**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(А)**

**«Класи з опрацюванням виняткових ситуацій»**

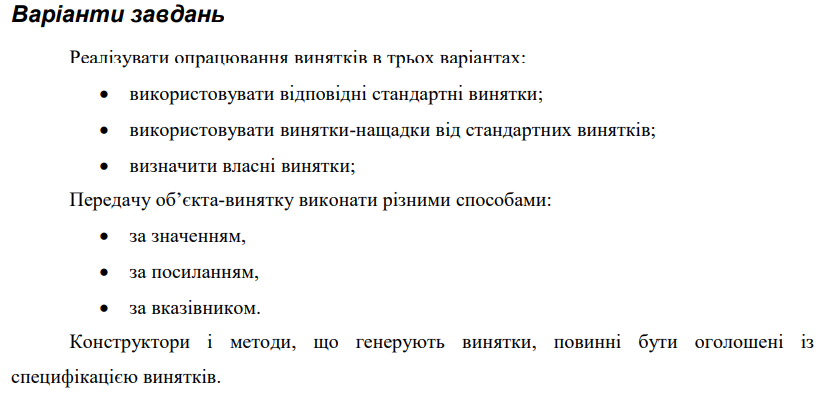
**З дисципліни**

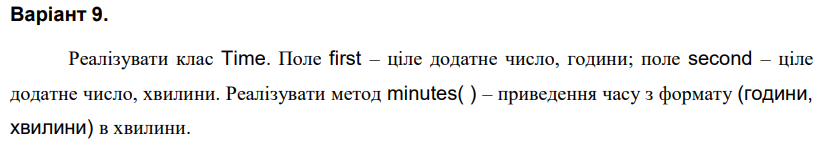
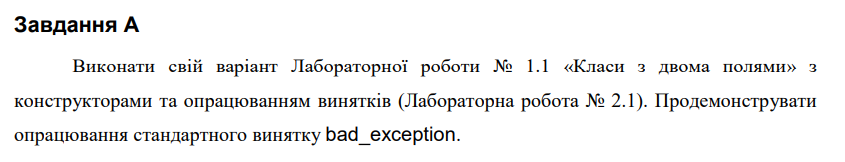
**«Об’єктно-орієнтоване програмування»**

**Студента групи ІТ-11**

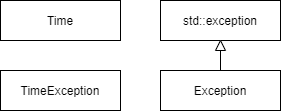
**Лисецького Володимира Любомировича**

**Умова завдання:**

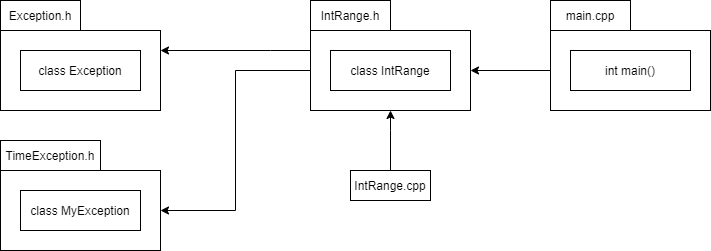
****

****

**UML-діаграма класів:**



**Структурна схема програми:**



**Текст програми:**

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

// Time.h

#pragma once

#include <string>

#include <sstream>

#include <iostream>

#include "TimeException.h"

#include "Exception.h"

using namespace std;

class Time

{

private:

int first, second;

public:

int minutes();

int get\_first() { return first; };

int get\_second() { return second; };

void set\_first(int value) { first = value; };

void set\_second(int value) { second = value; };

Time();

Time(int x, int y);

Time(const Time& obj);

operator string() const;

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

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

Time operator ++();

Time operator --();

Time operator ++(int);

Time operator --(int);

};

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

// Lab5\_1.cpp

//

#include <iostream>

#include "Time.h"

using namespace std;

Time createTime(int x, int y) {

if ((x < 0 || x >= 24) && (y < 0 || y > 60)) {

throw "Error";

}

Time t = Time(x, y);

return t;

}

void \_unexpected\_to\_bad()

{

cerr << "bad\_exception" << endl;

throw;

}

int main()

{

set\_unexpected(\_unexpected\_to\_bad);

set\_terminate(\_unexpected\_to\_bad);

Time n;

try {

cin >> n;

cout << n << endl;

cout << "Total minutes: " << n.minutes() << endl;;

}

catch (TimeException \*e)

{

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

}

catch (TimeException &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;

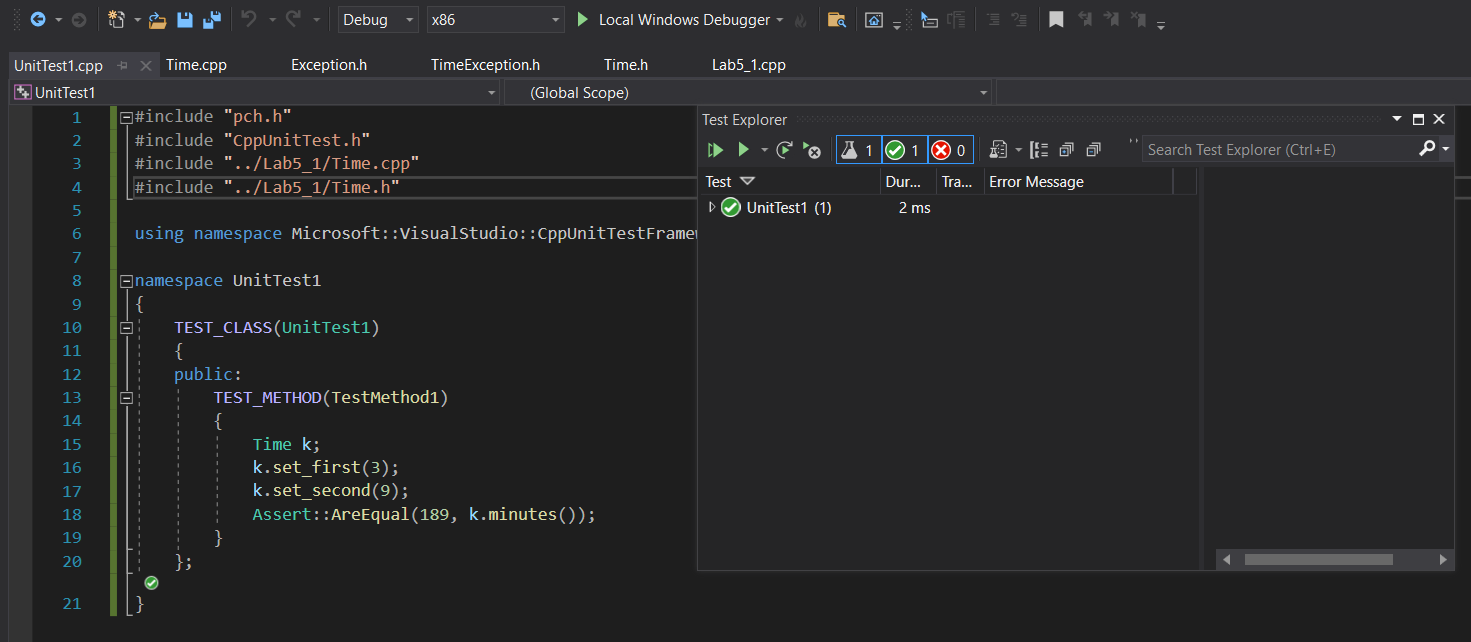
}

return 0;

}

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

**Результати unit-тесту:**

****