Пензенский государственный университет

Кафедра "Вычислительной техники"

**Отчёт**

По лабораторной №1

по дисциплине "Л и ОА в ИТ"

на тему "Простые структуры данных"

***Выполнил студент группы 19ВВ1:***

*Кочетов Д.В.*

***Приняли:***

*Митрохин М. А.*

*Юрова О. В.*

Пенза 2020

# Цель работы: изучить простые структуры данных в языке Си.

**Лабораторное задание:**

***Задание 1:*** написать программу, вычисляющую разницу между максимальным и минимальным элементами массива.

***Задание 2****:* написать программу, реализующую инициализацию массива случайными числами.

***Задание 3****:* написать программу, реализующую создание массива произвольного размера, вводимого с клавиатуры.

***Задание 4****:* написать программу, вычисляющую сумму значений в каждом столбце (или строке) двумерного массива.

***Задание 5:*** написать программу, осуществляющую поиск среди структур student структуру с заданными параметрами (фамилией, именем и т.д.).

**Теоретическая часть:**

К простым структурам данных языка Си относятся массивы, строки (массивы символов). ***Структуры*** (struct) – простейшая составная структура данных.

***Массив*** – это конечная совокупность данных одного типа.

Массивы могут состоять из целых чисел, чисел с плавающей запятой, символов и других типов данных. Существуют даже массивы массивов, – многомерные массивы.

***Динамический массив.*** Для использования функций динамического распределения памяти необходимо подключение библиотеки <malloc.h> или <stdlib.h> (в зависимости от компилятора и используемого стандарта языка).

*Функции динамического выделения памяти*

void\* malloc(*РазмерМассиваВБайтах*);  
 void\* calloc(*ЧислоЭлементов*, *РазмерЭлементаВБайтах*);

выделяют блок памяти, размером *РазмерМассиваВБайтах* или *ЧислоЭлементов*\**РазмерЭлементаВБайтах* байт, и возвращает указатель на начало блока.

Элементами массива могут быть как данные простых типов, так и составных типов (например, структуры), поэтому для точного определения размера элемента массива в общем случае используется функция

int sizeof(*ТипЭлемента*);  
Эта функция возвращает количество байт, занимаемое элементом указанного типа.

Память, динамически выделенная с использованием функций calloc(), malloc(), должна быть освобождена после окончания использования функцией

 free(*указатель*);

Для ***генерации случайных чисел*** в языке Си используется функция rand(), которая содержится в библиотеке <stdlib.h>.

При повторных запусках программы функция rand() будет возвращать одинаковые последовательности чисел. Для получения уникальных последовательностей необходимо инициализировать генератор случайных чисел различными значениями. Для этого используется функция srand(), аргументом которой является инициализирующее целое число.

***Структура*** - это совокупность переменных одного или нескольких типов, сгруппированных в один элемент.

**Практическая часть:**

**Листинг:**

Файл: Lab1.cpp

#include "stdafx.h"

#include "time.h"

#include "stdlib.h"

#include "conio.h"

#include "locale.h"

// Функция lab1f() реализует выполнение Задания 1

void lab1f() {

const int ARRAY\_SIZE = 20;

int max = 0, min = 0, A[ARRAY\_SIZE] = {0};

for(int i = 0; i < ARRAY\_SIZE; i++) {

A[i] = 1 + rand() % 250;

if( max < A[i]) max = A[i];

}

min = \*A;

for(int i = 0; i < ARRAY\_SIZE; i++) {

if( min > A[i]) min = A[i];

}

printf("Разность между наибольшим и наименьшим элементами массива: %d", max - min);

}

// Функция lab2f() реализует выполнение Задания 2

void lab2f() {

const int ARRAY\_SIZE = 20;

int A[ARRAY\_SIZE] = {0};

printf("\nМассив, заполненный произвольными значениями\n");

for(int i = 0; i < ARRAY\_SIZE; ) {

A[i] = rand();

printf("%d ", A[i++]);

}

}

// Функция lab3f() реализует выполнение Задания 3

void lab3f() {

int size = 0, \*A = NULL;

printf("\nВведите размер массива: ");

scanf("%d", &size);

A = (int\*)malloc(sizeof(int) \* size);

for(int i = 0; i < size; i++) {

\*(A + i) = rand();

printf("%d\n", \*(A + i));

}

free(A); A = NULL;

}

// Функция lab4f() реализует выполнение Задания 4

void lab4f() {

int M[15][8] = {0};

int colSum = 0, rowSum = 0;

for(int i = 0; i < 15; i++) {

for(int j = 0; j < 8; j++) {

M[i][j] = rand() % 399;

printf("%d\t", M[i][j]);

}

printf("\n");

}

for(int i = 0; i < 15; ) {

for(int j = 0; j < 8; j++) {

rowSum += M[i][j];

}

printf("Сумма значений %d-й строки: %d\n", ++i, rowSum);

rowSum = 0;

}

for(int j = 0; j < 8; ) {

for(int i = 0; i < 15; i++) {

colSum += M[i][j];

}

printf("Сумма значений %d-го столбца: %d\n", ++j, colSum);

colSum = 0;

}

}

int \_tmain(int argc, \_TCHAR\* argv[])

{

setlocale(LC\_ALL, "Rus");

srand(time(NULL));

lab1f();

lab2f();

lab3f();

lab4f();

getch();

return 0;

}

Файл lab1\_5.cpp

// Файл lab1\_5.cpp реализует выполнение Задания 5

#include "stdafx.h"

#include "Windows.h"

#include "conio.h"

#define ARR\_SZ 3

struct student {

char famil[20];

char name[20];

char facult[20];

int Nomzach;

} stud[ARR\_SZ];

void famSearch(student\* A) {

char sp[20];

printf("Введите фамилию студента, запись о котором хотите найти: ");

scanf("%s", sp);

for(int i = 0; i < ARR\_SZ; i++) {

if(!stricmp(sp, A[i].famil))

printf("Cтудент %s %s обучается на факультете %s, номер зачётной книжки %d \n",stud[i].famil,stud[i].name, stud[i].facult,stud[i].Nomzach);

}

}

void nameSearch(student\* A) {

char sp[20];

printf("Введите имя студента, запись о котором хотите найти: ");

scanf("%s", sp);

for(int i = 0; i < ARR\_SZ; i++) {

if(!stricmp(sp, A[i].name))

printf("Cтудент %s %s обучается на факультете %s, номер зачётной книжки %d \n",stud[i].famil,stud[i].name, stud[i].facult,stud[i].Nomzach);

}

}

void facultSearch(student\* A) {

char sp[20];

printf("Введите название факультета: ");

scanf("%s", sp);

for(int i = 0; i < ARR\_SZ; i++) {

if(!stricmp(sp, A[i].facult))

printf("Cтудент %s %s обучается на факультете %s, номер зачётной книжки %d \n",stud[i].famil,stud[i].name, stud[i].facult,stud[i].Nomzach);

}

}

void nzSearch(student\* A) {

int sp = 0, i = 0;

printf("Введите номер зачётной книжки: ");

scanf("%d", &sp);

while (sp != A[i].Nomzach) {

i++;

}

printf("Cтудент %s %s обучается на факультете %s, номер зачётной книжки %d \n",stud[i].famil,stud[i].name, stud[i].facult,stud[i].Nomzach);

}

int \_tmain(int argc, \_TCHAR\* argv[])

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

int i = 0;

for(i=0; i<ARR\_SZ; i++) {

printf("Введите фамилию студента\n");

scanf("%20s",stud[i].famil);

}

for(i=0; i<ARR\_SZ; i++) {

printf("Введите имя студента %s\n",stud[i].famil);

scanf("%20s",stud[i].name);

}

for(i=0; i<ARR\_SZ; i++) {

printf("Введите название факультета студента %s %s\n",stud[i].famil,stud[i].name);

scanf("%20s",stud[i].facult);

}

for(i=0; i<ARR\_SZ; i++) {

printf("Введите номер зачётной книжки студента %s %s\n",stud[i].famil,stud[i].name);

scanf("%d",&stud[i].Nomzach);

}

for(i=0;i<ARR\_SZ;i++) {

printf("Cтудент %s %s обучается на факультете %s, номер зачётной книжки %d \n",stud[i].famil,stud[i].name, stud[i].facult,stud[i].Nomzach);

}

printf("Выберите значение, по которому хотите искать:\n\n1 – фамилия;\n2 – имя;\n3 – факультет;\n4 – номер зачётной книжки\n");

scanf("%d", &i);

switch(i) {

case 1:

famSearch(stud);

break;

case 2:

nameSearch(stud);

break;

case 3:

facultSearch(stud);

break;

case 4:

nzSearch(stud);

break;

default:

printf("Неизвестный идентификатор.");

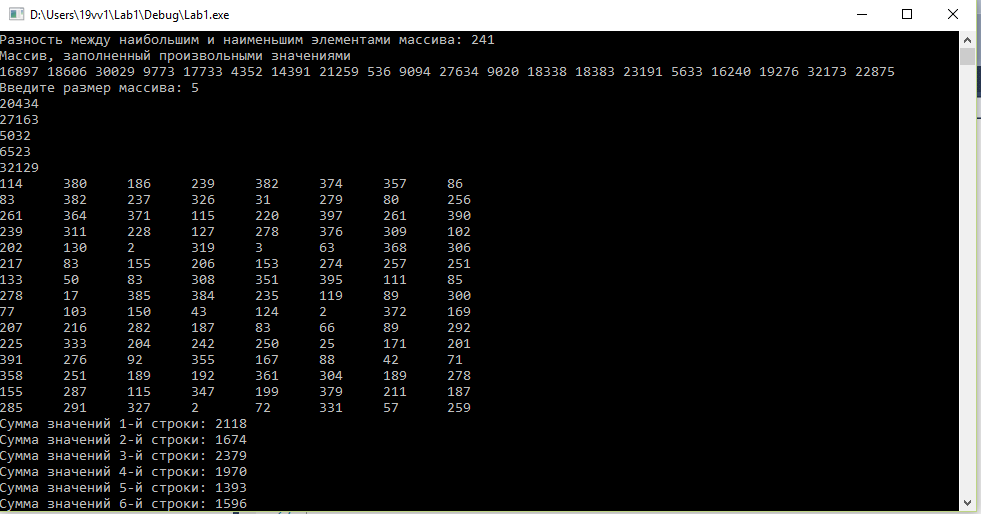
}

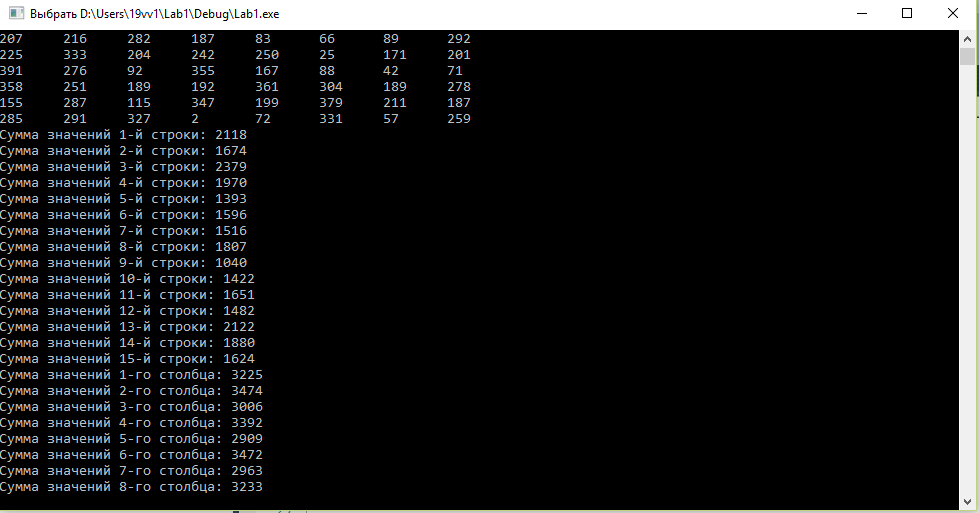
getch();

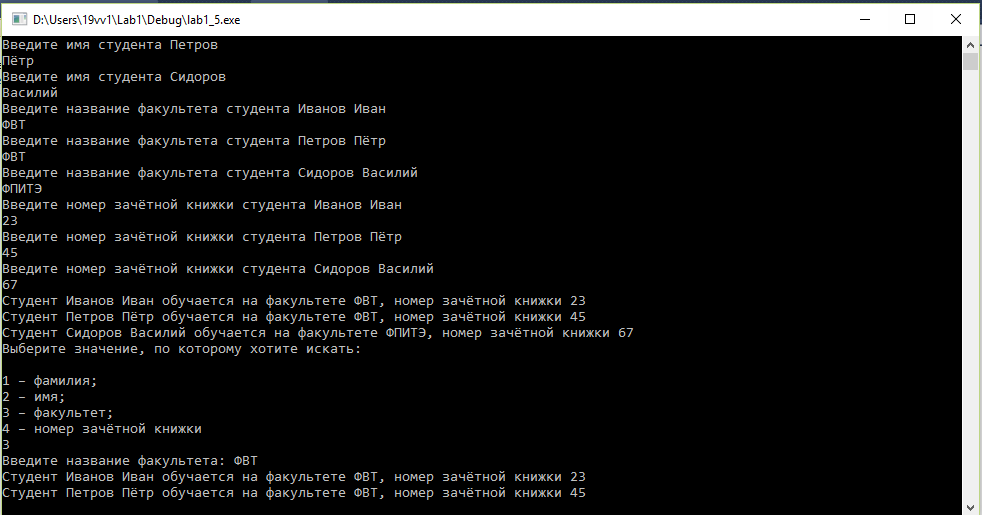
return 0;

}

**Результаты работы программы:**

****

****

****

**Вывод:** я изучил простые структуры данных в языке Си на практике.