Министерство образования Российской Федерации

Пензенский государственный университет

Кафедра «Вычислительная кафедра»

**ОТЧЕТ**

по лабораторной работе №1

по курсу «Логика и основы алгоритмизации в инженерных задачах»

на тему «Простые структуры данных»

Выполнили:

студент группы 23ВВВ2

Кокарев Д.С.

Приняли:

Юрова О.В.

Митрохин М.А.

Пенза 2024

**Название**

Простые структуры данных.

**Цель работы**

Изучение простых структур данных.

**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.).

**Описание метода решения задачи**

Задание 1. Находим разницу между максимальным и минимальным элементами массива. Инициализируются переменные min и max, чтобы хранить минимальное и максимальное значения массива. Во втором цикле происходит сравнение текущего элемента массива с текущими значениями min и max. Если элемент меньше текущего значения min, он обновляет min. Если больше — обновляет max.

Задание 2. Создаем массив аrray[10] и инициализируем его случайными числам, используя функцию rand.

Задание 3. Пользователь вводит нужный размер массива, используя динамический массив int\* array = new int[size]. Это позволяет создать массив произвольного размера. После массив заполняется случайными числами и выводится на экран.

Задание 4. При запуске нужно ввести необходимое количество строк и столбцов для двумерного массива. После заполнения случайными числами от 1 до 9, подсчитывается сумма во всех рядах и столбцах и выводится на экран.

Задание 5. Структура Student содержит поля Nomzach (номер зачетки), famil, name, facult. Программа запрашивает у пользователя количество студентов, которые нужно добавить, и выделяет динамическую память под массив студентов. Затем она поочерёдно запрашивает информацию о каждом студенте (фамилию, имя, факультет и номер зачётной книжки). После ввода данных программа запрашивает фамилию и имя для поиска студента. Используя функцию findStudent, программа выполняет поиск и выводит результат.

**Листинг**

**lab1.cpp**

//Задание 1

#include <iostream>

#include <limits>

int main() {

srand(time(NULL));

int min = INT\_MAX;

int max = INT\_MIN;

int array[10] = {};

for (int i = 0; i < 10; ++i) {

array[i] = (i \* 2) % 3;

}

for (int i = 0; i < 10; ++i) {

if (array[i] < min) {

min = array[i];

}

if (array[i] > max) {

max = array[i];

}

}

std::cout << "Array values: ";

for (int i = 0; i < 10; ++i) {

std::cout << array[i] << ' ';

}

std::cout << std::endl;

std::cout << "Min is " << min << std::endl;

std::cout << "Max is " << max << std::endl;

std::cout << "Difference between max and min: " << max - min << std::endl;

return 0;

}

//Задание 2

#include <iostream>

#include <limits>

int main() {

srand(time(NULL));

int min = INT\_MAX;

int max = INT\_MIN;

int array[10] = {};

for (int i = 0; i < 10; ++i) {

array[i] = 1 + rand() % 100;

}

for (int i = 0; i < 10; ++i) {

if (array[i] < min) {

min = array[i];

}

if (array[i] > max) {

max = array[i];

}

}

std::cout << "Array values: ";

for (int i = 0; i < 10; ++i) {

std::cout << array[i] << ' ';

}

std::cout << std::endl;

std::cout << "Min is " << min << std::endl;

std::cout << "Max is " << max << std::endl;

std::cout << "Difference between max and min: " << max - min << std::endl;

return 0;

}

//Задание 3

#include <iostream>

#include <limits>

int main() {

srand(time(NULL));

int min = INT\_MAX;

int max = INT\_MIN;

int size;

std::cout << "Enter array size: ";

std::cin >> size;

int\* array = new int[size];

std::cout << "Array values: ";

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

array[i] = 1 + rand() % 100;

std::cout << array[i] << ' ';

}

std::cout << std::endl;

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

if (array[i] < min) {

min = array[i];

}

if (array[i] > max) {

max = array[i];

}

}

std::cout << "Min is " << min << std::endl;

std::cout << "Max is " << max << std::endl;

std::cout << "Difference between max and min: " << max - min << std::endl;

delete[] array;

return 0;

}

//Задание 4

#include <iostream>

#include <ctime>

int main() {

srand(time(NULL));

int array[100][100];

int rows;

int cols;

std::cout << "Enter the number of rows: " << std::endl;

std::cin >> rows;

std::cout << "Enter the number of cols: " << std::endl;

std::cin >> cols;

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

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

array[i][j] = 1 + rand() % 9;

}

}

std::cout << "Array:" << std::endl;

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

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

std::cout << array[i][j] << ' ';

}

std::cout << std::endl;

}

std::cout << std::endl;

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

int sum\_row = 0;

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

sum\_row += array[i][j];

}

std::cout << "Sum in row " << i + 1 << ": " << sum\_row << std::endl;

}

std::cout << std::endl;

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

int sum\_col = 0;

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

sum\_col += array[i][j];

}

std::cout << "Sum in col " << j + 1 << ": " << sum\_col << std::endl;

}

return 0;

}

//Задание 5

#include <iostream>

#include <ctime>

#include <string>

#include <locale>

struct Student {

std::string famil;

std::string name;

std::string facult;

int Nomzach;

};

void findStudent(Student \*students, int count, const std::string& famil, const std::string& name) {

bool found = false;

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

if (students[i].famil == famil && students[i].name == name) {

std::cout << "Студент найден:" << std::endl;

std::cout << "Фамилия: " << students[i].famil << std::endl;

std::cout << "Имя: " << students[i].name << std::endl;

std::cout << "Факультет: " << students[i].facult << std::endl;

std::cout << "Номер зачётной книжки: " << students[i].Nomzach << std::endl;

found = true;

}

}

if (!found) {

std::cout << "Студент с указанными параметрами не найден." << std::endl;

}

}

int main() {

setlocale(LC\_ALL, "RUS");

int numStudents = 0;

std::cout << "Введите нужное количество студентов: " << std::endl;

std::cin >> numStudents;

Student \*students = new Student[numStudents];

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

std::cout << "Введите данные для студента " << (i + 1) << ":" << std::endl;

std::cout << "Фамилия: ";

std::cin >> students[i].famil;

std::cout << "Имя: ";

std::cin >> students[i].name;

std::cout << "Факультет: ";

std::cin >> students[i].facult;

std::cout << "Номер зачётной книжки: ";

std::cin >> students[i].Nomzach;

std::cout << std::endl;

}

std::string searchFamil, searchName;

std::cout << "Введите фамилию для поиска: ";

std::cin >> searchFamil;

std::cout << "Введите имя для поиска: ";

std::cin >> searchName;

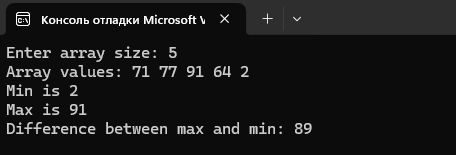
findStudent(students, numStudents, searchFamil, searchName);

return 0;

}

**Результаты работы программы**

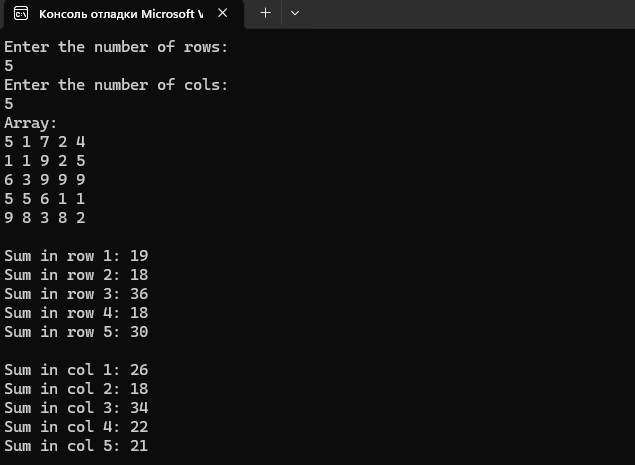
Результаты работы программы представлены на рисунках:



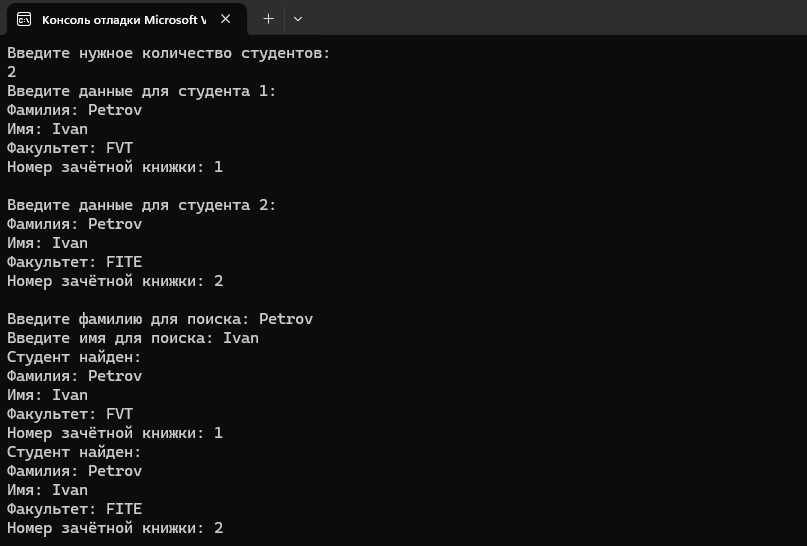
**а)**

Рисунок 1 - Задания 1,2,3

**б)**

Рисунок 2 - Задание 4

**в)**

Рисунок 3 — Задание 5

**Вывод**

В ходе выполнения лабораторной работы была разработана программа с использованием простых структур данных. Результаты работы программы совпали с ожидаемыми результатами, следовательно, программа работает без ошибок.

Был получен опыт в создании проектов в среде Microsoft Visual Studio, приобретен навык программирования алгоритмов.