | Выходные данные |
| Ввели структуру | Записывает в файл | Выводит на экран |
| sck rostov 10 15 21 petrov | sck Rostov 10 15 21 petrov | sck Rostov 10 15 21 petrov |
|  |  |  |
| Ввели структуры (так же происходит дополнение структуры) | | |
| sck rostov 10 15 21 petrov | lok taganrog 17 20 28 ivanov |  |
| Упорядочение(по убыванию) | | |
| lok taganrog 17 20 28 ivanov | sck rostov 10 15 21 petrov |  |

# Заключение

В данной работе мы познакомились со сложными типами данных, научились создавать массивы структур и обращаться к их элементам. Были изучены различные методы сортировки структур по определённому элементу и тем самым научились выводить упорядоченные структуры на экран и в файл.

# Код программы

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <stdlib.h>

#include <conio.h>

#include <cstring>

#include <algorithm>

struct Football

{

char name[30];

char city[30];

int numberOfGamesPlayed;

int points;

int numberOfPlayers;

char surnameOfTheTrainer[100];

};

void printOutputMonitor(Football \*play);

void inputUser(Football \*play);

void outputToFile(Football \*play);

void organizeStructuresPoints(Football \*play);

void organizeStructuresGames(Football \*play);

int cap = -1;

int main()

{

Football play[1024];

while (true)

{

printf("vvedite 0 - dopolnit strukt, 1 - yporiadothit po ochkam, 2 - yporiadothit po igram, 3 - vivod v file, 4 - vivod na monitor, 5 - exit");

int c = 0;

scanf("%d", &c);

if (c == 0)

{

inputUser(play);

}

else if ((c == 1)&&(cap != -1))

{

organizeStructuresPoints(play);

}

else if ((c == 2) && (cap != -1))

{

organizeStructuresGames(play);

}

else if ((c == 3) && (cap != -1))

{

outputToFile(play);

}

else if ((c == 4) && (cap != -1))

{

printOutputMonitor(play);

}

else if (c == 5)

{

return 0;

}

}

}

void inputUser(Football \*play)

{

cap++;

printf("\n enter all the data in turn\t");

printf("\n enter name\t");

scanf("%s", &play[cap].name);

printf("\n enter city\t");

scanf("%s", &play[cap].city);

printf("\n enter number of games played\t");

scanf("%d", &play[cap].numberOfGamesPlayed);

printf("\n enter points\t");

scanf("%d", &play[cap].points);

printf("\n enter number of players\t");

scanf("%d", &play[cap].numberOfPlayers);

printf("\n enter surname of the trainer\t");

scanf("%s", &play[cap].surnameOfTheTrainer);

}

void outputToFile(Football \*play)

{

FILE \*Out = fopen("C:/Users/Никита/source/repos/LAB 5-4/Debug/output.txt", "w");

for (int i = 0; i < cap + 1; i++)

{

fprintf(Out, "%s\t", &play[i].name);

fprintf(Out, "%s\t", &play[i].city);

fprintf(Out, "%d\t", &play[i].numberOfGamesPlayed);

fprintf(Out, "%d\t", &play[i].points);

fprintf(Out, "%d\t", &play[i].numberOfPlayers);

fprintf(Out, "%s\t", &play[i].surnameOfTheTrainer);

}

fclose(Out);

}

void organizeStructuresPoints(Football \*play)

{

int c = 0;

for (int i = 0; i < cap; i++)

{

for (int j = cap - 1; j >= i; j--)

{

if (play[j].points < play[j + 1].points)

{

c = play[j].points;

play[j].points = play[j + 1].points;

play[j + 1].points = c;

std::swap(play[j], play[j + 1]);

}

}

}

}

void organizeStructuresGames(Football \*play)

{

int c = 0;

for (int i = 0; i < cap; i++)

{

for (int j = cap - 1; j >= i; j--)

{

if (play[j].numberOfGamesPlayed < play[j + 1].numberOfGamesPlayed)

{

c = play[j].numberOfGamesPlayed;

play[j].numberOfGamesPlayed = play[j + 1].numberOfGamesPlayed;

play[j + 1].numberOfGamesPlayed = c;

std::swap(play[j], play[j + 1]);

}

}

}

}

void printOutputMonitor(Football \*play)

{

for (int i = 0; i < cap; i++)

{

printf("%s\t", &play[i].name);

printf("%s\t", &play[i].city);

printf("%d\t", &play[i].numberOfGamesPlayed);

printf("%d\t", &play[i].points);

printf("%d\t", &play[i].numberOfPlayers);

printf("%s\t", &play[i].surnameOfTheTrainer);

}

}