**мІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**нАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «лЬВІВСЬКА ПОЛІТЕХНІКА»**

**ІНСТИТУТ КОМП’ЮТЕРНИХ НАУК ТА ІНОРМАЦІЙНИХ ТЕХНОЛОГІЙ**

**Кафедра ІСМ**

|  |
| --- |
| **2022** |



**ЗВІТ**

**Про виконання лабораторної роботи №5.1(B)**

**«Класи з опрацюванням виняткових ситуацій»**

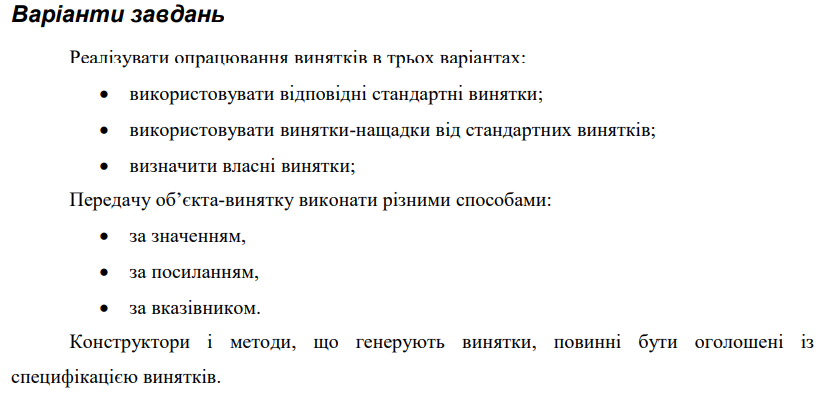
**З дисципліни**

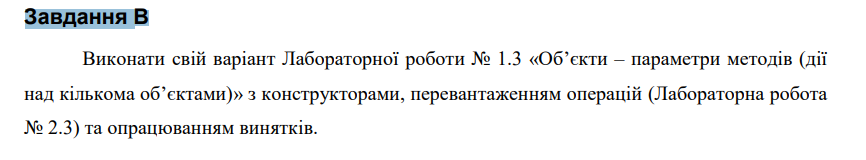
**«Об’єктно-орієнтоване програмування»**

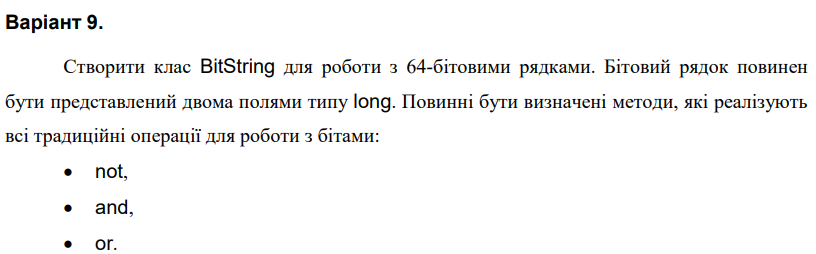
**Студента групи ІТ-11**

**Лисецького Володимира Любомировича**

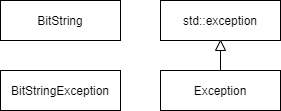
**Умова завдання:**

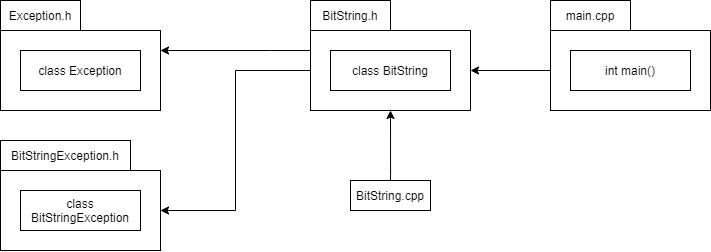
****

****

****

**UML-діаграма класів:**



**Структурна схема програми: **

**Текст програми:**

///////////////////////////////////////////////////////////////////////////

// BitString.h

#include <iostream>

#include <string>

#include "BitStringException.h"

#include "Exception.h"

#pragma once

using namespace std;

class BitString

{

private:

long a;

long b;

public:

long getA() const { return a; };

long getB() const { return b; };

void setA(long value) { a = value; }

void setB(long value) { b = value; }

operator string() const;

BitString& operator = (const BitString& obj);

BitString();

BitString(long x, long y);

BitString(BitString& obj);

BitString operator ++();

BitString operator --();

BitString operator ++(int);

BitString operator --(int);

friend BitString operator ! (BitString obj);

friend BitString operator | (BitString obj1, BitString obj2);

friend BitString operator & (BitString obj1, BitString obj2);

friend ostream& operator << (ostream& out, BitString& obj);

friend istream& operator >> (istream& in, BitString& obj);

};

#pragma once

#include<iostream>

#include<string>

using namespace std;

class BitStringException

{

string message;

public:

BitStringException(string message) : message(message) {}

string What() { return message; }

};

// Lab5\_1B.cpp : This file contains the 'main' function. Program execution begins and ends there.

//

#include <iostream>

#include "BitString.h"

using namespace std;

int main()

{

BitString b1, b2;

try {

cout << "first object" << endl;

cin >> b1;

}

catch (BitStringException& e)

{

cout << "NumberException " << e.What() << endl;

}

catch (BitStringException\* e)

{

cout << "NumberException " << e->What() << endl;

}

catch (Exception\* e)

{

cout << "exception& " << e->what() << endl;

}

catch (Exception& e)

{

cout << "exception& " << e.what() << endl;

}

catch (invalid\_argument b)

{

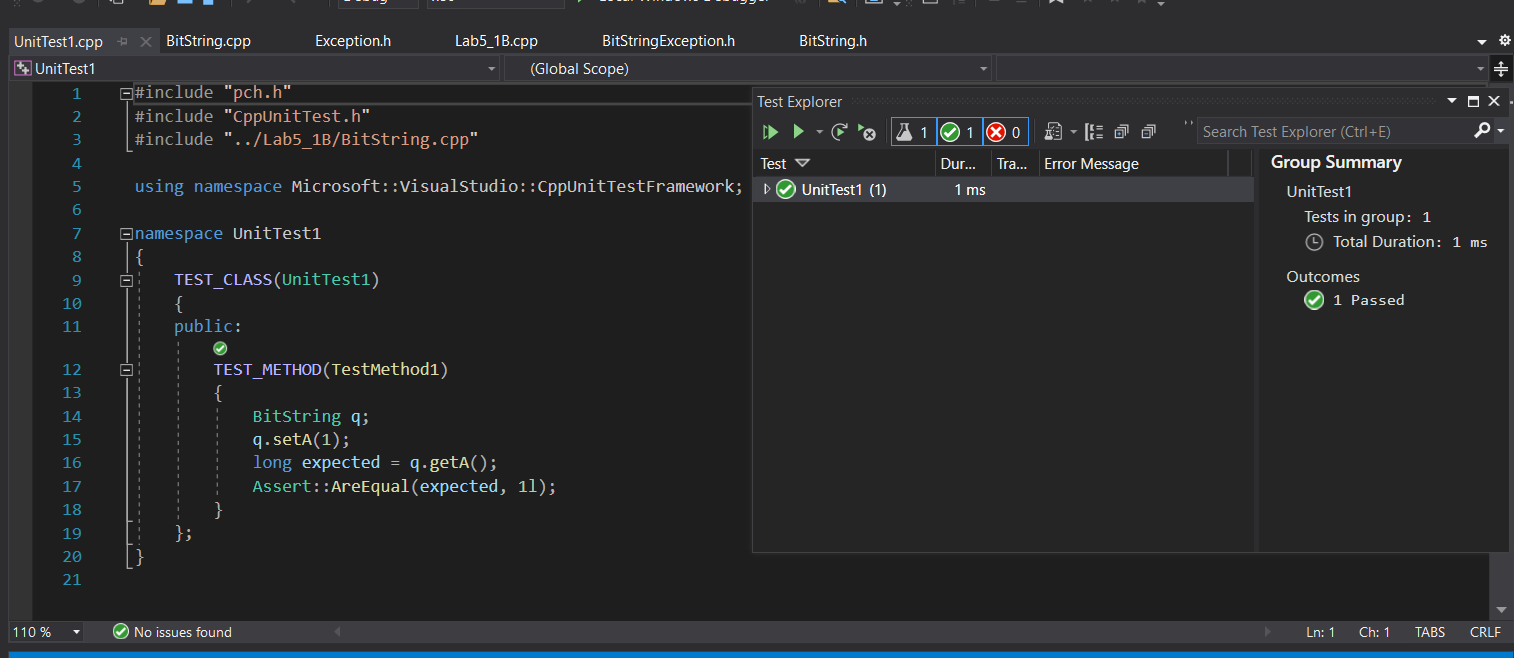
cerr << "invalid\_argument " << b.what() << endl;

}

}

**Посилання на git-репозиторій з проектом:**

**https://github.com/MarshmallowSoup/oop-lab-5-1**

**Результати unit-тесту:** ****