**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(E)**

**«Класи з опрацюванням виняткових ситуацій»**

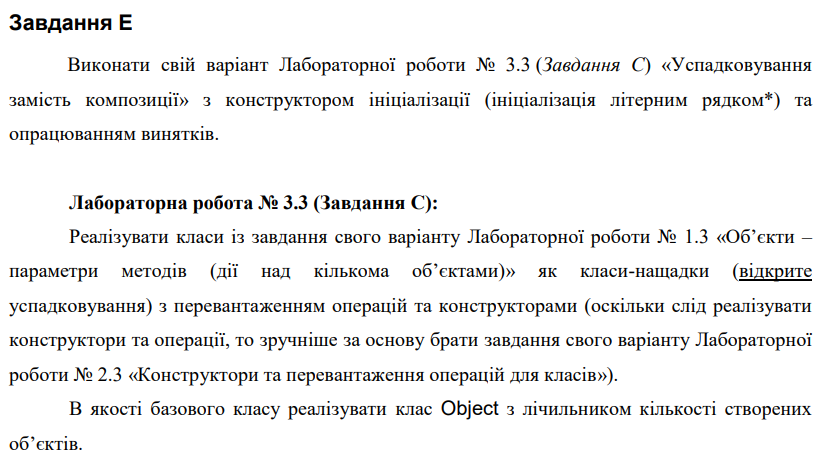
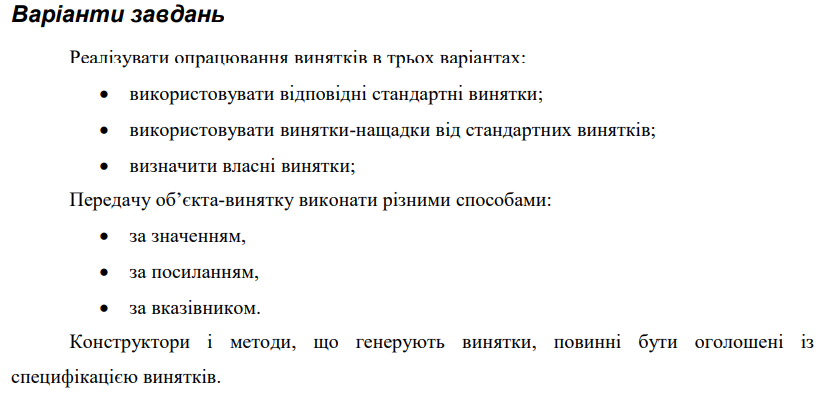
**З дисципліни**

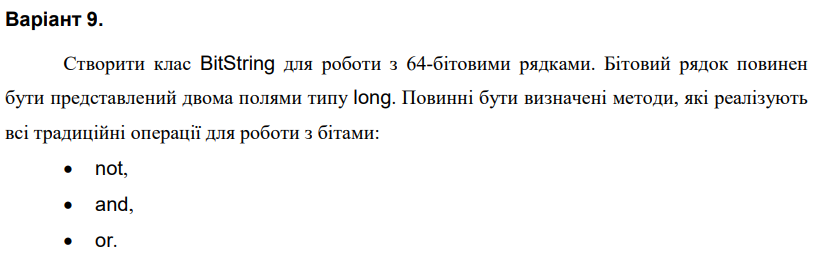
**«Об’єктно-орієнтоване програмування»**

**Студента групи ІТ-11**

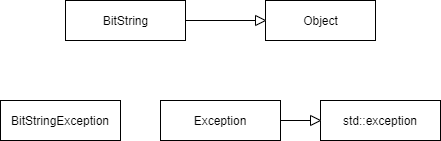
**Лисецького Володимира Любомировича**

**Умова завдання:**

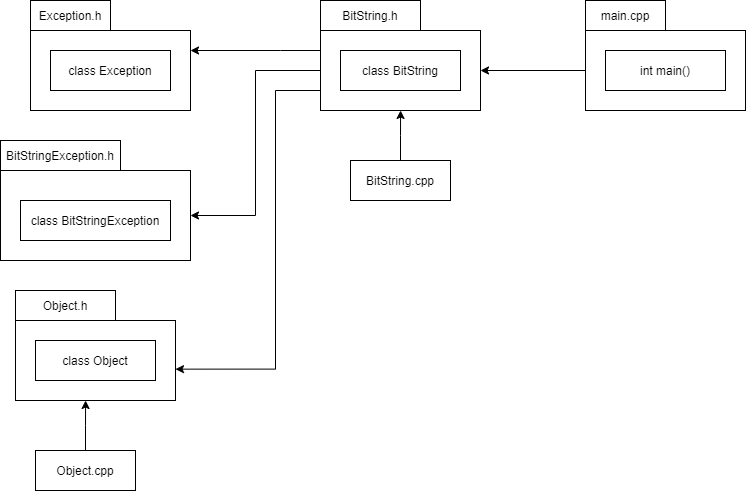
****

****

**UML-діаграма класів:**



**Структурна схема програми:**



**Текст програми:**

#pragma once

#include<iostream>

#include<string>

#include<exception>

using namespace std;

class Exception : public exception

{

string message;

public:

Exception(string message) : message(message) {}

string What() { return message; }

};

#pragma once

class Object

{

protected:

static unsigned int count;

public:

Object();

~Object();

unsigned int get\_count() { return count; };

};

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

// BitString.h

#pragma once

#include "Object.h"

#include <iostream>

#include <string>

#include "BitStringException.h"

#include "Exception.h"

using namespace std;

class BitString :

public Object

{

private:

long a;

long b;

public:

long getA() const { return a; };

long getB() const { return b; };

void setA(long value);

void setB(long value);

operator string() const;

BitString& operator = (const BitString& obj);

BitString();

BitString(long first, long second);

BitString(char\* first, char\* second);

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);

};

#include <iostream>

#include "BitString.h"

using namespace std;

int main()

{

BitString b1;

try {

cin >> b1;

cout << endl;

}

catch (BitStringException\* q) {

cout << "BaseBitStringException: " << q->What() << endl;

}

catch (BitStringException& q) {

cout << "BaseBitStringException: " << q.What() << endl;

}

catch (Exception& q) {

cout << "Exception: " << q.What() << endl;

}

catch (Exception\* q) {

cout << "Exception: " << q->What() << endl;

}

catch (invalid\_argument q) {

cout << "invalid\_argument: " << q.what() << endl;

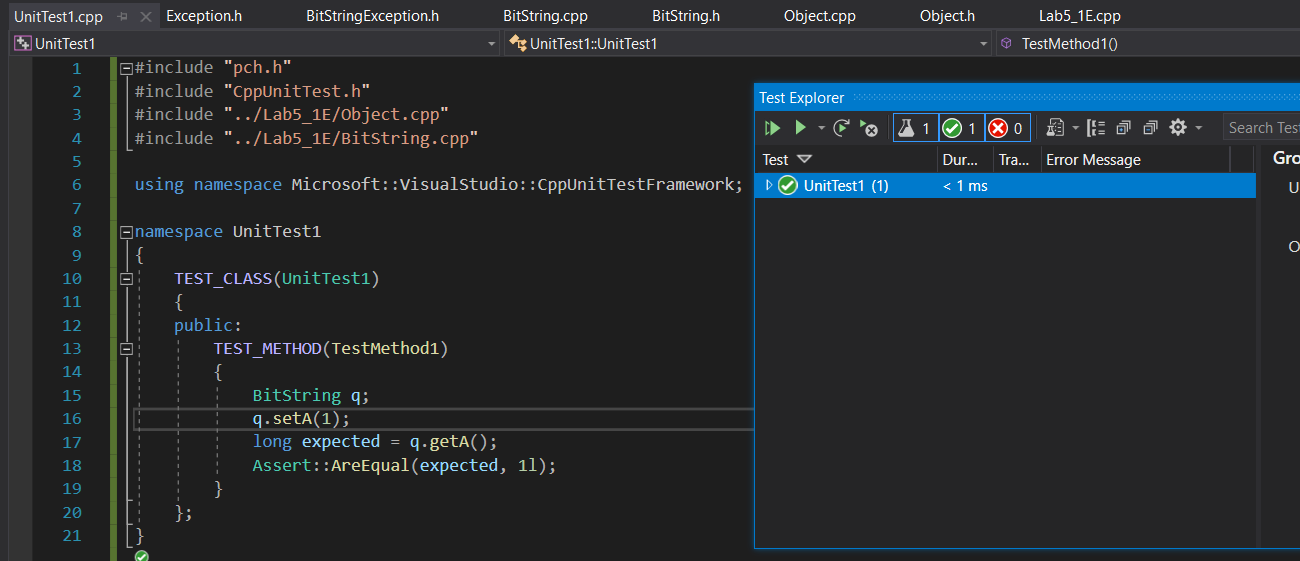
}

cout << b1 << endl;

cout << "Amount of BitString objects: " << b1.get\_count() << endl;

}

**Посилання на git-репозиторій з проектом:** **https://github.com/MarshmallowSoup/oop-lab-5-1**

**Результати unit-тесту:** ****