# **Звіт про виконання лабораторної роботи № 1.3**

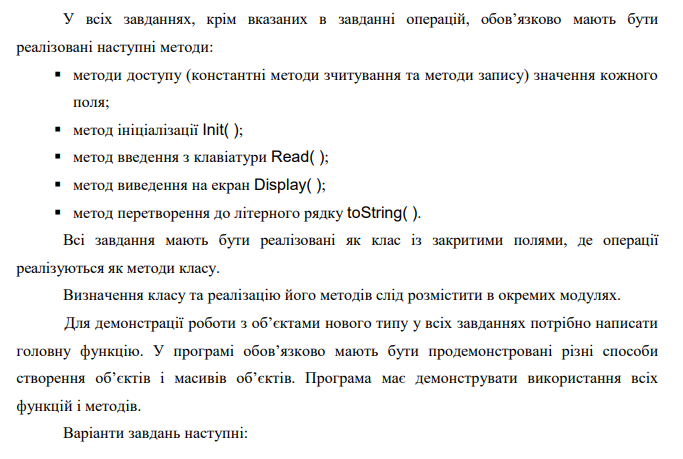
## «Оголошення та будова класу» з дисципліни «Об’єктно-орієнтоване програмування»

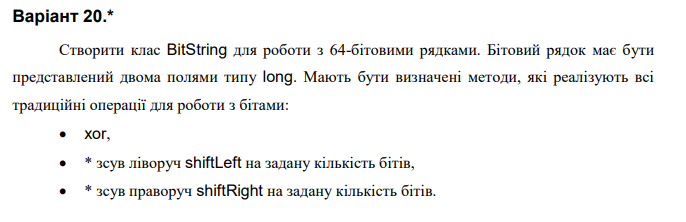
## Студента групи «**ІТ-12**» - **Степанчука Сергія**

## Мета роботи

Освоїти використання класів та об’єктів

## Умова завдання

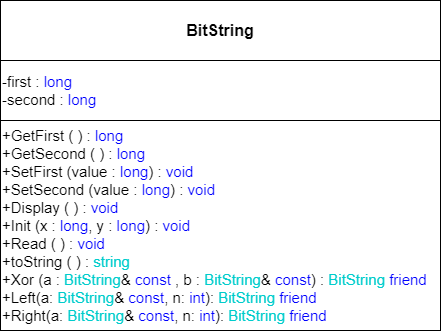




## Посилання на git-репозиторій з проектом:

<https://github.com/SergiyStepanchuk/OOP_Lab_1.3>

## UML-діаграма класів



## Структурна схема



## Текст програми

### // BitString.h

#pragma once

//визначення класу

#include <string>

#include <iostream>

#include <sstream>

class BitString

{

private:

long first;

long second;

public:

long GetFirst() const;

long GetSecond() const;

void SetFirst(double value);

void SetSecond(double value);

void Init(long, long);

void Read();

void Display() const;

std::string toString() const;

friend BitString Xor(const BitString& a, const BitString& b);

friend BitString Left(const BitString& a, int n);

friend BitString Right(const BitString& a, int n);

};

### // BitString.cpp

//реалізація методів класу

#include "BitString.h"

using namespace std;

long BitString::GetFirst() const

{

return first;

}

long BitString::GetSecond() const

{

return second;

}

void BitString::SetFirst(double value)

{

first = value;

}

void BitString::SetSecond(double value)

{

second = value;

}

void BitString::Display() const

{

cout << toString();

}

void BitString::Init(long x, long y)

{

first = x;

second = y;

}

void BitString::Read()

{

long x, y;

cout << "Input your numbers :" << endl;

cout << " First = "; cin >> x;

cout << " Second = "; cin >> y;

Init(x, y);

cout << endl;

}

string BitString::toString() const

{

stringstream sout;

sout << " First = " << first << endl;

sout << " Second = " << second << endl;

return sout.str();

}

BitString Xor(const BitString& a, const BitString& b)

{

BitString t;

t.first = a.first ^ b.first;

t.second = a.second ^ b.second;

return t;

}

BitString Left(const BitString& a, int n)

{

BitString t = a;

long tmp[2];

const int last\_bit = sizeof(long) \* 8 - 1;

for (size\_t i = 0; i < n; i++)

{

tmp[0] = t.first << 1;

tmp[1] = t.second << 1;

if (t.first & (1 << last\_bit))

tmp[1] |= (1 << 0);

else

tmp[1] &= ~(1 << 0);

if (t.second & (1 << last\_bit))

tmp[0] |= (1 << 0);

else

tmp[0] &= ~(1 << 0);

t.first = tmp[0];

t.second = tmp[1];

}

return t;

}

BitString Right(const BitString& a, int n)

{

BitString t = a;

long tmp[2];

const int last\_bit = sizeof(long) \* 8 - 1;

for (size\_t i = 0; i < n; i++)

{

tmp[0] = t.first >> 1;

tmp[1] = t.second >> 1;

if (t.first & (1 << 0))

tmp[1] |= (1 << last\_bit);

else

tmp[1] &= ~(1 << last\_bit);

if (t.second & (1 << 0))

tmp[0] |= (1 << last\_bit);

else

tmp[0] &= ~(1 << last\_bit);

t.first = tmp[0];

t.second = tmp[1];

}

return t;

}

### // main.cpp

// функція main

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <iostream>

#include "BitString.h"

using namespace std;

int main()

{

BitString o;

o.Read();

int n;

cout << o.toString() << endl << endl;

// ShiftLeft

cout << "How many bits do you want to shift left?" << endl;

cout << "Input: "; cin >> n; cout << endl;

BitString a = Left(o, n);

cout << a.toString() << endl;

// ShiftRight

cout << "How many bits do you want to shift right?" << endl;

cout << "Input: "; cin >> n; cout << endl;

BitString b = Left(o, n);

cout << Right(o, n).toString() << endl;

// Xor

cout << "Xor: " << endl;

cout << Xor(a, b).toString() << endl;

cin.get();

return 0;

}

## UnitTest

### Код

#include "pch.h"

#include "CppUnitTest.h"

#include "../Program/BitString.h"

#include "../Program/BitString.cpp"

using namespace Microsoft::VisualStudio::CppUnitTestFramework;

namespace UnitTest

{

TEST\_CLASS(UnitTest)

{

public:

TEST\_METHOD(Xorr)

{

BitString a, b;

a.SetFirst(4);

a.SetSecond(4);

b.SetFirst(8);

b.SetSecond(8);

BitString c = Xor(a, b);

Assert::AreEqual(c.GetFirst(), 12l);

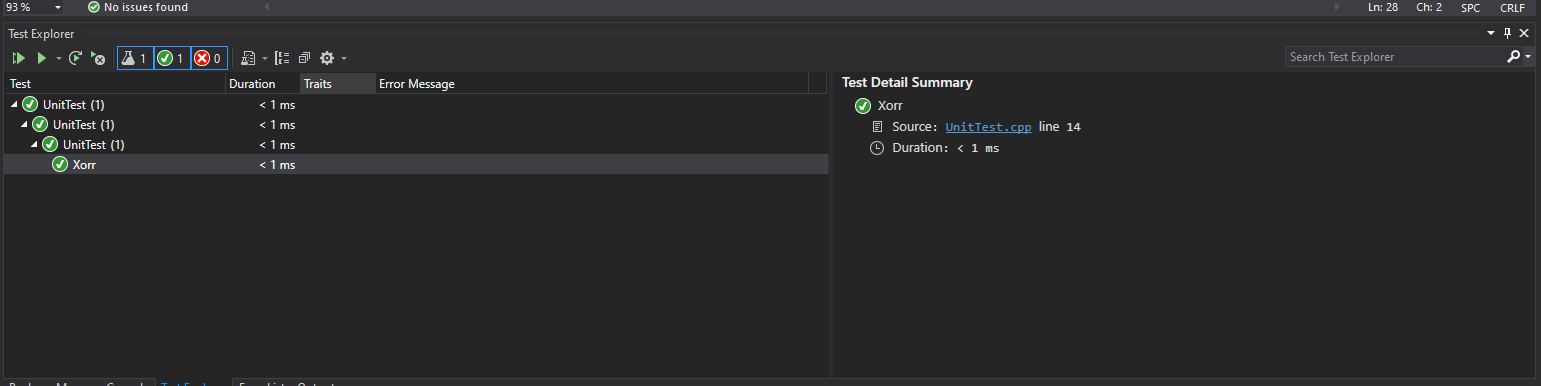
Assert::AreEqual(c.GetSecond(), 12l);

}

};

}

### Результат



## Висновок

Навчився використовувати класи та обєкти.