**Пермский национальный исследовательский политехнический университет**

Кафедра “Информационные технологии и автоматизированные системы”

**Лабораторная работа №27**

По дисциплине «Основы алгоритмизации и программирования»

**Тема:**

Сохранение данных в файле с использованием потоков.

Выполнила:

студентка группы ИВТ-19-1б

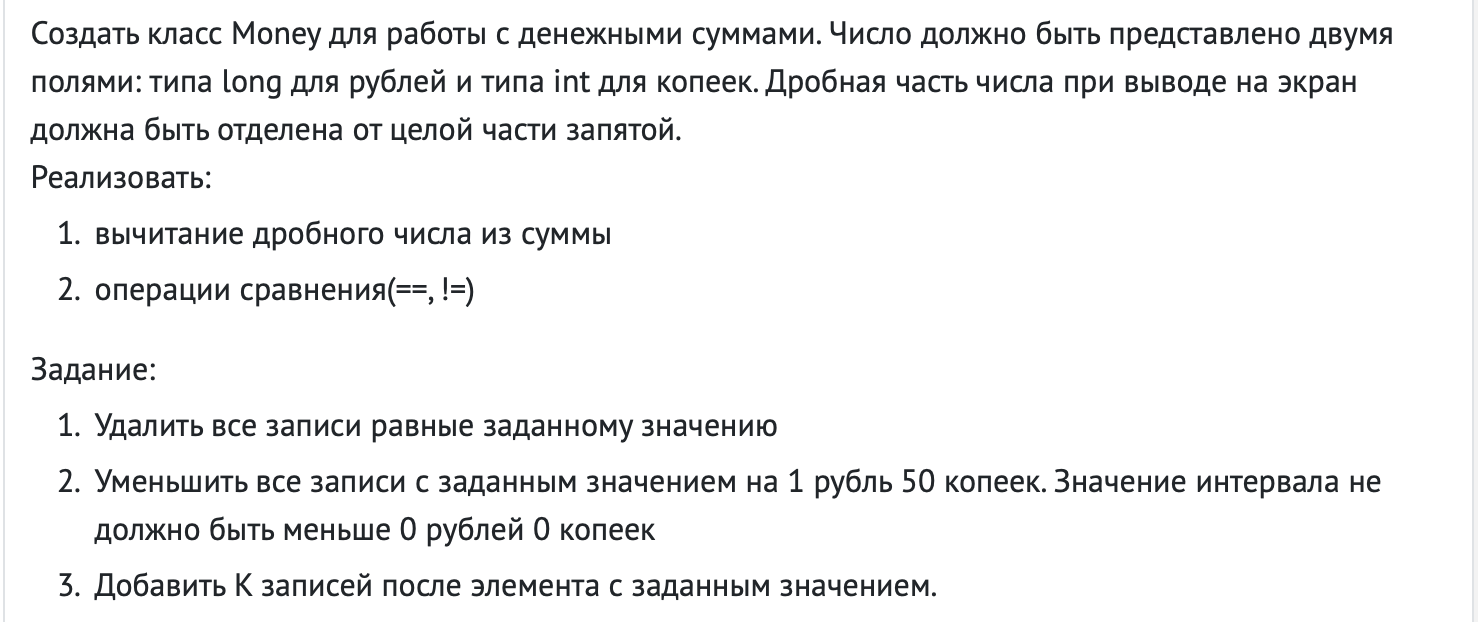
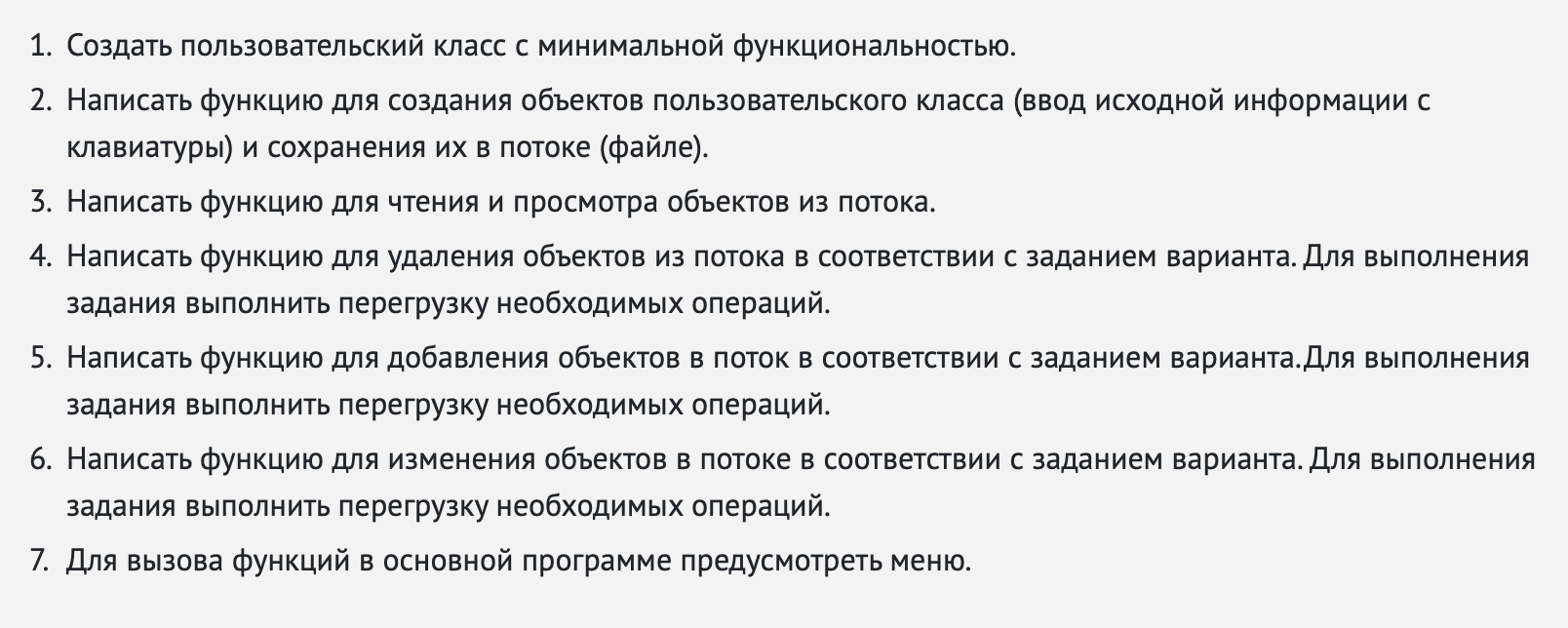
Еремеева Полина Алексеевна

Проверила:

доцент кафедры “ИТАС”

Полякова О.А.

Пермь, 2020

**Постановка задачи**

**Код**

Main.cpp

#include "file\_work.hpp"

#include "Money.hpp"

#include <iostream>

#include <string>

#include <fstream>

**using** **namespace** std;

**int** main()

{

setlocale(LC\_ALL, "rus");

**char** Name[10] = "File.txt";

make\_file(Name);

print\_file(Name);

Money num1;

**bool** f;

**int** t;

**int** h;

cout << "После какой записи добавить: ";

cin >> t;

cout << "Количество: ";

cin >> h;

**int** k;

k = add\_file(Name, t, h);

**if** (k < 0) cout << "Can't read file";

**if** (k == 0) k = add\_end(Name);

print\_file(Name);

cout << "Введите число для сравнения: " << endl;

cin >> num1;

f = comparison(Name, num1);

**if** (f == **true**)

{

cout << "Есть число" << endl;

cout << endl << "Уменьшено на 1,5: ";

reduce\_file(Name, num1);

}

**else**

{

cout << "Нет числа" << endl;

}

cout << endl << endl << endl;

}

file\_work.hpp

#include "Money.hpp"

#include <iostream>

#include <fstream>

**int** make\_file(**const** **char**\* file)

{

fstream stream(file, ios::out | ios::trunc);

**if** (!stream)**return** -1;

Money m;

**int** n;

cout << "Количество: ";

cin >> n;

cout << endl;

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

{

cin >> m;

stream << m << endl;

}

stream.close();

**return** n;

}

**int** print\_file(**const** **char**\* file)

{

fstream stream(file, ios::in);

**if** (!stream)**return** -1;

Money m;

**int** i = 0;

cout << endl;

**while** (stream >> m)

{

cout << m << " ";

i++;

}

cout << endl << endl;

stream.close();

**return** i;

}

**bool** comparison(**const** **char**\* file, Money& c)

{

fstream temp("temp", ios::out);

fstream stream(file, ios::in);

**if** (!stream)**return** -1;

**bool** f = **false**;

Money m;

**while** (stream >> m)

{

**if** (stream.eof())**break**;

**if** ((m == c))

{

temp << m;

f = **true**;

}

}

stream.close();

temp.close();

remove(file);

rename("temp", file);

**return** f;

}

**void** reduce\_file(**const** **char**\* f\_name, Money& c)

{

fstream temp("temp", ios::out);

fstream stream(f\_name, ios::in);

//if (!stream)return -1;

**int** i = 0;

Money m;

**while** (stream >> m)

{

**if** (stream.eof())**break**;

i++;

**if** (m == c)

{

temp << m - 1;

cout << m << " " << endl;

}

}

stream.close();

temp.close();

remove(f\_name);

rename("temp", f\_name);

}

**int** add\_end(**const** **char**\* f\_name)

{

**int** i = 0;

fstream stream(f\_name, ios::app);

**if** (!stream)**return** -1;

Money A;

cout << "New object\n";

cin >> A;

stream << A;

i++;

**return** i;

}

**int** add\_file(**const** **char**\* f\_name, **int** k, **int** v)

{

fstream temp("temp", ios::out);

fstream stream(f\_name, ios::in);

**if** (!stream)**return** -1;

Money m;

**int** x = 0;

**int** i = 0;

**int** l = 0;

**while** (stream >> m)

{

**if** (stream.eof())**break**;

**if** (i == k)

{

**while** (x < v)

{

Money mm;

cin >> mm;

temp << mm;

x++;

l++;

}

}

i++;

temp << m;

}

stream.close();

temp.close();

remove(f\_name);

rename("temp", f\_name);

**return** l;

}

Money.hpp

#pragma once

#include <iostream>

#include <string>

#include <fstream>

**using** **namespace** std;

**class** Money {

**private**:

**long** ruble;

**int** penny;

**public**:

Money();

Money(**long**, **int**);

Money(**const** Money&);

~Money() {};

**friend** **bool** **operator**==(Money& a, Money& b);

**friend** **bool** **operator**!=(Money& a, Money& b);

**friend** Money **operator**-(Money& c, **int**);

Money& **operator**++();

Money **operator**++(**int**);

**friend** ostream& **operator** <<(ostream& out\_1, **const** Money& p);

**friend** istream& **operator** >>(istream& in\_1, Money& p);

**friend** fstream& **operator**>>(fstream& in\_2, Money& p);

**friend** fstream& **operator**<<(fstream& out\_2, **const** Money& p);

};

Money.cpp

#include "Money.hpp"

#include <iostream>

#include <string>

**using** **namespace** std;

Money::Money()

{

ruble = 0;

penny = 0;

}

Money::Money(**long** rub, **int** pen)

{

ruble = rub;

penny = pen;

}

Money::Money(**const** Money& p)

{

ruble = p.ruble;

penny = p.penny;

}

**bool** **operator**==(Money& a, Money& b)

{

**if** ((a.ruble == b.ruble) && (a.penny == b.penny))

{

**return** **true**;

}

**else**

{

**return** **false**;

}

}

**bool** **operator**!=(Money& a, Money& b)

{

**if** ((a.ruble != b.ruble) && (a.penny != b.penny))

{

**return** **true**;

}

**else**

{

**return** **false**;

}

}

Money **operator**-(Money& c, **int** q)

{

**long** k = c.ruble - 1;

**int** t = c.penny - 5;

c.ruble = k;

c.penny = t;

**return** c;

}

Money& Money::**operator**++()

{

++penny;

**return** \***this**;

}

Money Money::**operator**++(**int**)

{

Money temp(ruble, penny);

**long** k = 0;

k = temp.ruble \* 100 + temp.penny;

k++;

temp.ruble = k / 100;

temp.penny = k - ((k / 100) \* 100);

**return** temp;

}

ostream& **operator**<<(ostream& out\_1, **const** Money& p)

{

**long** k = 0;

k = p.ruble \* 100 + p.penny;

out\_1 << k / 100 << "," << k % 100 << " ";

**return** out\_1;

}

istream& **operator**>>(istream& in\_1, Money& p)

{

cout << "Рубли: "; in\_1 >> p.ruble;

cout << "Копейки: "; in\_1 >> p.penny;

**return** in\_1;

}

fstream& **operator**>>(fstream& in\_2, Money& p)

{

in\_2 >> p.ruble;

in\_2 >> p.penny;

**return** in\_2;

}

fstream& **operator**<<(fstream& out\_2, **const** Money& p)

{

out\_2 << p.ruble << endl << p.penny << endl;

**return** out\_2;

}

**Тестирование**

