Содержание

[Постановка задачи и общие требования 2](#_Toc99236263)

[Авторское понимание 2](#_Toc99236264)

[Текст программы 3](#_Toc99236265)

[Результаты тестирования 6](#_Toc99236266)

[Вывод 6](#_Toc99236267)

# Постановка задачи и общие требования

Реализовать класс-оболочку Number для числового типа float. Реализовать методы сложения и деления. Создать производный класс Real, в котором реализовать метод возведения в произвольную степень, и метод для вычисления логарифма числа. Общие требования:

Во всех заданиях реализовать вывод на экран, методы получения значений полей и методы установки значений полей, а также необходимые конструкторы (если это не указано в задании явно). Конструкторы и методы обязательно должны проверять параметры на допустимость; в случае неправильных данных — выводить сообщение об ошибке и заканчивать работу. Преобразование в строку реализовать в виде функции преобразования toString().

Для демонстрации работы с объектами нового типа во всех заданиях требуется написать главную функцию. В программе должны присутствовать различные способы создания объектов и массивов объектов. Программа должна демонстрировать использование всех функций и методов.

Продемонстрировать принцип подстановки.

Для выполнения работы необходимо:

* формализовать задачу;
* составить описание структуры и класса (*устно*);
* составить алгоритм для каждого метода, описанного в структуре и классе, и определить все методы (*устно*);
* создать объекты описанного типа;
* разработать набор тестов, на которых будет проверяться программа (*устно*);
* продемонстрировать работу программы на наборе тестов;
* оформить работу и отчитаться по ней.

# Авторское понимание

Для решения данной задачи необходимо создать класс Number, в котором определены методы сложения чисел и деления чисел. Далее создать класс наследник Real , в котором определен метод возведения в произвольную степень и метод вычисления логарифма числа.

# Текст программы

Number.h

#pragma once

#include <iostream>

#include<string>

#include<math.h>

using namespace std;

class Number   
{

protected:

float a;

float b;

public:

Number();

Number(float odin, float dva);

int init();

float slojenie\_chisel(float a, float b);

float delenie\_chisel(float a, float b);

string toString();

};

Number.cpp

#include "Number.h"

Number::Number():a(0.0),b(0.0)   
{

}

Number::Number(float odin, float dva) : a(odin), b(dva)   
{

}

int Number::init()   
{

if ( b == 0.0)   
{

return 0;

}

else   
{

return 1;

}

}

float Number::slojenie\_chisel(float a, float b)   
{

if ((a > 0) && (b > 0))   
{

a = a + b;

}

else if ((a > 0) && (b < 0))   
{

a = a - b;

}

else if ((a < 0) && (b > 0))   
{

a = b - a;

}

else   
{

a = 0 - (a + b);

}

return a;

}

float Number::delenie\_chisel(float a, float b)   
{

a = a / b;

return a;

}

string Number::toString()   
{

string A;

A += "Pervoe:" + to\_string(a) + "\nVtoroe: " + to\_string(b) + "\nSumma: " + to\_string(slojenie\_chisel(a,b)) + "\nDelenie: " + to\_string(delenie\_chisel(a, b));

return A;

}

Real.h

#pragma once

#include "Number.h"

class Real :

public Number

{

float c; int n;

public:

Real();

Real(float first);

void vo3vedenie\_v\_step(int n);

void vichislenie\_logarifma(int x,int z);

float get\_step();

float get\_logarifm();

friend ostream& operator<<(ostream& out, Real& p);

friend istream& operator>>(istream& in, Real& p);

};

Real.cpp

#include "Real.h"

Real::Real():c(0.0)   
{

}

Real::Real(float first):c(first)   
{

}

void Real::vo3vedenie\_v\_step(int n)   
{

c= pow(c, n);

}

float Real::get\_step()   
{

return c;

}

void Real::vichislenie\_logarifma(int x, int z)   
{

c= (log(z)) / (log(x));

}

float Real::get\_logarifm()   
{

return c;

}

ostream& operator<<(ostream& out, Real& p)   
{

out << p.toString();

p.vo3vedenie\_v\_step(p.n);

out << "\nChislo posle vozvedeniya: " << p.get\_step() << endl;

p.vichislenie\_logarifma(3,5);

out << "Logarifm raven: " << p.get\_logarifm() << endl;

return out;

}

istream& operator>>(istream& in, Real& p)   
{

cout << "Vvedite pervoe chislo: ";

in >>p.a;

cout << "Vvedie vtoroe chislo: ";

in >> p.b;

cout << "Vvedite c: ";

in >> p.c;

cout << "Vvedite stepen: ";

in >> p.n;

return in;

}

Main.cpp

#include "Number.h"

#include "Real.h"

#include <iostream>

#include <conio.h>

using namespace std;

int main()   
{

Real chislo;

cin >> chislo;

if (chislo.init() == 0)   
{

cout << "Oshibka" << endl;

}

else   
{

cout << chislo << endl;

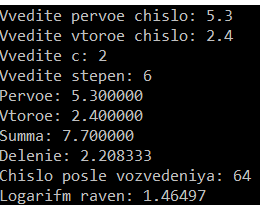
}

\_getch();

return 0;

}

# Результаты тестирования



# Вывод

Результаты работы программы совпадают с ожидаемыми результатами, а это значит, что программа работает правильно.