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.).

**Листинг**

//Задание 1

**main.cpp**

#include "task.h"

int main()

{

system("chcp 1251 > nul");

int mas\_test[9] = { 9, 12, 3, 14, 6, 15, 1, 20, 4 };

//int\* mas\_test = NULL; // Устанавливаем указатель на массив в NULL, чтобы обозначить пустой массив

int number\_of\_elements = sizeof(mas\_test) / sizeof(mas\_test[0]);

//int number\_of\_elements = 0; // Устанавливаем количество элементов в 0

int result = calculate\_difference(mas\_test, number\_of\_elements);

if (result != -1)

printf("Разница между максимальным и минимальным элементами массива: %d\n", result);

else

printf("Массив пустой.\n");

return 0;

}

**[task.h](https://github.com/dimaxaz/LiOAvIZ_lab_1/commit/a59592259be6c87867cd48c9098f34438b74f11b" \l "diff-0a5b0a05598e7c1bea7ccb1ba32f8711747a3caccb1da35732901c8bc7304c84" \o "task.h)**

[#pragma once](https://github.com/dimaxaz/LiOAvIZ_lab_1/commit/a59592259be6c87867cd48c9098f34438b74f11b" \l "diff-3380d17b4fd884967f039ef2a6c310aebffc0b0239ccf7e4472fd39534c0cfc4" \o "calculate_difference.cpp)

[#include <stdio.h>](https://github.com/dimaxaz/LiOAvIZ_lab_1/commit/a59592259be6c87867cd48c9098f34438b74f11b" \l "diff-3380d17b4fd884967f039ef2a6c310aebffc0b0239ccf7e4472fd39534c0cfc4" \o "calculate_difference.cpp)

[#include <stdlib.h>](https://github.com/dimaxaz/LiOAvIZ_lab_1/commit/a59592259be6c87867cd48c9098f34438b74f11b" \l "diff-3380d17b4fd884967f039ef2a6c310aebffc0b0239ccf7e4472fd39534c0cfc4" \o "calculate_difference.cpp)

[int calculate\_difference(int mas[], int size);](https://github.com/dimaxaz/LiOAvIZ_lab_1/commit/a59592259be6c87867cd48c9098f34438b74f11b" \l "diff-3380d17b4fd884967f039ef2a6c310aebffc0b0239ccf7e4472fd39534c0cfc4" \o "calculate_difference.cpp)

**[calculate\_difference.cpp](https://github.com/dimaxaz/LiOAvIZ_lab_1/commit/a59592259be6c87867cd48c9098f34438b74f11b" \l "diff-3380d17b4fd884967f039ef2a6c310aebffc0b0239ccf7e4472fd39534c0cfc4" \o "calculate_difference.cpp)**

#include "task.h"

int calculate\_difference(int mas[], int size)

{

if (size == 0) return -1;

int max\_element = mas[0];

int min\_element = mas[0];

for (int i = 1; i < size; i++)

{

if (mas[i] > max\_element)

max\_element = mas[i];

if (mas[i] < min\_element)

min\_element = mas[i];

}

return max\_element - min\_element;

}

//Задание 2

**[main.cpp](https://github.com/dimaxaz/LiOAvIZ_lab_1/commit/690a5a5221ea00a7b45e3933df09d80513cf82b9" \l "diff-608d8de3fba954c50110b6d7386988f27295de845e9d7174e40095ba5efcf1bb" \o "main.cpp)**

int main()

{

system("chcp 1251 > nul");

int mas\_size;

printf("Введите размер массива: ");

if (scanf("%d", &mas\_size) != 1)

{

fprintf(stderr, "Ошибка ввода размера массива.\n");

return 1;

}

if (mas\_size <= 0)

{

printf("Размер массива должен быть положительным числом.\n");

return 1;

}

int\* mas\_rand = (int\*)malloc(mas\_size \* sizeof(int));

if (mas\_rand == NULL)

{

perror("Ошибка выделения памяти");

return 1;

}

srand(time\_t(NULL));

initialize\_mas(mas\_rand, mas\_size);

printf("Массив, инициализированный случайными числами:\n");

print\_mas\_rand(mas\_rand, mas\_size);

int result = calculate\_difference(mas\_rand, mas\_size);

if (result != -1)

printf("Разница между максимальным и минимальным элементами массива: %d\n", result);

else

printf("Массив пустой.\n");

write\_mas\_to\_file(mas\_rand, mas\_size, FILE\_NAME, result);

printf("Массив успешно записан в файл %s\n", FILE\_NAME);

free(mas\_rand);

return 0;

}

**[task.h](https://github.com/dimaxaz/LiOAvIZ_lab_1/commit/690a5a5221ea00a7b45e3933df09d80513cf82b9" \l "diff-0a5b0a05598e7c1bea7ccb1ba32f8711747a3caccb1da35732901c8bc7304c84" \o "task.h)**

#pragma once

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

#define FILE\_NAME "output.txt"

int calculate\_difference(int mas[], int size);

void initialize\_mas(int\* mas, int size);

void print\_mas\_rand(int\* mas, int size);

void write\_mas\_to\_file(int\* mas, int size, const char\* filename, int result);

**[rand\_mas.cpp](https://github.com/dimaxaz/LiOAvIZ_lab_1/commit/690a5a5221ea00a7b45e3933df09d80513cf82b9" \l "diff-0c4bb9d7582ff94a964944730c6835b8dd114152debbcd04f3e769b2430c5351" \o "rand_mas.cpp)**

#include "task.h"

void initialize\_mas(int\* mas, int size)

{

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

mas[i] = rand() % 100;

}

void print\_mas\_rand(int\* mas, int size)

{

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

printf("%d, ", mas[i]);

printf("\n");

}

**[write\_mas\_to\_file.cpp](https://github.com/dimaxaz/LiOAvIZ_lab_1/commit/690a5a5221ea00a7b45e3933df09d80513cf82b9" \l "diff-88631b77cf4895db8b648027423e5f4f2942065c2dfe6d0d277066ee768b72bd" \o "write_mas_to_file.cpp)**

#include "task.h"

void write\_mas\_to\_file(int\* mas, int size, const char\* filename, int result)

{

FILE\* file = fopen(filename, "w");

if (file == NULL)

{

perror("Îøèáêà îòêðûòèÿ ôàéëà");

return;

}

fprintf(file, "Ìàññèâ: ");

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

fprintf(file, "%d ", mas[i]);

fprintf(file, "\n");

fprintf(file, "Ðàçíèöà ìåæäó ìàêñèìàëüíûì è ìèíèìàëüíûì ýëåìåíòàìè ìàññèâà: %d ", result);

fclose(file);

}

//Задание 3,4

**main.cpp**

#include "task.h"

int main()

{

system("chcp 1251 > nul");

int size\_r = NULL;

int size\_c = NULL;

int choice, choice\_plus;

int result;

if (initialize\_mas\_size(&size\_r,&size\_c) != 0) return 1;

int\*\* mas\_rand = (int\*\*)malloc(size\_r \* sizeof(int\*));

for (int i = 0; i < size\_r; i++)

mas\_rand[i] = (int\*)malloc(size\_c \* sizeof(int));

initialize\_mas\_rand(&mas\_rand, size\_r, size\_c);

printf("Массив, инициализированный случайными числами:\n");

print\_mas\_rand(mas\_rand, size\_r, size\_c);

do

{

printf("Выберите задание для проверки:\n \

1) Вычисляющую разницу между максимальным и минимальным элементами массива\n \

2) Вычисляющую сумму значений в каждом столбце(строке)\n \

3) Заново сгенерировать числа\n \

4) Записать результаты в файл\n \

5) Выйти из программы\n \

Введите ваш выбор: ");

if (scanf("%d", &choice) != 1)

{

printf("Неверный ввод. Попробуйте снова.\n");

while (getchar() != '\n');

continue;

}

switch (choice)

{

case 1:

{

result = calculate\_difference(mas\_rand, size\_r, size\_c);

printf("Разница между максимальным и минимальным элементами массива: %d\n", result);

printf("\n");

} break;

case 2:

{

printf("\n\

1) Сумму значений в каждом столбце\n \

2) Сумму значений в каждой строке\n \

Введите ваш выбор: ");

scanf("%d", &choice\_plus);

switch (choice\_plus)

{

@@ -70,11 +68,6 @@ int main()

calculate\_sum\_columns(mas\_rand, size\_r, size\_c, sum\_column);

} break;

case 2:

@@ -89,10 +82,6 @@ int main()

calculate\_sum\_rows(mas\_rand, size\_r, size\_c, sum\_row);

free(sum\_row);

} break;

@@ -109,10 +98,9 @@ int main()

} break;

case 4:

write\_mas\_to\_file(mas\_rand, size\_r, size\_c, FILE\_NAME, result);

break;

case 5:

printf("Выход из программы...\n");

break;

default:

printf("Неверный выбор. Пожалуйста, попробуйте снова.\n");

}

} while (choice != 5);

for (int i = 0; i < size\_r; i++)

free(mas\_rand[i]);

free(mas\_rand);

return 0;

}

**calculate\_difference.cpp**

#include "task.h"

int calculate\_difference(int\*\* mas, int size\_r, int size\_c)

{

int max\_element = mas[0][0];

int min\_element = mas[0][0];

for (int i = 0; i < size\_r; i++)

for (int j = 0; j < size\_c; j++)

{

if (mas[i][j] > max\_element)

max\_element = mas[i][j];

if (mas[i][j] < min\_element)

min\_element = mas[i][j];

}

return max\_element - min\_element;

}

**calculate\_sum.cpp**

#include "task.h"

int calculate\_sum\_columns(int\*\* mas, int size\_r, int size\_c, int\* sums)

{

for (int j = 0; j < size\_c; j++)

{

sums[j] = 0;

for (int i = 0; i < size\_r; i++)

sums[j] += mas[i][j];

}

printf("Ñóììû çíà÷åíèé â êàæäîì ñòîëáöå:\n");

for (int j = 0; j < size\_c; j++)

printf("Ñóììà ñòîëáöà %d: %d\n", j + 1, sums[j]);

printf("\n");

free(sums);

return 0;

}

int calculate\_sum\_rows(int\*\* mas, int size\_r, int size\_c, int\* sums)

{

sums = (int\*)malloc(size\_r \* sizeof(int));

if (!sums)

{

perror("Îøèáêà âûäåëåíèÿ ïàìÿòè äëÿ ñóìì ñòðîê");

@@ -31,12 +32,13 @@ int calculate\_sum\_rows(int\*\* arr, int size\_r, int size\_c, int\* sums)

{

sums[i] = 0;

for (int j = 0; j < size\_c; j++)

sums[i] += mas[i][j];

}

printf("Ñóììû çíà÷åíèé â êàæäîé ñòðîêå:\n");

for (int i = 0; i < size\_r; i++)

printf("Ñóììà ñòðîêè %d: %d\n", i + 1, sums[i]);

printf("\n");

free(sums);

return 0;

}

**initialize\_mas\_rand.cpp**

#include "task.h"

int initialize\_mas\_rand(int\*\*\* mas, int size\_r, int size\_c)

{

\*mas = (int\*\*)malloc(size\_r \* sizeof(int\*));

if (\*mas == NULL)

{

perror("Îøèáêà âûäåëåíèÿ ïàìÿòè");

return 1;

}

for (int i = 0; i < size\_r; i++)

{

(\*mas)[i] = (int\*)malloc(size\_c \* sizeof(int));

if ((\*mas)[i] == NULL)

{

perror("Îøèáêà âûäåëåíèÿ ïàìÿòè");

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

free((\*mas)[j]);

free(\*mas);

return 1;

}

}

srand((unsigned int)time(NULL));

for (int i = 0; i < size\_r; i++)

for (int j = 0; j < size\_c; j++)

(\*mas)[i][j] = rand() % 100;

return 0;

}

void free\_mas(int\*\* mas, int size\_r)

{

for (int i = 0; i < size\_r; i++)

free(mas[i]);

free(mas);

}

void print\_mas\_rand(int\*\* mas, int size\_r, int size\_c)

{

for (int i = 0; i < size\_r; i++)

{

for (int j = 0; j < size\_c; j++)

printf("%d ", mas[i][j]);

printf("\n");

}

printf("\n");

}

**write\_mas\_to\_file.cpp**

#include "task.h"

void write\_mas\_to\_file(int\*\* arr, int size\_r, int size\_c, const char\* filename, int result)

{

FILE\* file = fopen(filename, "w");

if (!file)

{

perror("Îøèáêà îòêðûòèÿ ôàéëà");

return;

}

fprintf(file, "Ìàññèâ:\n");

for (int i = 0; i < size\_r; i++)

{

for (int j = 0; j < size\_c; j++)

fprintf(file, "%d ", arr[i][j]);

fprintf(file, "\n");

}

fprintf(file, "Ðàçíèöà ìåæäó ìàêñèìàëüíûì è ìèíèìàëüíûì ýëåìåíòàìè: %d\n", result);

fclose(file);

printf("Ðåçóëüòàòû çàïèñàíû â ôàéë '%s'.\n", filename);

printf("\n");

}

//Задание 5

**main.cpp**

#include "student.h"

void hide\_cursor()

{

HANDLE hConsole = GetStdHandle(STD\_OUTPUT\_HANDLE);

CONSOLE\_CURSOR\_INFO cursorInfo;

GetConsoleCursorInfo(hConsole, &cursorInfo);

cursorInfo.bVisible = FALSE;

SetConsoleCursorInfo(hConsole, &cursorInfo);

}

void loading\_animation()

{

const char\* loading\_chars = "|/-\\";

for (int i = 0; i < 25; i++)

{

system("cls");

printf("Загрузка");

for (int j = 0; j < 3; j++)

{

printf(".");

fflush(stdout);

Sleep(30);

}

printf(" %c\n", loading\_chars[i % 4]);

}

system("cls");

printf("Загрузка завершена!\n");

}

int main()

{

system("chcp 1251 > nul");

Student students[100];

int count = 0;

int choice;

int sort\_choice;

hide\_cursor();

loading\_animation();

Sleep(1000);

do

{

system("cls");

printf("\n=======================\n");

printf(" Студенты Меню\n");

printf("=======================\n");

printf("1. Добавить студента\n");

printf("2. Удалить студента\n");

printf("3. Искать студента\n");

printf("4. Показать всех студентов\n");

printf("5. Сортировать студентов\n");

printf("6. Сохранить в файл\n");

printf("7. Загрузить из файла\n");

printf("0. Выход\n");

printf("=======================\n");

printf("Выберите действие: ");

while (scanf("%d", &choice) != 1 || choice < 0 || choice > 7)

{

printf("Ошибка: Пожалуйста, введите число от 0 до 7.\n");

while (getchar() != '\n');

printf("Выберите опцию: ");

}

switch (choice)

{

case 1:

add\_student(students, &count);

break;

case 2:

delete\_student(students, &count);

break;

case 3:

search\_student(students, count);

break;

case 4:

if (count == 0)

printf("Список студентов пуст.\n");

else

display\_students(students, count);

break;

case 5:

printf("Выберите параметр сортировки:\n");

printf("1. Имя\n");

printf("2. Фамилия\n");

printf("3. Год рождения\n");

scanf("%d", &sort\_choice);

if (sort\_choice < 1 || sort\_choice > 3)

printf("Ошибка: Неверный выбор сортировки.\n");

else

{

sort\_students(students, count, sort\_choice);

printf("Студенты успешно отсортированы!\n");

}

break;

case 6:

save\_to\_file(students, count, "students.txt");

printf("Данные успешно сохранены в файл 'students.txt'.\n");

break;

case 7:

load\_from\_file(students, &count, "students.txt");

printf("Данные успешно загружены из файла 'students.txt'.\n");

break;

case 0:

printf("Выход...\n");

break;

default:

printf("Неверный выбор. Попробуйте еще раз.\n");

}

printf("\nНажмите Enter, чтобы продолжить...");

while (getchar() != '\n');

getchar();

} while (choice != 0);

return 0;

}

**students.h**

#ifndef STUDENT\_H

#define STUDENT\_H

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <string.h>

#include <windows.h>

#include <stdio.h>

#define MAX\_STUDENTS 100

typedef struct

{

int id;

char first\_name[50];

char last\_name[50];

int year\_of\_birth;

} Student;

void add\_student(Student\* students, int\* count);

void delete\_student(Student\* students, int\* count);

void search\_student(const Student\* students, int count);

void display\_students(const Student\* const students, int count);

void sort\_students(Student\* students, int count, int choice);

void save\_to\_file(const Student\* students, int count, const char\* filename);

void load\_from\_file(Student\* students, int\* count, const char\* filename);

#endif

**students.cpp**

#include "student.h"

void add\_student(Student\* students, int\* count)

{

Student new\_student;

new\_student.id = \*count + 1;

printf("Введите имя студента: ");

scanf("%s", new\_student.first\_name);

printf("Введите фамилию студента: ");

scanf("%s", new\_student.last\_name);

printf("Введите год рождения студента (число): ");

while (scanf("%d", &new\_student.year\_of\_birth) != 1) {

printf("Неверный ввод. Пожалуйста, введите число для года рождения: ");

while (getchar() != '\n');

}

students[\*count] = new\_student;

(\*count)++;

printf("Студент успешно добавлен!\n");

}

void delete\_student(Student\* students, int\* count)

{

int id;

printf("Введите ID студента для удаления: ");

scanf("%d", &id);

for (int i = 0; i < \*count; i++)

if (students[i].id == id)

for (int j = i; j < \*count - 1; j++)

students[j] = students[j + 1];

(\*count)--; // Уменьшаем количество студентов

printf("Студент с ID %d удален.\n", id);

return;

printf("Студент с ID %d не найден.\n", id);

}

void search\_student(const Student\* students, int count)

{

char name[50];

printf("Введите имя или фамилию для поиска: ");

scanf("%s", name);

int found = 0;

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

if (strcmp(students[i].first\_name, name) == 0 || strcmp(students[i].last\_name, name) == 0)

printf("ID: %d | Имя: %s | Фамилия: %s | Год рождения: %d\n",

students[i].id, students[i].first\_name, students[i].last\_name, students[i].year\_of\_birth);

found = 1;

if (!found) printf("Студенты с именем или фамилией '%s' не найдены.\n", name);

}

void display\_students(const Student\* const students, int count)

{

printf("Список студентов:\n");

printf("\n%-10s %-20s %-20s %-15s\n", "ID", "Имя", "Фамилия", "Год рождения");

printf("----------------------------------------------------------\n");

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

printf(" % -10d % -20s % -20s % -15d\n",

students[i].id, students[i].first\_name, students[i].last\_name, students[i].year\_of\_birth);

printf("----------------------------------------------------------\n");

}

void sort\_students(Student\* students, int count, int choice)

{

for (int i = 0; i < count - 1; i++)

for (int j = 0; j < count - i - 1; j++)

{

int cmp;

if (choice == 1) { // Сортировка по имени

cmp = strcmp(students[j].first\_name, students[j + 1].first\_name);

}

else if (choice == 2) { // Сортировка по фамилии

cmp = strcmp(students[j].last\_name, students[j + 1].last\_name);

}

else if (choice == 3) { // Сортировка по году рождения

cmp = students[j].year\_of\_birth - students[j + 1].year\_of\_birth;

}

// Сортировка по выбранному критерию

if ((choice == 1 || choice == 2) ? cmp > 0 : cmp < 0)

{

Student temp = students[j];

students[j] = students[j + 1];

students[j + 1] = temp;

}

}

}

**File.cpp**

#include "student.h"

void save\_to\_file(const Student\* students, int count, const char\* filename)

{

FILE\* file = fopen(filename, "w");

if (!file)

{

printf("Не удалось открыть файл для записи.\n");

return;

}

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

fprintf(file, "%d %s %s %d\n", students[i].id, students[i].first\_name, students[i].last\_name, students[i].year\_of\_birth);

fclose(file);

printf("Данные сохранены в файл '%s'.\n", filename);

}

void load\_from\_file(Student\* students, int\* count, const char\* filename)

{

FILE\* file = fopen(filename, "r");

if (!file)

{

printf("Не удалось открыть файл для чтения.\n");

return;

}

while (fscanf(file, "%d %s %s %d", &students[\*count].id, students[\*count].first\_name, students[\*count].last\_name, &students[\*count].year\_of\_birth) != EOF)

(\*count)++;

fclose(file);

printf("Данные загружены из файла '%s'.\n", filename);

}

**Результаты работы программы**

Результаты работы программы представлены на рисунках:

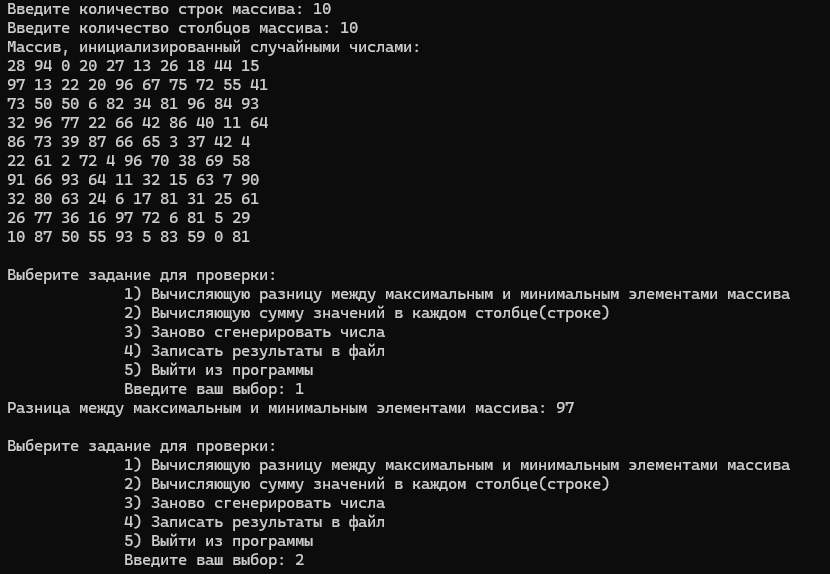


Рисунок 1 - Задания 1,2,3,4

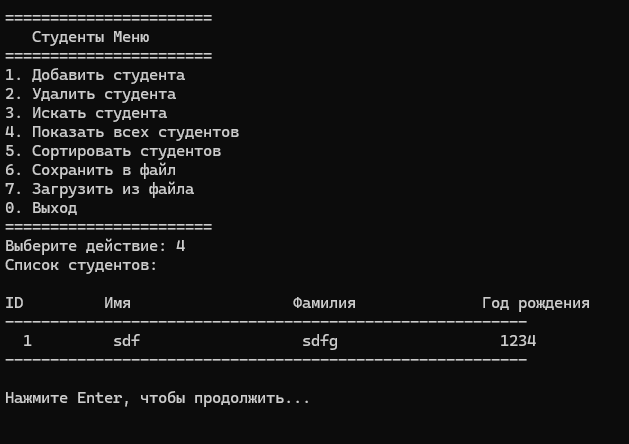
****

Рисунок 2 — 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.