



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.

Цель работы:

Освоить основы работы со структурами в языке C++

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 использовались 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 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, условные 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

```

C:\Users\genom\source\repos
БРЕНД: ROBOROCK МАТЕРИАЛ: CARBON Версия: PRO Форма: КРУГ Мощность: 7160 Цена: 26781 Уровень шума: 59
БРЕНД: DYSON МАТЕРИАЛ: PLASTIC Версия: LITE Форма: КРУГ Мощность: 4236 Цена: 13733 Уровень шума: 49
БРЕНД: XIAOMI МАТЕРИАЛ: ALUMINUM Версия: PRO Форма: КРУГ Мощность: 3476 Цена: 39948 Уровень шума: 42
БРЕНД: DYSON МАТЕРИАЛ: METAL Версия: LITE Форма: КРУГ Мощность: 9067 Цена: 24356 Уровень шума: 44
БРЕНД: ROBOROCK МАТЕРИАЛ: CARBON Версия: LITE Форма: КРУГ Мощность: 4200 Цена: 26423 Уровень шума: 60
БРЕНД: XIAOMI МАТЕРИАЛ: ALUMINUM Версия: STANDARD Форма: КРУГ Мощность: 8244 Цена: 36173 Уровень шума: 22
БРЕНД: DREAME МАТЕРИАЛ: PLASTIC Версия: STANDARD Форма: КРУГ Мощность: 8304 Цена: 42711 Уровень шума: 25
БРЕНД: DREAME МАТЕРИАЛ: CARBON Версия: STANDARD Форма: КВАДРАТ Мощность: 4073 Цена: 38498 Уровень шума: 50
БРЕНД: DREAME МАТЕРИАЛ: METAL Версия: LITE Форма: КРУГ Мощность: 4817 Цена: 27221 Уровень шума: 33
БРЕНД: ROBOROCK МАТЕРИАЛ: PLASTIC Версия: PRO Форма: КВАДРАТ Мощность: 6790 Цена: 30048 Уровень шума: 61
БРЕНД: DREAME МАТЕРИАЛ: ALUMINUM Версия: PRO Форма: КРУГ Мощность: 5242 Цена: 38450 Уровень шума: 34
БРЕНД: DREAME МАТЕРИАЛ: PLASTIC Версия: ULTIMATE Форма: КВАДРАТ Мощность: 8887 Цена: 28883 Уровень шума: 21
БРЕНД: XIAOMI МАТЕРИАЛ: METAL Версия: ULTIMATE Форма: КРУГ Мощность: 6912 Цена: 10303 Уровень шума: 36
БРЕНД: ROBOROCK МАТЕРИАЛ: METAL Версия: STANDARD Форма: КРУГ Мощность: 7345 Цена: 40516 Уровень шума: 34
БРЕНД: ROBOROCK МАТЕРИАЛ: METAL Версия: ULTIMATE Форма: КВАДРАТ Мощность: 5076 Цена: 16420 Уровень шума: 25

```

Пример 2

```

C:\Users\genom\source\repos
Посчитать кол-во пылесосов каждого бренда
СТАТИСТИКА ПО БРЕНДАМ:
ROBOROCK: 5 шт.
DREAME: 5 шт.
XIAOMI: 3 шт.
DYSON: 2 шт.

```

Пример 3

```

C:\Users\genom\source\repos
Определить модели с ценой, меньшей чем n руб.
Введите число n: 25000
Модели, дешевле назначенной цены:
БРЕНД: DYSON МАТЕРИАЛ: PLASTIC Версия: LITE Форма: КРУГ Мощность: 4236 Цена: 13733 Уровень шума: 49
БРЕНД: DYSON МАТЕРИАЛ: METAL Версия: LITE Форма: КРУГ Мощность: 9067 Цена: 24356 Уровень шума: 44
БРЕНД: XIAOMI МАТЕРИАЛ: METAL Версия: ULTIMATE Форма: КРУГ Мощность: 6912 Цена: 10303 Уровень шума: 36
БРЕНД: ROBOROCK МАТЕРИАЛ: METAL Версия: ULTIMATE Форма: КВАДРАТ Мощность: 5076 Цена: 16420 Уровень шума: 25

```


Пример 4

```
C:\Users\genom\source\repos  ×  +  ▾  
Найти модель с мощностью, большей чем среднее арифметическое мощностей и по самой низкой цене  
НАЙДЕННАЯ МОДЕЛЬ:  
БРЕНД: ХІАОМІ МАТЕРІАЛ: МЕТАL Версія: ULTIMATE Форма: КРУГ Мощность: 6912 Цена: 10303 Уровень шума: 36  
|
```