 МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ   
ІМЕНІ ІГОРЯ СІКОРСЬКОГО»

ФАКУЛЬТЕТ БІОМЕДИЧНОЇ ІНЖЕНЕРІЇ

КАФЕДРА БІОМЕДИЧНОЇ КІБЕРНЕТИКИ

**Компьютерный практикум №7**

з дисципліни «Об'єктно-орієнтоване програмування»

# на тему: «Виняткові ситуації»

Варіант №7

**Виконав:**

студент гр. БС-03

Затуловский Г. А.

**Перевірив:**

ас. каф. БМК Бабенко В.О.

Зараховано від \_\_\_.\_\_\_.\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

                      (підпис викладача)

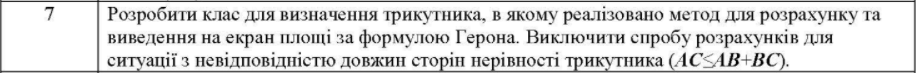
Київ-2022

**Завдання:**

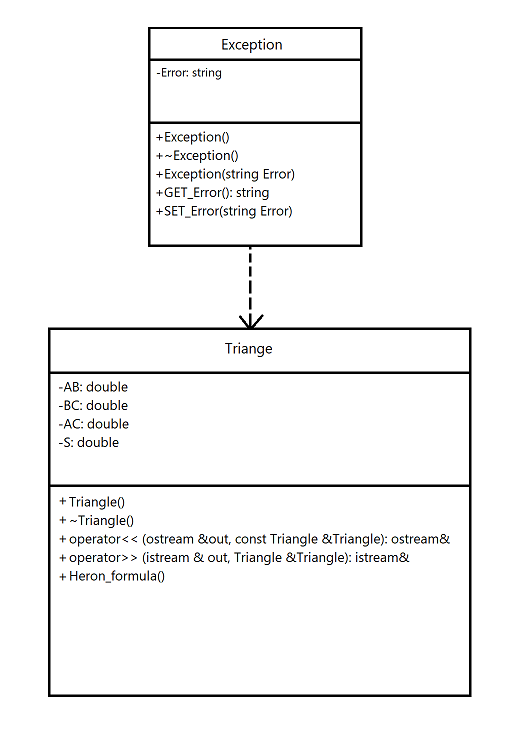
1. Ознайомитися з теоретичними вiдомостями обробки виняткових ситуацій в мові C++

2. Створити клас Exception, що дозволяє обробляти зазначену в завданнi виняткову ситуацію

3. Розробити діаграму класів в нотації UML та програмний застосунок для демонстрації роботи програми в нормальному режимi та під час виникнення виняткової ситуації відповiдно до свого варіанту (визначення класів та їх реалізації мають бути розташовані у файлах \* .h та \* срр відповідно; для ініціалізації даних класів використовувати введення з клавіатури; обробка виняткової ситуації має включати виведення на скран вiдповiдного повідомлення з текстом помилки):

**UML діаграма:**

****

**Код програми:**

**Exception.h**

#pragma once

#include <math.h>

#include <iostream>

#include <string>

using namespace std;

class Exception

{

private:

string Error;

public:

Exception();

Exception(string Error);

~Exception();

string GET\_Error();

void SET\_Error(string Error);

};

inline Exception::Exception()

{

}

inline Exception::Exception(string Error)

{

this->Error = Error;

}

inline Exception::~Exception()

{

}

inline string Exception::GET\_Error()

{

return Error;

}

inline void Exception::SET\_Error(string Error)

{

this->Error = Error;

}

**Triange.cpp**

#include "Triangle.h"

#include "Exception.h"

#include <math.h>

Triangle::Triangle()

{

AB = BC = AC = S = 0;

}

Triangle::~Triangle()

{

}

void Triangle::Heron\_formula()

{

if (AC > (BC + AB))

{

throw Exception("Incorrect input: AC > BC + AB");

}

double p = (AC + BC + AB) / 2;

cout << "p = " << p << endl;

if (AC >= p || BC >= p || AB >= p)

{

if(AC > p)

{

throw Exception("Incorrect data: AC >= p");

}

else if(BC > p)

{

throw Exception("Incorrect data: BC >= p");

}

else

{

throw Exception("Incorrect data: AB >= p");

}

}

S = sqrt(p \* (p - AC) \* (p - BC) \* (p - AB));

}

ostream& operator<<(ostream& out, const Triangle& Triangle)

{

cout << "S = " << Triangle.S << endl;

return out;

}

istream& operator>>(istream& out, Triangle& Triangle)

{

cout << "Input AB:";

cin >> Triangle.AB;

cout << "Input BC:";

cin >> Triangle.BC;

cout << "Input AC:";

cin >> Triangle.AC;

return out;

}

**Triangle.h**

#pragma once

#include <math.h>

#include <iostream>

#include <string>

#include "Exception.h"

using namespace std;

class Triangle

{

private:

double AB;

double BC;

double AC;

double S;

public:

Triangle();

~Triangle();

friend ostream& operator<< (ostream &out, const Triangle &Triangle);

friend istream& operator>> (istream & out, Triangle &Triangle);

void Heron\_formula();

};

**OOP\_7.cpp**

#include "Triangle.h"

#include <iostream>

using namespace std;

int main()

{

int choice = 1;

while (choice)

{

switch (choice)

{

case 1:

{

try

{

Triangle triangle;

cin >> triangle;

triangle.Heron\_formula();

cout << triangle;

}

catch (Exception ex)

{

cout << "Error:" << ex.GET\_Error() << endl;

}

break;

}

default:

{

cout << "Incorrect choice" << endl;

}

}

cout << "\nContinue?\nYes - 1\nNo - 0" << endl;

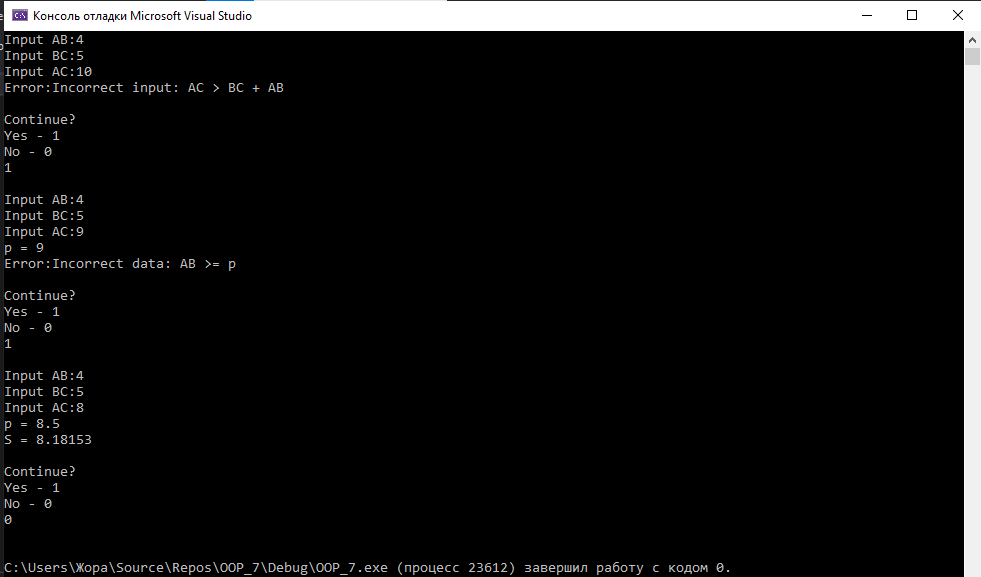
cin >> choice;

cout << endl;

}

}

**Результати:**



**Контрольні запитання**

1. **Що називається виняткової ситуацією?**

Виняткова ситуація - ситуація,виникнення якої спричиняє порушення передбаченої послідовності виконанняідля якоїумові програмування існують засоби, щодозволяють визначити,збудити,розпізнати,проігнорувати або виправитицюситуацію.

1. **Що називається обробкою виняткової ситуації?**

Обробка винятків — механізм мов програмування, призначений для обробки помилок часу виконання і інших можливих проблем (винятків), які можуть виникнути при виконанні програми.

1. **Як здійснюється обробка виняткової ситуації в С++?**

У мові C++ передбачено три ключових слова для обробки виняткових ситуацій: try (контролювати), catch (ловити) і throw (генерувати, породжувати, кидати, посилати, формувати). не знайдений, викликається функція terminate() для завершення програми.

1. **Чи можна помістити блок try у функцію, залишивши блок catch для обробки виняткових ситуацій у функції main() ?**

Це не можливо. Програма автоматично закриється та видасть помилку

1. **До чого призведення передавання винятку в блок catch за посиланням ?**

Передавання винятку в блок catch за посиланням не буде створювати копію.

1. **Для чого використовують вираз throw, що не містить операдів?**

Вираз throw, який не містить опердантів використовується для повторного виключення.

1. **До чого призведе генерація похідної виняткової ситуації за умови наявності блоку catch, що призначений для обробки базової виняткової ситуації ? Відповідь пояснити?**

Дочірній виняток буде перехоплений базовим catch.

1. **Назвіть і опишіть стандартні типи винятків в бібліотеці std?**

logic\_error. До категорії логічних помилок належать помилки, які (принаймні теоретично) можна запобігти, наприклад, додаткової перевіркою аргументів функції. Зокрема, до логічних помилок належить порушення логічних передумов чи інваріанту класу

Винятки мовної підтримки використовуються на рівні мови C++, тому їх правильніше було б віднести до базової мови, ніж до бібліотеки. Етн винятки генеруються при невдалих спробах виконання деяких операцій

Винятки, похідні від runtime\_error, повідомляють про події, які не контролюються програмою. У стандартну бібліотеку C++ включено такі класи помилок часу виконання:

1. **Для чого під час обробки виняткової ситуації використовують призначені функції terminate(), unexpected(), abort() .**

Функція terminate() надає механізм часу виконання, який сповіщає користувача, що відповідного обробника не знайшлося.

Функція unexpected() викликається середовищем виконання C++, коли порушується специфікація динамічного виключення: виняток видається з функції, специфікація виключення якої забороняє виключення цього типу. unexpected() також може бути викликаний безпосередньо з програми.

Функція abort() викликає негайне припинення програми. Чищення буферів файлів не виконується. Функція повертає викликає процесу значення 3 (зазвичай операційній системі).

**10. Навести приклад коду, за допомогою якого можна вивести на екран текстове повідомлення з номером строчки, під час виконання якої сталася виняткова ситуація, та назвою файла, що містить цю стрiчку.**

#include <fstream>

#include <string>

#include <iostream>

using namespace std;

class

{

public:

string c;

int a;

};

int main()

{

ifstream fin;

fin.exceptions(ifstream::badbit | ifstream::failbit);

string path = "WrongFile.txt";

try

{

fin.open(path);

}

catch (const ifstream::failure &error)

{

cout << "row is" << 20 << "\nname file"<<path <<error.what();

cout << error.code();

}

}