# **Звіт про виконання лабораторної роботи № 2.2**

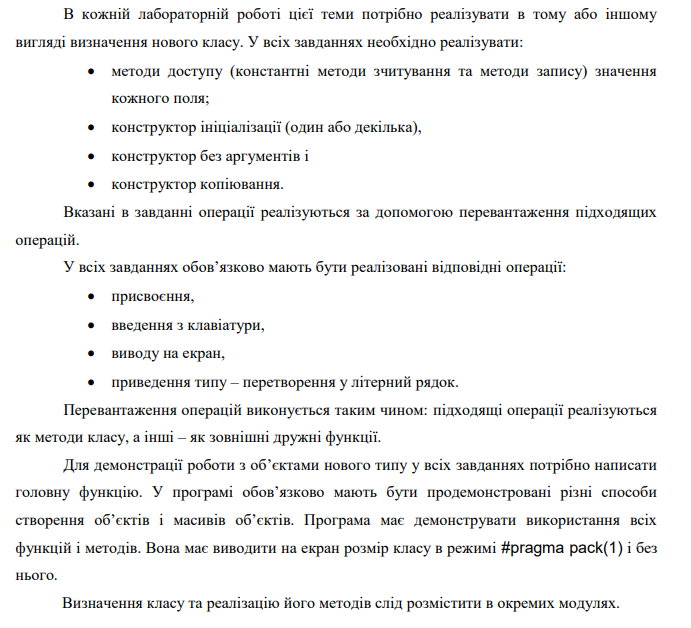
## «Перевантаження операцій» з дисципліни «Об’єктно-орієнтоване програмування»

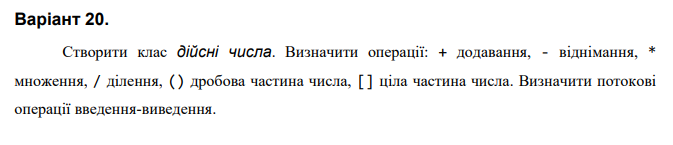
## Студента групи «**ІТ-12**» - **Степанчука Сергія**

## Мета роботи

Освоїти використання конструкторів та перевантаження операцій.

## Умова завдання

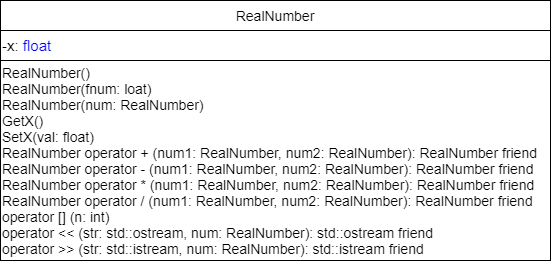




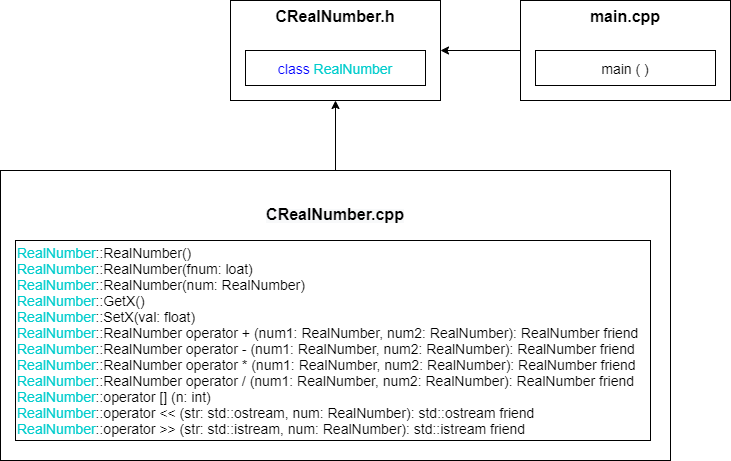
## Посилання на git-репозиторій з проектом:

<https://github.com/SergiyStepanchuk/OOP_Lab_2.2>

## UML-діаграма класів



## Структурна схема



## Текст програми

### // CRealNumber.h

#pragma once

#include <iostream>

class RealNumber

{

float x;

public:

float GetX() const { return x; }

void SetX(float value) { x = value; }

RealNumber();

RealNumber(float);

RealNumber(const RealNumber&);

friend RealNumber operator + (RealNumber&, RealNumber&);

friend RealNumber operator - (RealNumber&, RealNumber&);

friend RealNumber operator \* (RealNumber&, RealNumber&);

friend RealNumber operator / (RealNumber&, RealNumber&);

int operator [] (int);

float operator () ();

friend std::ostream& operator << (std::ostream&, const RealNumber&);

friend std::istream& operator >> (std::istream&, RealNumber&);

};

### // CRealNumber.cpp

#include "CRealNumber.h"

#include <cmath>

using namespace std;

RealNumber::RealNumber()

{

x = 0;

}

RealNumber::RealNumber(float y)

{

x = y;

}

RealNumber::RealNumber(const RealNumber& r)

{

x = r.x;

}

RealNumber operator+(RealNumber& a, RealNumber& b)

{

RealNumber t(0);

t.x = a.x + b.x;

return t;

}

RealNumber operator-(RealNumber& a, RealNumber& b)

{

RealNumber t(0);

t.x = a.x - b.x;

return t;

}

RealNumber operator\*(RealNumber& a, RealNumber& b)

{

RealNumber t(0);

t.x = a.x \* b.x;

return t;

}

RealNumber operator/(RealNumber& a, RealNumber& b)

{

if (b.x)

{

RealNumber t(0);

t.x = a.x / b.x;

return t;

}

else

{

cerr << "Error!" << endl;

return -1;

}

}

int RealNumber::operator[](int i)

{

float full;

modf(x, &full);

return full;

}

float RealNumber::operator ()()

{

float full;

return modf(x, &full);

}

ostream& operator << (ostream& out, const RealNumber& a)

{

out << a.x;

return out;

}

istream& operator >> (istream& in, RealNumber& a)

{

in >> a.x;

return in;

}

### // main.cpp

#include "CRealNumber.h"

using namespace std;

int main()

{

RealNumber a, b;

cout << "a = ? "; cin >> a;

cout << "b = ? "; cin >> b;

cout << endl;

cout << "a + b = " << a + b << endl;

cout << "a - b = " << a - b << endl;

cout << "a \* b = " << a \* b << endl;

cout << "a / b = " << a / b << endl

<< endl;

cout << "a[]: " << a[2] << endl;

cout << "a(): " << a() << endl;

return 0;

}

## UnitTest

### Код

#include "pch.h"

#include "CppUnitTest.h"

#include "../Program/CRealNumber.h"

#include "../Program/CRealNumber.cpp"

using namespace Microsoft::VisualStudio::CppUnitTestFramework;

namespace UnitTest

{

TEST\_CLASS(UnitTest)

{

public:

TEST\_METHOD(TestMethod1)

{

RealNumber rr(33.1);

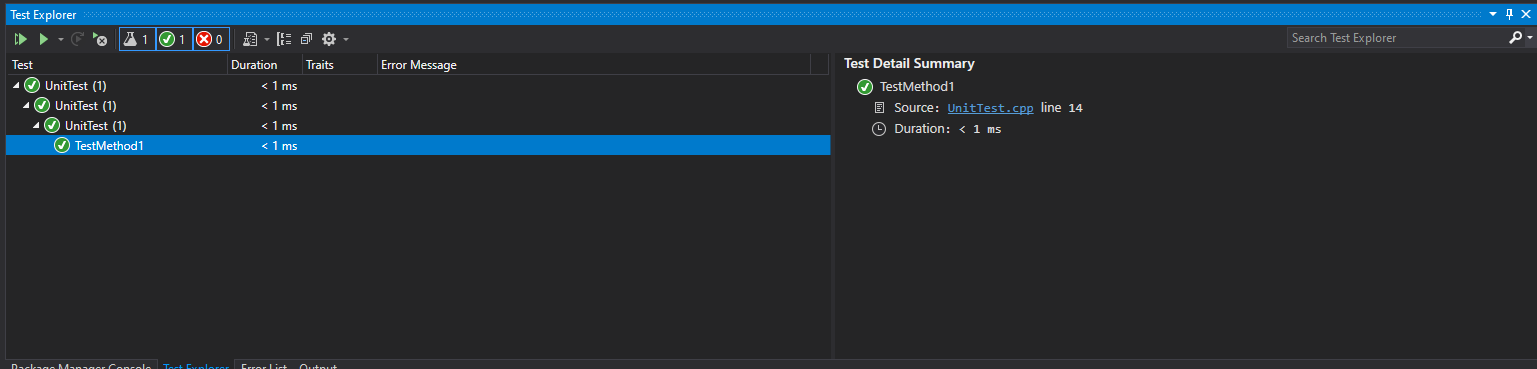
Assert::AreEqual(rr.GetX(), 33.1f);

}

};

}

### Результат



## Висновок

Освоїв використання конструкторів та перевантаження операцій.