МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное автономное образовательное учреждение высшего образования

«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ   
АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ»

КАФЕДРА ИНФОРМАЦИОННО-СЕТЕВЫХ ТЕХНОЛОГИЙ

КУРСОВАЯ РАБОТА  
ЗАЩИЩЕНА С ОЦЕНКОЙ

РУКОВОДИТЕЛЬ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| доц., канд. техн. наук |  |  |  | Курицын К.А. |
| должность, уч. степень, звание |  | подпись, дата |  | инициалы, фамилия |

|  |
| --- |
| ПОЯСНИТЕЛЬНАЯ ЗАПИСКА К КУРСОВОЙ РАБОТЕ |
| КОНСОЛЬНОЕ ПРИЛОЖЕНИЕ «МАРКИ» |
| по дисциплине: ТЕХНОЛОГИЯ ПРОГРАММИРОВАНИЯ |
|  |
|  |

РАБОТУ ВЫПОЛНИЛ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| СТУДЕНТ ГР.№ | 1542 |  |  |  | Абросимов С.С. |
|  |  |  | подпись, дата |  | инициалы, фамилия |

Санкт-Петербург 2018

Содержание

1. Задание ……………………………………………………………………….…...2
2. Функциональные требования……………………………………………….…...2
3. Требование к оборудованию и стороннему ПО…………………………….….2
4. Технические спецификации……………………………………………………..2
5. Взаимодействие с внешними и внутренними системами……………………..3
6. Программа и методика испытаний……………………………………………...3
7. Литература………………………………………………………………………..4

Приложение 1 – Описание формата………………………………………………...5

Приложение 2 – Листинг………………………………………………………….....6

Приложение 3 – Примеры, скриншоты…………………………………………….12

1. **Задание**

Создать сборник марок, который может включать в себя марки реальных государств и виртуальных государств. Пользователь может выбирать ,какие марки необходимо добавить в сборник (характеристики марок: к какому государству относятся, назначение, по характеру эмитента, форма, материал), и в каком количестве. Использовать паттерн “Фабричный метод”. Результат создания сборника марок и их характеристики записываются в выходной файл, разделенный по типам марок – реальное государство или нет. Пользователь может загрузить сборник марок из файла.

1. **Функциональные требования**

2.1) Выход из приложения

Реализовать выход из приложения.

2.2) Добавление марок

Реализовать добавление новых марок в файлы-каталоги. Добавление должно происходить либо из консоли, либо из файла.

2.3) Вывод на консоли марок.

Реализовать вывод на консоль всех марок, либо созданных пользователем, либо из файла.

2.4) Запись результата

Определяется тип марки и по данному типу результат записывается в txt файл формата указанного в Приложении 1. Марки записываются в два выходных файла-каталога. Файлы должны быть доступны для просмотра в любое время.

2.5) Удобный интерфейс

Интерфейс должен быть удобным и понятным. Необходимо очищать консоль от лишних элементов.

2.6) Язык С++

Приложение должно быть реализовано при помощи языка С++ и использовании классов.

1. **Требования к оборудованию и стороннему ПО**

Консольное приложение. Операционная система Windows 7/8/10 и Visual Studio 2013/2015/2017.

1. **Технические спецификации**

4.1) При запуске приложения перед пользователем отображается название приложения и варианты действий для работы с марок.

4.2) Пункт “q” отвечает за выход из приложения.

4.3) Пункт “a” отвечает за добавление новой марки.

Если выбран этот вариант, то пользователю предлагается выбрать, как он будет добавлять характеристики марке: из консоли или из файла. После чего марка автоматически добавляется в каталог. Затем пользователь вновь попадает в основное меню.

4.3.1)Если пользователь хочет добавить характеристики из файла, он вводит “file”. После пользователь вводит тип марки. Если пользователь ввел неверный тип, на консоль выведется сообщение и пользователь сможет ввести тип заново. Затем выводится количество характеристик этой марки в файле, и пользователь выбирает одну из них.

4.3.2)Если пользователь хочет добавить характеристики из файла, он вводит “console”.После пользователь вводит тип марки. Если пользователь ввел неверный тип, на консоль выведется сообщение и пользователь сможет ввести тип заново. Затем вручную вводит все характеристики марки.

4.4) Пункт “p” отвечает за вывод на консоль всех марок.

Если выбран этот вариант, то пользователю предлагается выбрать, какие марки вывести, те, что есть в файле, или те, которые пользователь создал. Затем пользователь вновь попадает в основное меню.

4.4.1) Если пользователь хочет вывести марки, которые есть в файле, он вводит “file” и нажимает Enter. После чего вводит тип марки. Если пользователь ввел неверный тип, на консоль выведется сообщение и пользователь сможет ввести тип заново. Затем на консоли выводятся все характеристики этого типа из файла.

4.4.2)Если пользователь хочет вывести марки, которые создал, он вводит “console” и нажимает Enter. После чего вводит тип марки. Если пользователь ввел неверный тип, на консоль выведется сообщение и пользователь сможет ввести тип заново. Затем на консоли выводятся все характеристики этого типа.

4.5) Пункт “s” отвечает за сохранение результата в выходные файлы.

При выборе этого пункта, автоматически все созданные пользователем марки сохраняются в выходные файлы, которые определяются по типу вводимой информации пользователем и на консоль выводится сообщение, что сохранение прошло успешно. После чего пользователь вновь попадает в основное меню.

1. **Взаимодействия с внешними и внутренними системами**

Взаимодействие типа консоль-файл. Приложение взаимодействует с тремя файлами:

Work, Add, Katalog.

1. **Программа и методика испытаний**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Сценарий проверки | Ожидаемый результат | ФТ | Тех. Спец. | Результат |
| 1 | Включение приложения | Отображение названия программы и всех пунктов для работы с файлом каталога:  1)Добавление марок  2)Вывод марок  3)Сохранение результата  4)Выход | 2.5 | 4.1 | Готово  Рисунок1. |
| 2 | В меню выбрать q | Выход из приложения | 2.1 | 4.2 | Готово |
| 3 | В меню выбрать a | Добавление марок | 2.2 | 4.3 | Готово |
|  |  | Добавление марок из файла | 2.2 | 4.3.1 | Готово  Рисуоно 2. |
|  |  | Введен неверный тип марки | - | 4.3.1 | Готово  Рисунок 7. |
|  |  | Выбрана характеристика из файла | - | 4.3.1 | Готово  Рисунок 2. |
|  |  | Добавление марки с помощью консоли | 2.2 | 4.3.2 | Готово  Рисунок 3. |
|  |  | Введен неверный тип марки | - | 4.3.2 | Готово  Рисунок 8. |
| 4 | В меню выбрать p | Вывод на консоль | 2.3 | 4.4 | Готово |
|  |  | Вывод на консоль марок из файлов | 02.3 | 4.4.1 | Готово  Рисунок 5.  Рисунок 9 |
|  |  | Вывод на консоль созданных пользователем марок | 2.3 | 4.4.2 | Готово  Рисунок 4. |
| 5 | В меню выбрать s | Сохранение результата | 2.4 | 4.5 | Готово  Рисунок 6.  Рисунок 10 |
| 6 | Возможность просмотра файла каталога, рабочего файла | Отображение актуальной информации | 2.5 | - | Готово |
| 7 | Возможность изменение файла добавления марок и использование его в процессе работы с консолью | Добавление не приводящее к ошибкам | - |  | Готово |

1. **Литература**

* https://support.microsoft.com
* http://www.cyberforum.ru
* https://ru.stackoverflow.com
* http://cppstudio.com
* http://www.c-cpp.ru
* Б. Керниган и Д. Ритчи – Язык программирования Си
* Шилдт Г. Самоучитель С++
* Бьерн Страуструп - Язык программирования С++

**Приложение №1 Описание формата**

Формат файла должен отвечать следующим требованиям:

1. Разделение марок по типу

В начале каждой марки выводится его тип. Например(“real\_state” , “virtuall\_state”)

1. Информация о марке

Информация о каждом марке должна быть представлена в виде набора данных разделённого построчно. Каждая строка отвечает за отдельный тип информации.

1. Порядок представления информации

Информация о марке должна быть представлена в следующем порядке (сверху вниз, построчно):

К какому государству относится, назначение, по характеру эмитента, форма, материал.

Все строчные переменные должны быть представлены в виде одного слова. Для разделения слов использовать символ: «\_».

4) Разделение марок между собой:   
  
Записи не имеют специальных разделителей. На очередную марку отводится 6 строк файла.

5) Информация о марке:   
  
Информация о каждой марке должна быть представлена в виде набора данных разделённого построчно. Каждая строка отвечает за отдельный тип информации.

6) Входной файл должны содержать данные всех марок.   
  
Формат входного файла должен отвечать следующим требованиям:   
Начало и конец описания одной марки:   
Описание марки начинается с ключевых слов на отдельной строке: разделенных по типу марок (real\_state , virtuall\_state)  
Описание каждой марки заканчивается специальным символом на отдельной строке:   
";"

**Приложение №2 Листинг**

Ссылка на Github https://github.com/abrosimovss/KursovickStamps

**Листинг файла factory.h**

#include <list>

#include <string>

#include <iostream>

#include "marki.h"

using namespace std;

string word\_in;

string mas[8];

class factory

{

public:

~factory();

list <Mark\*> list;

Mark \* Add(string name, istream &is);

Mark \* Add\_file(string name, string id, fstream &ofs);

void Print(string type, ostream &os);

void Save(ostream &fr, ostream &fv,string type);

};

factory::~factory()

{

for (Mark \*ptr : list)

delete ptr;

}

void factory::Save(ostream &fr, ostream &fv, string type)

for (Mark \*ptr : list)

{

if (ptr->Type().name == "real\_state") {

fr << ptr->Type().name << endl;

fr << (\*ptr) << endl;

}

if (ptr->Type().name == "virtuall\_state") {

fv << ptr->Type().name << endl;

fv << (\*ptr) << endl;

}

}

}

Mark \* factory::Add\_file(string typ, string id, fstream &ofs)

{

int i = 0;

for (MarkType \*type : get\_types())

if (type->name == typ)

{

Mark \*ptr = type->make();

while (ofs >> word\_in)

{

if (word\_in == typ)

{

ofs >> word\_in;

if (word\_in == id) {

while (word\_in != ";")

{

mas[i] = word\_in;

ofs >> word\_in;

i++;

}

list.push\_back(ptr);

ptr->set(mas, ptr);

ptr->Print2(ofs);

return ptr;

}

else {

while (word\_in != ";")

ofs >> word\_in;

}

}

else {

while (word\_in != ";")

ofs >> word\_in;

}

}

}

}

Mark \* factory::Add(string name, istream &is)

{

for (MarkType \*type : get\_types())

if (type->name == name)

{

Mark \*ptr = type->make();

list.push\_back(ptr);

is >> (\*ptr);

return ptr;

}

}

void factory::Print(string type, ostream &os)

{

bool flag = (type == "");

for (Mark \*ptr : list)

if (flag || ptr->Type().name == type)

{

os << endl << (\*ptr);

}

}

**Листинг файла marki.h**

#include <list>

#include <string>

#include <iostream>

#include <sstream>

#include <iomanip>

using namespace std;

void clrFile(ostream &fname) {

fname.clear();

}

class Mark;

struct MarkType;

template <class T>

Mark \* MarkMaker() { return new T; }

list <MarkType\*> & get\_types()

{

static list <MarkType\*> types;

return types;

}

struct MarkType

{

int count;

string name;

Mark \*(\*make)();

MarkType(const char \*n, Mark \*(\*m)())

: name(n), make(m), count(0) {

get\_types().push\_back(this);

}

};

class Mark {

protected:

string appointment;

string issuer;

string form;

string material;

int number\_of;

public:

virtual const MarkType & Type() const = 0;

virtual void Print(ostream &os = cout) const;

virtual void Print2(fstream &os) const;

virtual void Read(istream &is);

virtual void Save(ostream &os);

virtual void set(string \*mas, Mark \*d);

virtual ~Mark() {}

};

void Mark::Save(ostream &os)

{

os << appointment << endl;

os << issuer << endl;

os << form << endl;

os << material << endl;

os << number\_of << endl;

}

void Mark::Read(istream &is)

{

cout << "appointment: "; cin >> appointment;

cout << "issuer: "; cin >> issuer;

cout << "form: "; cin >> form;

cout << "material: "; cin >> material;

cout << "number\_of: "; cin >> number\_of;

}

void Mark::set(string \*mas, Mark \*d)

{

d->appointment = mas[1];

d->issuer = mas[2];

d->form = mas[3];

d->material = mas[4];

std::istringstream ist2(mas[5]);

ist2 >> d->number\_of;

};

void Mark::Print(ostream &os) const

{

os << "appointment: " << appointment << endl;

os << "by the nature of the issuer : " << issuer << endl;

os << "form: " << form << endl;

os << "material: " << material << endl;

os << "number of: " << number\_of << endl;

};

void Mark::Print2(fstream &os) const

{

cout << "appointment: " << appointment << endl;

cout << "issuer: " << issuer << endl;

cout << "form: " << form << endl;

cout << "material: " << material << endl;

cout << "number\_of: " << number\_of << endl;

}

class real\_state :public Mark {

static MarkType type;

public:

real\_state() { ++type.count; }

void Print(ostream &os);

void Print2(fstream &os);

const MarkType & Type() const { return type; }

};

class virtuall\_state :public Mark {

static MarkType type;

public:

virtuall\_state() { ++type.count; }

const MarkType & Type() const { return type; }

void Print(ostream &os);

void Print2(fstream &os);

};

void real\_state::Print(ostream &os) { Mark::Print(os); }

void real\_state::Print2(fstream &os) { Mark::Print2(os); }

void virtuall\_state::Print(ostream &os) { Mark::Print(os); }

void virtuall\_state::Print2(fstream &os) { Mark::Print2(os); }

MarkType real\_state::type("real\_state", &MarkMaker<real\_state>);

MarkType virtuall\_state::type("virtuall\_state", &MarkMaker<virtuall\_state>);

istream & operator >> (istream &is, Mark &d)

{

d.Read(is); return is;

}

ostream & operator<<(ostream &os, const Mark &d)

{

d.Print(os); return os;

}

**Листинг файла main.cpp**

[#include](https://vk.com/im?sel=92311422&st=%23include) <iostream>   
[#include](https://vk.com/im?sel=92311422&st=%23include) <string>   
[#include](https://vk.com/im?sel=92311422&st=%23include) <fstream>   
[#include](https://vk.com/im?sel=92311422&st=%23include) <conio.h>   
[#include](https://vk.com/im?sel=92311422&st=%23include) "factory.h"   
using namespace std;   
  
int main()   
{   
setlocale(LC\_ALL, "RUSSIAN");   
string word\_in, file\_in, file\_in2;   
fstream filestr("C:\\Users\\Сергей\\Desktop\\ GUAP.LABS\\KursovickStamps \\doc\\tree.txt", fstream::in | fstream::out);   
file\_in2 = "(" C:\\Users\\Сергей\\Desktop\\ GUAP.LABS\\ KursovickStamps \\doc \\tree.txt";   
ofstream fout("("C:\\Users\\Сергей\\Desktop\\ GUAP.LABS\\ KursovickStamps \\doc \\two.txt");   
ofstream fout2("("C:\\Users\\Сергей\\Desktop\\ GUAP.LABS\\ KursovickStamps \\doc \\one.txt");   
  
factory list;   
string type;   
string otobraj;   
char choice = 0;   
do   
{   
system("cls");   
if (choice == 'a' || choice == 'A')   
{   
cout « "input characteristics from a file or console (file, concole)"; cin » otobraj;   
cout « "Input mark type (real\_state or virtuall\_state ) ->";   
int kol = 0;   
string type;   
metka1: cin » type;   
if (type != "real\_state" && type != "virtuall\_state" ) { cout « "entered an incorrect type! Try again :" « endl; goto metka1; }   
else   
{   
if (otobraj == "File" || otobraj == "file" || otobraj == "f")   
{   
fstream filestr("("C:\\Users\\Сергей\\Desktop\\ GUAP.LABS\\ KursovickStamps \\doc \\tree.txt", fstream::in | fstream::out);   
while (filestr » word\_in)   
{   
if (word\_in == type)   
{   
while (word\_in != ";") { filestr » word\_in; }   
kol++;   
}   
else { while (word\_in != ";")filestr » word\_in; }   
}   
filestr.close();   
fstream filestr1("("C C:\\Users\\Сергей\\Desktop\\ GUAP.LABS\\ KursovickStamps \\doc \\tree.txt", fstream::in | fstream::out);   
string id;   
cout « "number of characteristics = " « kol « endl « "Choose charachteristic ->";   
cin » id;   
list.Add\_file(type, id, filestr1);   
filestr1.close();   
}   
else { list.Add(type, cin); }   
}   
}   
  
else if (choice == 'p' || choice == 'P')   
{   
cout « "Example from file or console (file, console)"; cin » otobraj;   
cout « "Input mark type (real\_state or virtuall\_state ) ->";   
metka2: cin » type;   
if (type != "real\_state" && type != "virtuall\_state") { cout « "entered an incorrect type! Try again :" « endl; goto metka2; }   
else   
{   
if (otobraj == "File" || otobraj == "file" || otobraj == "f")   
{   
fstream filestr("("C:\\Users\\Сергей\\Desktop\\ GUAP.LABS\\ KursovickStamps \\doc\\tree.txt", fstream::in | fstream::out);   
while (filestr » word\_in)   
{   
if (word\_in == type)   
{   
while (word\_in != ";") {   
cout « word\_in « endl;   
filestr » word\_in;   
}   
}   
}   
filestr.close();   
}   
else { list.Print(type, cout); }   
}   
}   
  
else if (choice == 's' || choice == 'S')   
{   
list.Save(fout,fout2,type);   
cout « "saved " « endl;   
}   
cout « endl «   
"A - add new element to list" « endl   
« "P - print list of marki" « endl   
« "S - save results into file" « endl   
« "Q - quit" « endl;   
  
} while ((choice =\_getch()) != 'q' && choice != 'Q');   
fout.close();   
}

**Приложение №3 Примеры**

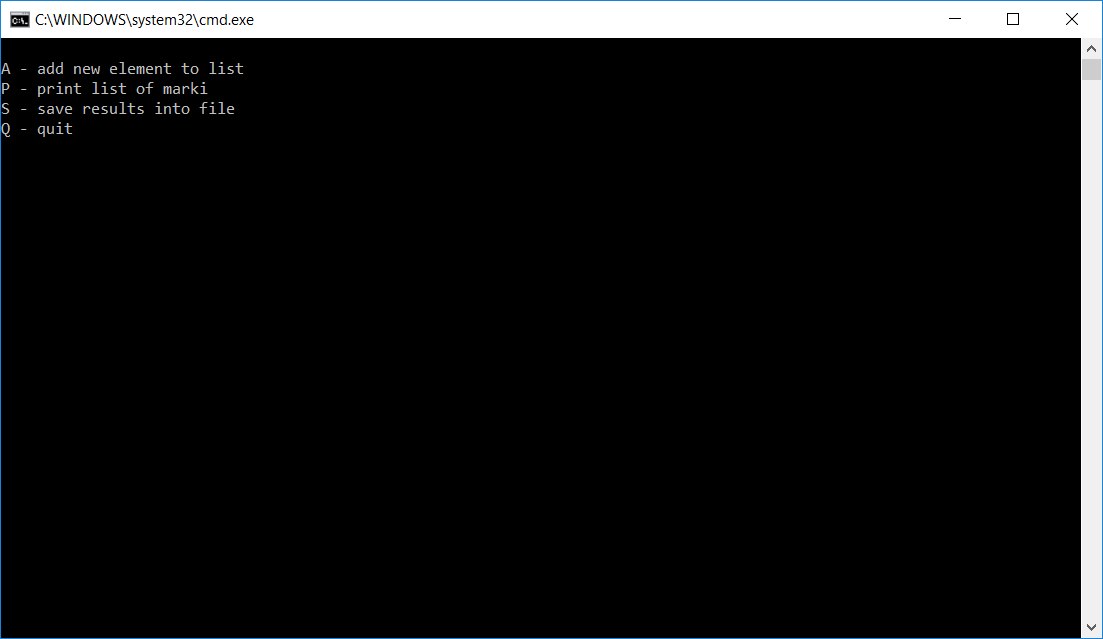


Рисунок 1. Главное меню.

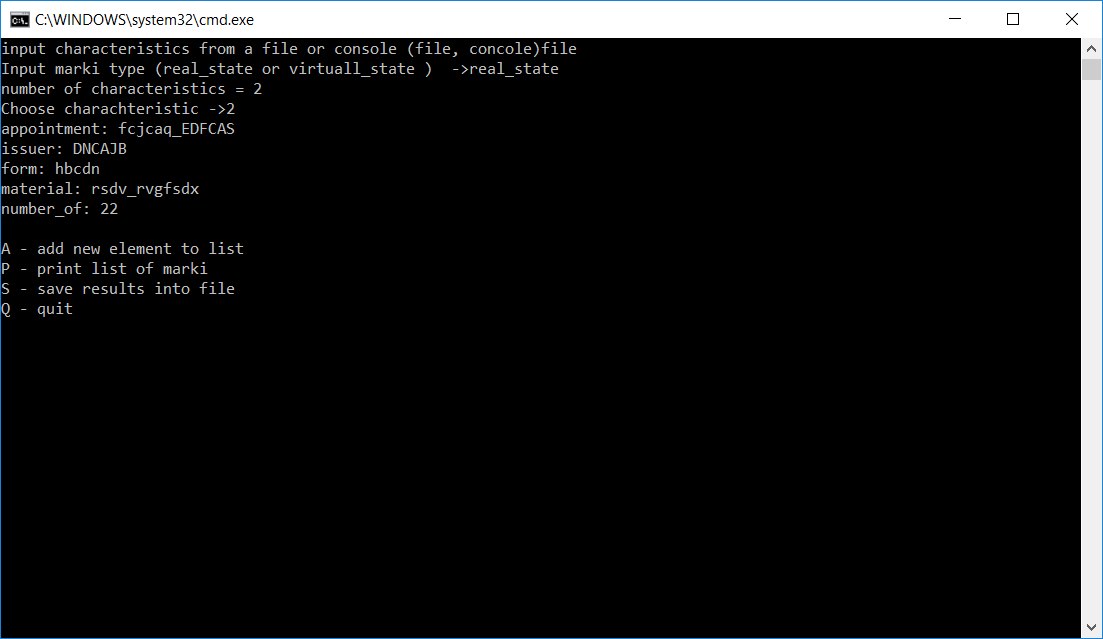
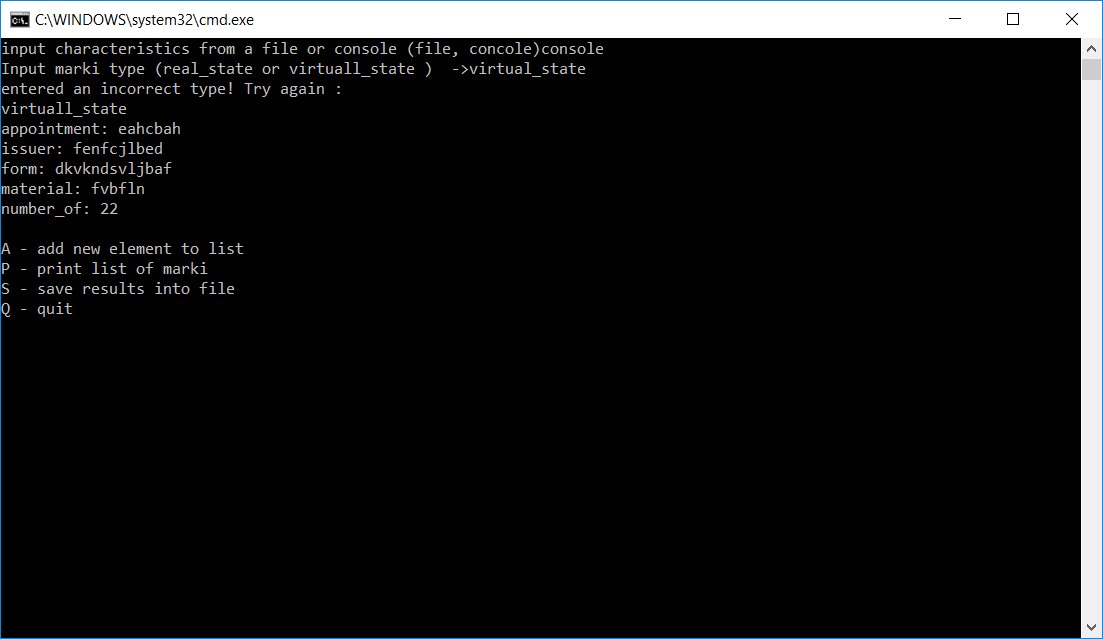


Рисунок 2. Добавление марок (характеристики из файла).



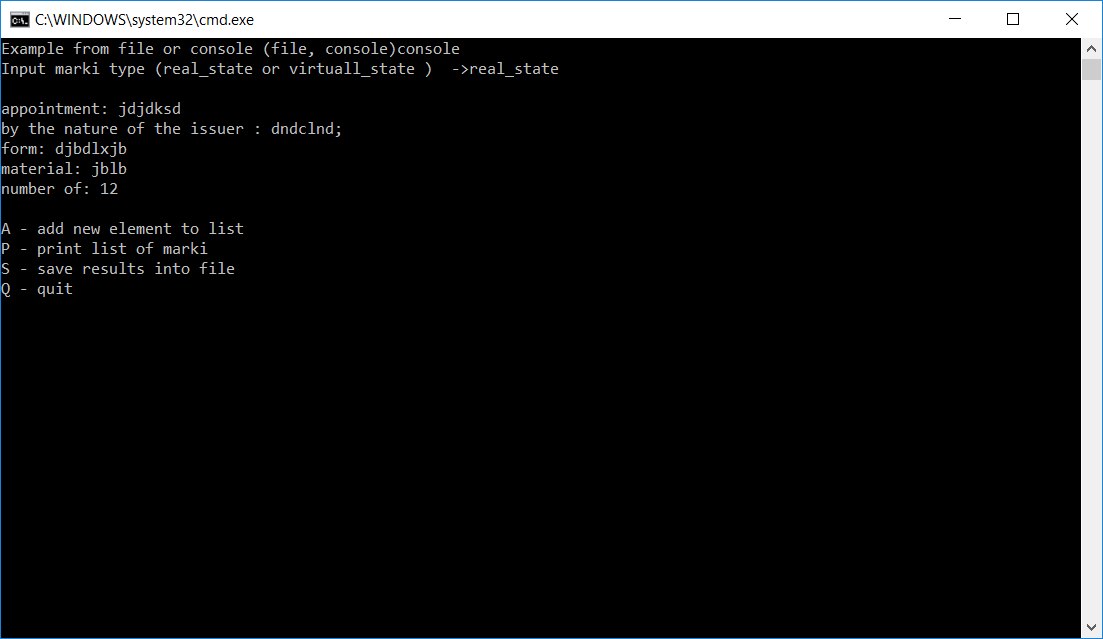


Рисунок 3. Добавление марки (характеристики из консоли).

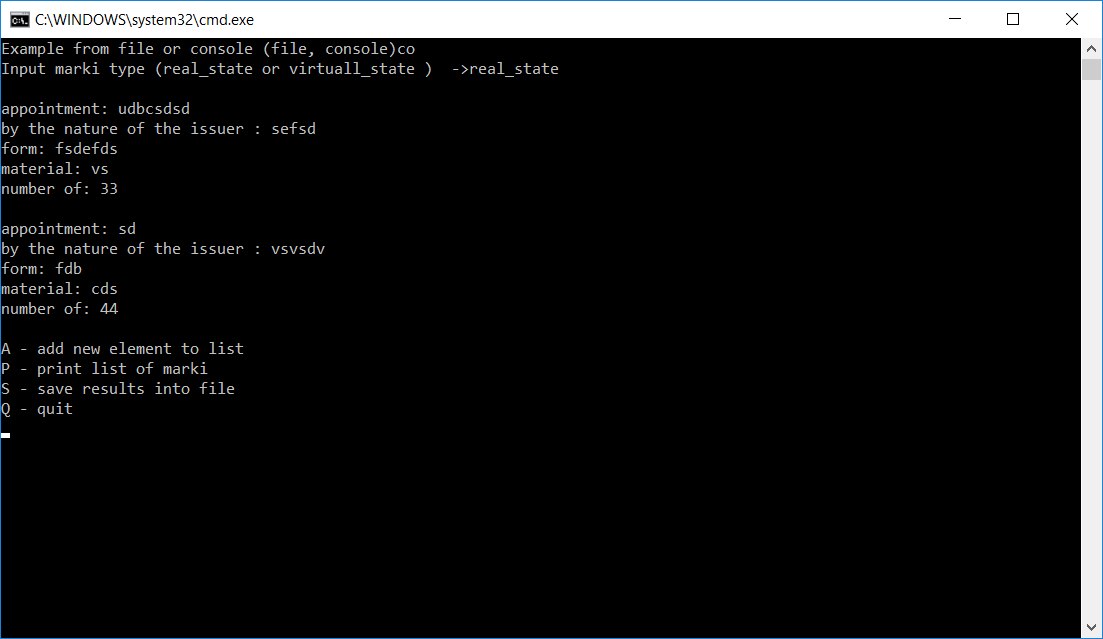


Рисунок 4. Вывод марок созданных пользователем

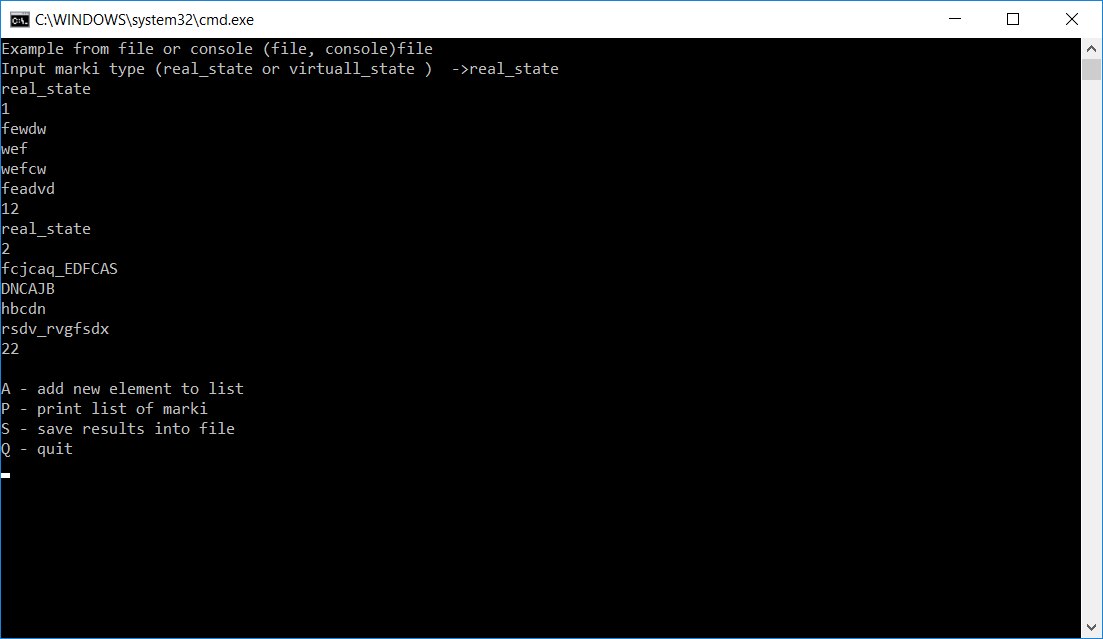


Рисунок 5. Вывод марок из файла.

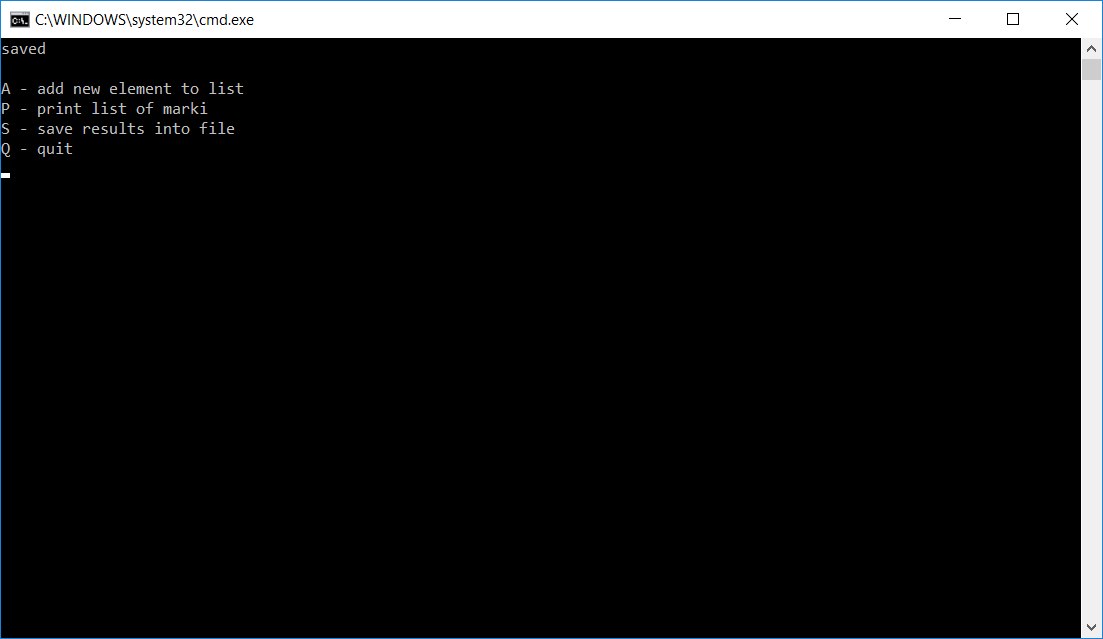


Рисунок 6. Сообщение при сохранение марки в выходной файл.

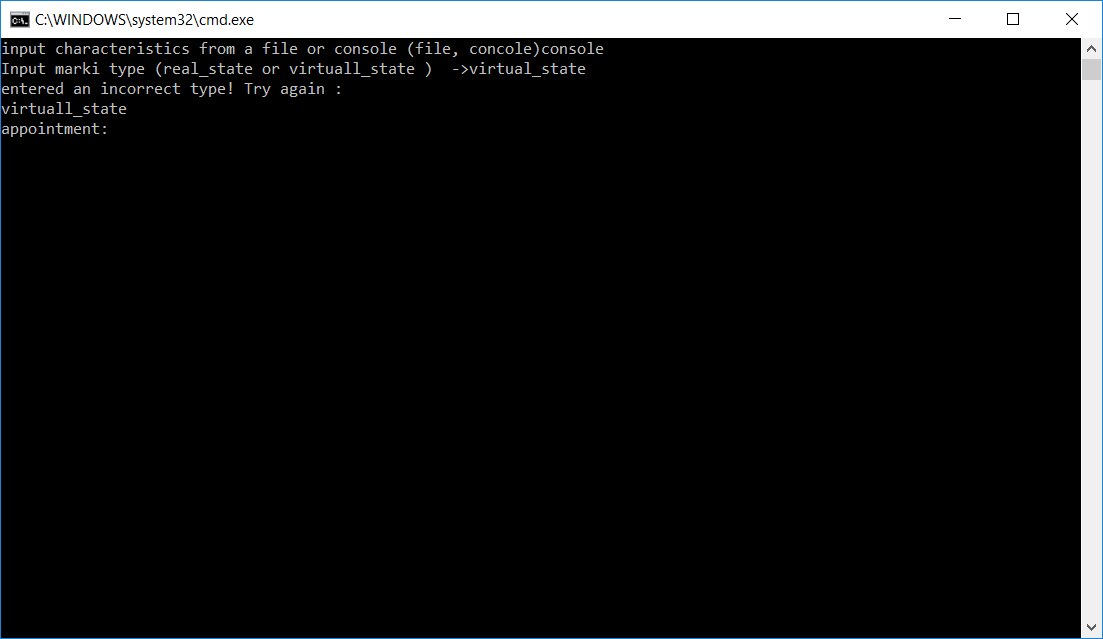


Рисунок 7. Добавление марки из консоли(Ввод неверного типа марки).

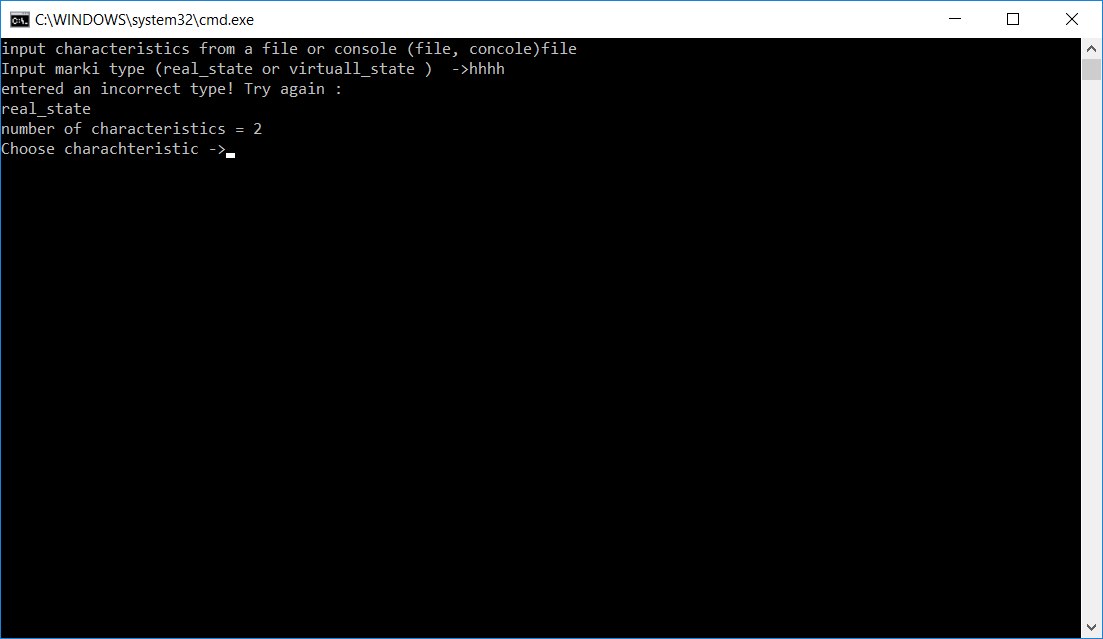


Рисунок 8. Добавление марки из файла (Ввод неверного типа марки).

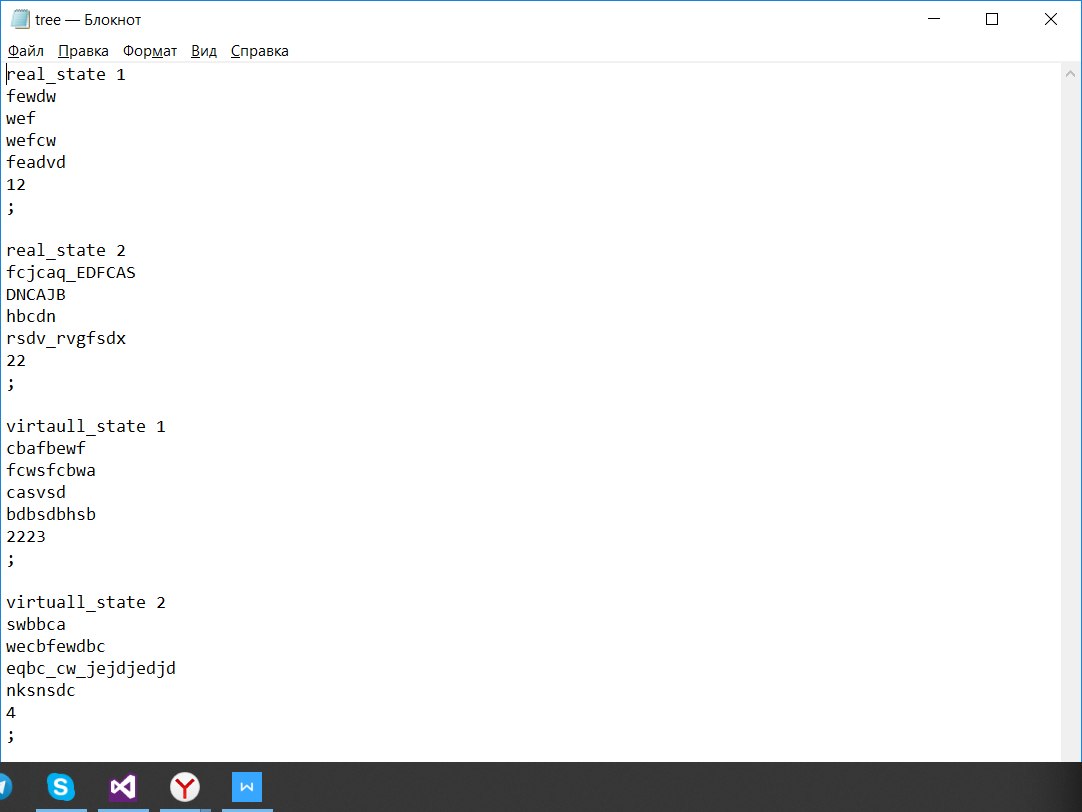


Рисунок 9. Характеристики марок в файле.

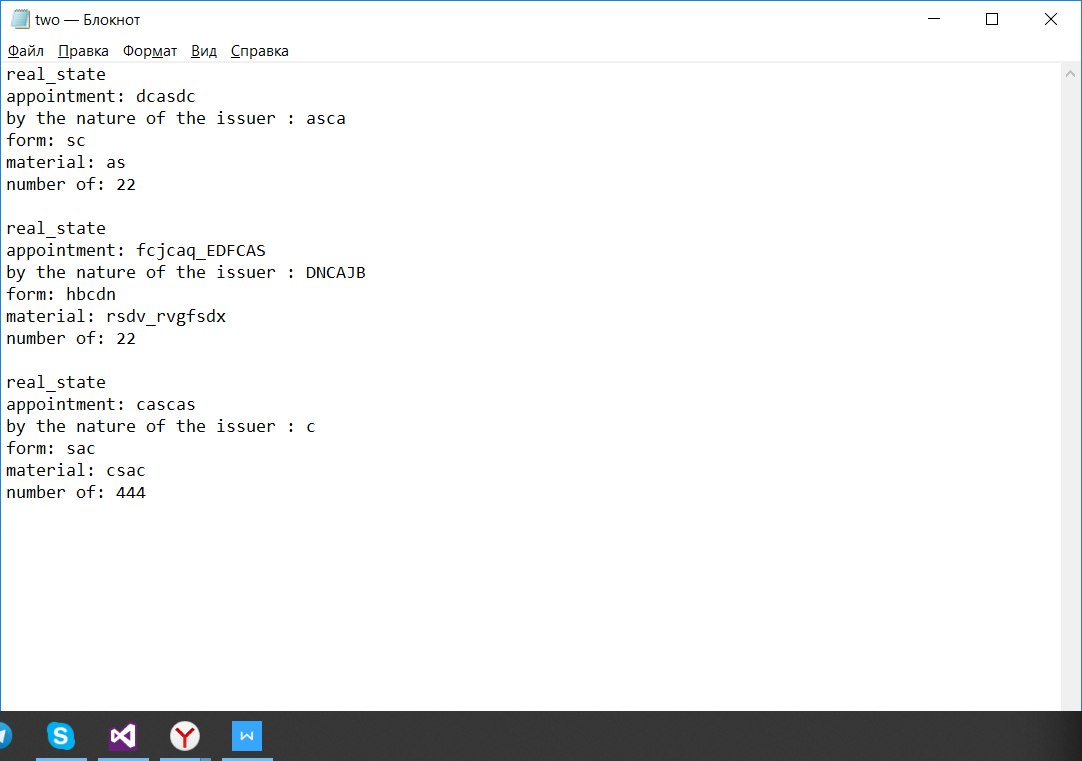


Рисунок 10. Сохранение результата в файл.