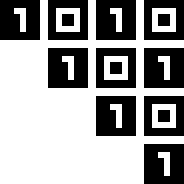
Министерство науки и высшего образования РФ

ФГБОУ ВО «Тамбовский государственный технический университет»

Кафедра

«САПР»



Отчет по лабораторной работе №2

по дисциплине «Алгоритмизация, модели данных и основы ИИ»

Выполнил: студент группы БВТ242

Прохоровский В.В.

Проверил: преподаватель

Патутин К. И.

Тамбов 2025

**Цель работы:**

Освоить основы работы со структурами в языке С++

**Задание**:

Сформировать структуру из ≥4 строковых и ≥3 числовых полей для работы с данными определенной предметной (проблемной) области. Используя данную структуру, сформировать массив данных. Составить функции добавления в массив данных, просмотра массива, записи данных массива в файл и чтения данных в массив из файла. Составить три или более функции расчета или подсчета статистики по данным массива.

**Решение:**

Для решения данной задачи использовалась следующие библиотеки: conio.h, iostream, stdio.h, fstream, структура: cleaner с 7 полями, для работы с файлами в бинарном виде были объявлены объекты класса ifstream для чтения из файла и ofstream для записи в файл, метод close() для закрытия потока, метод write для для записи блока памяти в файл в бинарном режиме, метод read для чтения блока данных из бинарного файла, функция system() для выполнения команд операционной системы(в данном случае «cls», функция \_getch() для чтения одного символа с клавиатуры, функция srand() для инициализации генератора случайных чисел, rand() для генерации псевдослучайных целых чисел, объект cout для вывода информации в консоль, условный операторы if, switch, оператор цикла for, while, оператор % для получения остатка от деления, оператор деления /, оператор сложения +, оператор вычитания -, оператор присваивания += , = , оператор сравнения != , < , >, оператор sizeof() для измерения объема памяти используемой переменной.

**Листинг программы**

#include <conio.h>

#include <iostream>

#include <stdio.h>

#include <fstream>

using namespace std;

struct cleaner

{

int power;

int price;

int noise;

string brend;

string material;

string version;

string form;

};

void fill\_ctr(cleaner& per) {

string BREND[4] = { "ROBOROCK", "DREAME", "XIAOMI","DYSON" };

string MATERIAL[4] = { "PLASTIC", "METAL", "ALUMINUM", "CARBON" };

string VERSION[4] = { "PRO", "LITE", "STANDARD", "ULTIMATE" };

string FORM[2] = { "КРУГ", "КВАДРАТ" };

per.brend = BREND[rand() % 4];

per.material = MATERIAL[rand() % 4];

per.version = VERSION[rand() % 4];

per.form = FORM[rand() % 2];

per.power = rand() % (10000 - 3000) + 3000;

per.noise = rand() % (71 - 20) + 20;

per.price = rand() % (100000 - 10000 + 1) + 10000;

}

void show\_struct(cleaner& per) {

cout << "БРЕНД: " << per.brend << " МАТЕРИАЛ: " << per.material << " Версия: " << per.version << " Форма: " << per.form;

cout << " Мощность: " << per.power << " Цена: " << per.price << " Уровень шума: " << per.noise << endl;

}

int main()

{

const int MENU\_1 = 49;

const int MENU\_2 = 50;

const int MENU\_3 = 51;

const int MENU\_4 = 52;

const int MENU\_5 = 53;

const int MENU\_6 = 54;

const int MENU\_7 = 55;

const int MENU\_8 = 56;

const int MENU\_EXIT = 27;

srand(time(0));

setlocale(LC\_ALL, "rus");

const int n = 15;

cleaner robot[n];

int req = 0;

while (req != MENU\_EXIT) {

system("cls");

cout << " 1. Заполнить базу случайными данными.\n";

cout << " 2. Сохранить базу данных в файл.\n";

cout << " 3. Загрузить базу данных из файла.\n";

cout << " 4. Показать базу данных.\n";

cout << " 5. Посчитать кол-во пылесосов каждого бренда.\n";

cout << " 6. Определить модели с ценой, меньшей чем n руб.\n";

cout << " 7. Найти модель с мощностью, большей чем среднее арифметическое мощностей и по самой низкой цене.\n";

cout << " ESC. Выход.\n";

req = \_getch();

switch (req)

{

case MENU\_1: {

system("cls");

for (int i = 0; i < n; i++) { fill\_ctr(robot[i]); }

cout << "Заполнил случайно";

\_getch();

break;

}

case MENU\_2: {

system("cls");

ofstream file("text.t33", ios::binary);

file.write((char\*)robot, sizeof(robot));

file.close();

cout << "ЗАПИСАЛ В ФАЙЛ";

\_getch();

break;

}

case MENU\_3: {

system("cls");

ifstream file("text.t33", ios::binary);

file.read((char\*)robot, sizeof(robot));

file.close();

cout << "ПРОЧИТАЛ ИЗ ФАЙЛ";

\_getch();

break;

}

case MENU\_4: {

system("cls");

for (int i = 0; i < n; i++) { show\_struct(robot[i]); }

\_getch();

break;

}

case MENU\_5: {

system("cls");

int count\_roborock = 0, count\_dreame = 0, count\_xiaomi = 0, count\_dyson = 0;

cout << "Посчитать кол-во пылесосов каждого бренда\n";

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

if (robot[i].brend == "ROBOROCK") count\_roborock++;

else if (robot[i].brend == "DREAME") count\_dreame++;

else if (robot[i].brend == "XIAOMI") count\_xiaomi++;

else if (robot[i].brend == "DYSON") count\_dyson++;

}

cout << "СТАТИСТИКА ПО БРЕНДАМ:\n";

cout << "ROBOROCK: " << count\_roborock << " шт.\n";

cout << "DREAME: " << count\_dreame << " шт.\n";

cout << "XIAOMI: " << count\_xiaomi << " шт.\n";

cout << "DYSON: " << count\_dyson << " шт.\n";

\_getch();

break;

}

case MENU\_6: {

system("cls");

int price = 0;

cout << "Определить модели с ценой, меньшей чем n руб.\n";

cout << "Введите число n: ";

cin >> price;

cout << "Модели, дешевле назначенной цены: " << endl;

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

{

if (robot[i].price < price)

{

show\_struct(robot[i]);

}

}

\_getch();

break;

}

case MENU\_7: {

system("cls");

int count = 0, sum = 0;

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

{

sum += robot[i].power;

count++;

}

if (count == 0 or sum <= 0) {

cout << "ERROR";

\_getch();

break;

}

float sr\_arif = (float)sum / count;

int min\_price = 110000;

cleaner pust;

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

{

if (robot[i].power > sr\_arif and robot[i].price < min\_price)

{

min\_price = robot[i].price;

pust = robot[i];

}

}

cout << "Найти модель с мощностью, большей чем среднее арифметическое мощностей и по самой низкой цене" << endl;

cout << "НАЙДЕННАЯ МОДЕЛЬ: " << endl;

show\_struct(pust);

\_getch();

break;

}

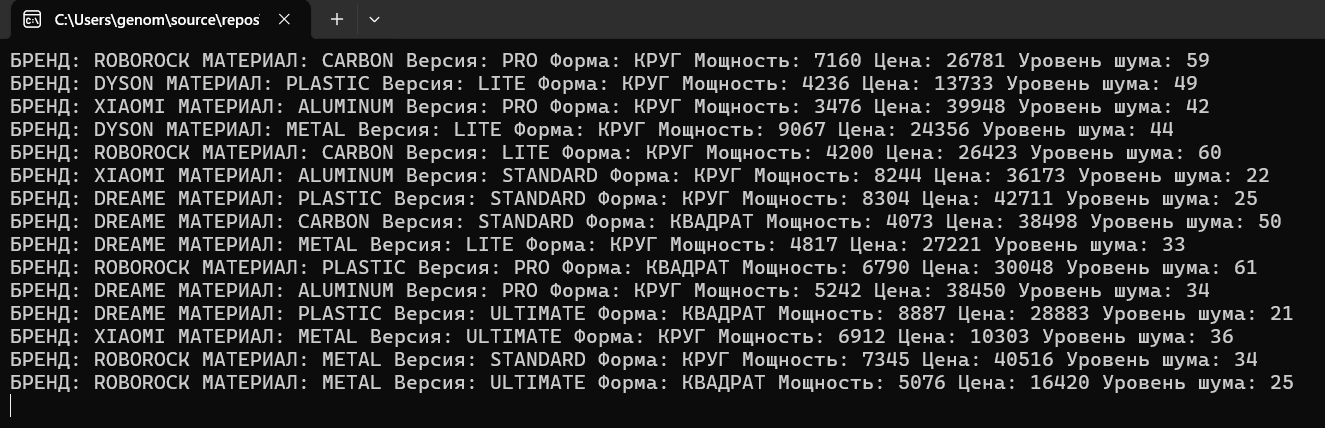
}

}

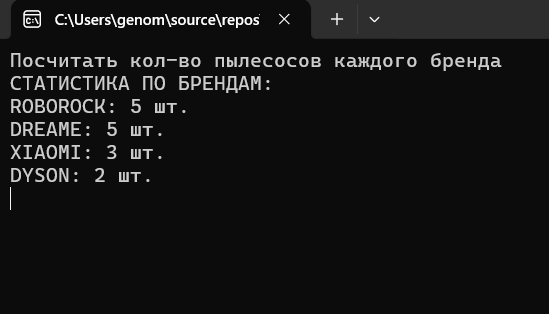
}

**Результаты работы программы**

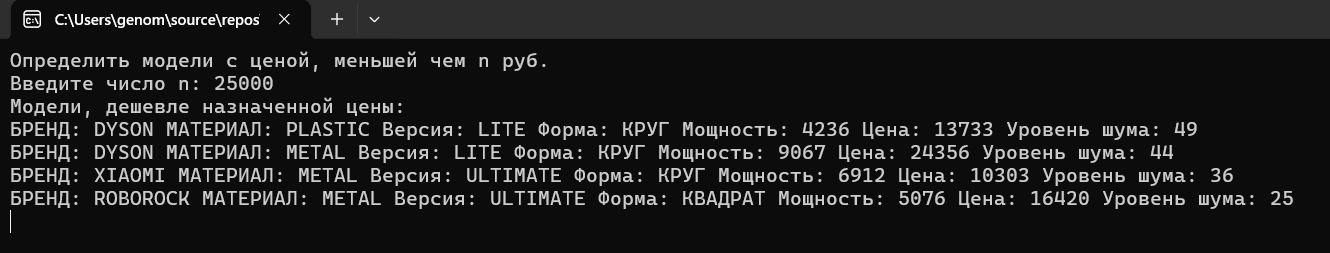
Пример 1



Пример 2



Пример 3



Пример 4

