**МИНОБРНАУКИ РОССИИ**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ  
ВЫСШЕГО ОБРАЗОВАНИЯ  
**«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ  
ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ им. В.Г.ШУХОВА»  
(БГТУ им. В.Г.Шухова)**

Кафедра программного обеспечения вычислительной техники и автоматизированных систем

Лабораторная работа №2

Дисциплина: Объектно-ориентированное программирование

по теме **Модульное программирование. Интерфейсы.**

Выполнил: ст. группы ВТ-22  
Макаров Даниил Сергеевич

Проверил: Буханов Д.Г.

**Белгород 2019**

Лабораторная работа №2

**Цель работы**: Получение навыков модульной декомпозиции предметной области, создания модулей. Разработка интерфейсов.

**Задание:** Разработать программу, состоящую из трех модулей в соответствии с указанным вариантом задания. Первый модуль – основной код программы; второй содержит интерфейсы; третий модуль – реализацию этих интерфейсов. Количество структур данных ("объектов") не менее пяти.

**Содержание отчета:**

1. Тема, цель работы, вариант задания.
2. Реализация задачи на языке C++.

**Контрольные вопросы:**

1. Что такое модуль?
2. Какие бывают типы модулей в C++?
3. Структура типа «запись» (Struct).
4. Что такое интерфейс?

Программа «Справочник нумизмата»

**Coin\_base.hpp**

#include <stdio.h>

#include <malloc.h>

#include <ctype.h>

#include <cstring>

#include <stdlib.h>

#include <iostream>

#include <fstream>

using namespace std;

struct coin\_size{

double l;

double w;

double t;

void read\_size();

double get\_l();

double get\_w();

double get\_t();

};

struct coin\_discr{

double weight;

coin\_size s;

//1 - серебро, 2 - медь, 3 - золото, 4 - прочее(биметал и тд).

int material;

int condition;

void read();

void print();

};

struct coin{

char value[20];

char country[20];

unsigned year;

void read();

void print();

};

struct coin\_record{

coin c;

coin\_discr discription;

void read\_record();

void print\_title\_record();

void print\_embed\_record();

};

struct record\_arr{

coin\_record \*arr;

int size=0;

void init\_records();

void open\_records(char \*datafile\_name);

int search\_value(char\* value);

int search\_country(char\* country);

int remove\_record(int key);

void save\_records(char \*datafile\_name);

void free\_records();

};

**Coin\_base.cpp**

#include "coin\_base.hpp"

void coin\_size::read\_size(){

cout<<"Input coin sizes - ";

cin>>this->l>>this->w>>this->t;

};

double coin\_size::get\_l(){

return this->l;

};

double coin\_size::get\_w(){

return this->w;

};

double coin\_size::get\_t(){

return this->t;

};

void coin\_discr::read(){

cout<<"Input coin discription: ";

cout<<"Weight - ";

cin>>this->weight;

cout<<"Condition (1-10) - ";

cin>>this->condition;

cout<<"Material (1 - silver, 2 - copper, 3 - gold, 4 - other) - ";

cin>>this->material;

this->s.read\_size();

};

void coin\_discr::print(){

cout<<"Weight - "<<this->weight;

cout<<"Condition (1-10) - "<<this->condition;

cout<<"Material - ";

switch(this->material){

case 1:

cout<<"Silver";

break;

case 2:

cout<<"Copper";

break;

case 3:

cout<<"Gold";

break;

default:

cout<<"Other";

break;

};

cout<<"Size - ("<<this->s.get\_l()<<","<<this->s.get\_w()<<","<<this->s.get\_t()<<").";

};

void coin::read(){

cout<<"Input coin info:";

cout<<"Value - ";

cin>>value;

cout<<"Country - ";

cin>>country;

cout<<"Year - ";

cin>>year;

};

void coin::print(){

cout<<value<<", "<<country<<", "<<this->year;

};

void coin\_record::read\_record(){

this->c.read();

this->discription.read();

};

void coin\_record::print\_title\_record(){

this->c.print();

};

void coin\_record::print\_embed\_record(){

this->c.print();

this->discription.print();

};

void record\_arr::init\_records(){

this->arr=new coin\_record[this->size\*2];

};

void record\_arr::free\_records(){

free(this->arr);

};

void record\_arr::open\_records(char \*datafile\_name){

ifstream data;

data.open(datafile\_name);

data >> this->size;

this->init\_records();

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

data >> arr[i].c.value >> arr[i].c.country >> arr[i].c.year

>> arr[i].discription.weight>> arr[i].discription.material >> arr[i].discription.condition

>> arr[i].discription.s.l >> arr[i].discription.s.w>> arr[i].discription.s.t;

}

data.close();

};

void record\_arr::save\_records(char \*datafile\_name){

ofstream data;

data.open(datafile\_name);

data << this->size;

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

data << arr[i].c.value << arr[i].c.country << arr[i].c.year

<< arr[i].discription.weight << arr[i].discription.material << arr[i].discription.condition

<< arr[i].discription.s.l << arr[i].discription.s.w << arr[i].discription.s.t;

}

data.close();

}

int record\_arr::search\_value(char\* value){

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

if (strcmp (arr[i].c.value, value) == 0){

arr[i].print\_embed\_record();

return i;

}

}

cout << "-- not found --" << endl;

};

int record\_arr::search\_country(char\* country){

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

if (strcmp (arr[i].c.country, country) == 0){

arr[i].print\_embed\_record();

return i;

}

}

cout << "-- not found --" << endl;

};

int record\_arr::remove\_record(int key){

swap(arr[key],arr[this->size-1]);

this->size--;

};

**Main.cpp**

#include "coin\_base.hpp"

int main(){

int mode, pos;

record\_arr base;

char\* filename;

cout<<"Input filename: ";

cin>>filename;

base.open\_records(filename);

cout<<"Waiting command ..."<<endl;

while (mode != '0'){

mode = getchar();

mode = tolower(mode);

switch (mode){

case 'e':

cin >> pos;

base.arr[pos].print\_embed\_record();

break;

case 'a':

base.arr[base.size].read\_record();

base.size++;

break;

case 'p':

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

cout<<i+1<<". ";

base.arr[i].print\_title\_record();

cout<<endl;

}

break;

case 'r':

cin >> pos;

if (pos-1 > base.size){

cout << "unable to remove!" << endl;

}

base.remove\_record(pos-1);

break;

case 's':

base.save\_records(filename);

break;

case 'q':

base.save\_records(filename);

return 0;

case 'h':

cout<<"Coin records base:"<<endl;

cout<<" h - show this page."<<endl;

cout<<" a - add coin record."<<endl;

cout<<" p - print all records."<<endl;

cout<<" s - save records file."<<endl;

cout<<" q - close programm and save records to origin file."<<endl;

cout<<" r X - remove coin record with index X."<<endl;

cout<<" e X - print embed info about coin."<<endl;

break;

default:

cout<<"Unknown command. h to help"<<endl;

break;

}

}

return 0;

}