**Лабораторна №9**

**Тема:** Вивчення механізмів оброблення виняткових ситуацій.

**Мета:** навчитись обробляти ситуації появлення виняткових ситуацій, вивчити механізми їх оброблення.

Завдання 1.

Додайте клас винятків до програми ARROVER таким чином, щоб індекси, що виходять за межі масиву, викликали генерацію винятку. Блокпастка catch може виводити користувачеві повідомлення про помилку.

//////program ARROVER////// #include using namespace std; #include //for exit() const int LIMIT = 100; //array size

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

class safearay { private: int arr[LIMIT]; public: int& operator [](int n) { if( n< 0 || n>=LIMIT ) { cout << "\nIndex out of bounds"; exit(1); } return arr[n]; } }; ////////////////////////////////////////////////////////////////

int main() { safearay sa1; for(int j=0; j<< "Element " << j << " is " << temp << endl; } return 0; } \

3. Реалізуйте програму згідно завдання 2.

Завдання 2. Модифікуйте програму з попереднього завдання таким чином, щоб в повідомлення про помилку входила інформація про значення індексу, який призвів до збою.

4. Модифікуйте текст програми, розробленої згідно індивідуального завдання лабораторної роботи №3, добавивши до нього (в місця можливого виникнення помилок) процедури оброблення виняткових ситуацій, які будуть генерувати об’єкти класу, який міститиме такі атрибути: місце виникнення помилки; значення, яке призвело до помилки; параметризований конструктор; перевизначену операції виводу, яка виводитиме значення на екран (або зберігатиме виняткову ситуацію у файл).

//#include <iostream>

//using namespace std;

//class asd {

//public:

// int asd1;

// asd() {

// asd1 = 1;

// }

// asd(int a2) {

// try

// {

// asd1 = a2;

// }

// catch (const std::exception&)

// {

// cout << "Введене значення не є числом";

// }

//

// }

// void operator + (int a1) {

// try

// {

// this->asd1 = a1 + this->asd1;

// }

// catch (const std::exception&)

// {

// cout << "Якийсь із доданків не є числом";

// }

// }

// void operator = (asd a1) {

// try

// {

// this->asd1 = a1.asd1;

// }

// catch (const std::exception&)

// {

// cout << "Якесь значення не є числом";

// }

//

// }

// void Output() {

// std::cout << asd1;

// }

//

//};

//

//int main()

//{

// try

// {

// asd\* mas1 = new asd(3);

// asd\* rez = new asd(0);

// \*mas1 + 2;

// rez = mas1;

// rez->Output();

// }

// catch (const std::exception&)

// {

// cout << "Проблеми при використанні конструктора";

// }

//

//

//}

//////program ARROVER//////

#include <iostream>

using namespace std;

#include <process.h> //for exit()

const int LIMIT = 100; //array size

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

class safearay

{

private:

int arr[LIMIT];

public:

int& operator [](int n)

{

if (n < 0 || n >= LIMIT)

{

cout << "\nIndex out of bounds"; exit(1);

}

return arr[n];

}

};

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

int main()

{

try

{

safearay sa1;

for (int j = 0; j < LIMIT; j++)

sa1[j] = j \* 10; //\*left\* side of equal sign

for (int j = 0; j < LIMIT; j++) //display elements

{

int temp = sa1[j]; //\*right\* side of equal sign

cout << "Element " << j << " is " << temp << endl;

}

return 0;

}

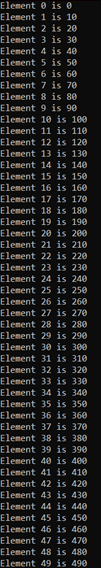
catch (int j)//const std::exception&)

{

cout << "\nIndex out of bounds"; exit(1);

}

}



Висновок: я навчився обробляти ситуації появлення виняткових ситуацій, вивчити механізми їх оброблення