

Лабораторна робота №9

Тема: Вивчення механізмів оброблення виняткових ситуацій.

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

Завдання 1

Завдання 1. Додайте клас винятків до програми ARROVER таким чином, щоб індекси, що виходять за межі масиву, викликали генерацію винятку. Блок-пастка catch може виводити користувачеві повідомлення про помилку.

КОД ПРОГРАМИ

```
#include <iostream>
using namespace std;
#include <process.h>
const int LIMIT = 10;
const int sizeArray = 5;

class ArrayException : public exception
{
public:
    void what()
    {
        cout << "Type error is index out of bounds";
    }
};

class safearray
{
private:
    int arr[LIMIT];
public:
    int& operator [](int n)
    {
        if (n < 0 || n >= sizeArray - 1)
```

```


        {
            cout << "\nIndex out of bounds"; exit(1);
        }
        return arr[n];
    }
};

int main()
{
    safearray sa1;
    try {
        for (int j = 0; j < LIMIT; j++) {
            if (j >= sizeArray - 1)
            {
                ArrayException a;
                throw a;
            }
            else
            {
                sa1[j] = j * 10;
            }
        }
        for (int j = 0; j < LIMIT; j++)
        {
            if (j >= sizeArray - 1)
            {
                ArrayException a;
                throw a;
            }
            else {
                int temp = sa1[j];
                cout << "Element " << j << " is " << temp << endl;
            }
        }
    }
    catch (ArrayException a)
    {
        a.whayt();
    }

    return 0;
}

```

РЕЗУЛЬТАТ ВИКОНАННЯ ПРОГРАМИ

 Microsoft Visual Studio Debug Console

```

Type error is index out of bounds
C:\Users\ASUS\source\repos\ConsoleApplication32\

```

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.

КОД ПРОГРАМИ

```
#include <iostream>
using namespace std;
#include <process.h>
const int LIMIT = 10;
const int sizeArray = 5;

class ArrayException : public exception
{
private:
    int errorIndex;
public:
    ArrayException(int errorIndex)
    {
        this->errorIndex = errorIndex;
    }

    void what()
    {
        cout << "ERROR!!!!!! Type error is index out of bounds" << endl;
        cout << "Index array - " << errorIndex << endl;
    }
};

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

int main()
{
    safearray sa1;
    try {
        for (int j = 0; j < LIMIT; j++) {
```

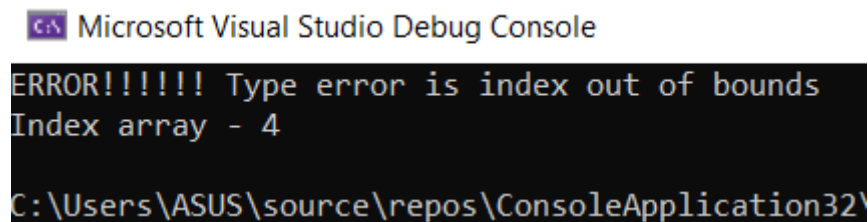
```

        if (j >= sizeArray - 1)
        {
            ArrayException a(j);
            throw a;
        }
        else
        {
            sa1[j] = j * 10;
        }
    }
    for (int j = 0; j < LIMIT; j++) {
        if (j >= sizeArray - 1)
        {
            ArrayException a(j);
            throw a;
        }
        else {
            int temp = sa1[j];
            cout << "Element " << j << " is " << temp << endl;
        }
    }
}
catch (ArrayException a)
{
    a.what();
}

return 0;
}

```

РЕЗУЛЬТАТ ВИКОНАННЯ ПРОГРАМИ



Microsoft Visual Studio Debug Console

```

ERROR!!!!!! Type error is index out of bounds
Index array - 4

C:\Users\ASUS\source\repos\ConsoleApplication32\

```

Завдання 3

4. Модифікуйте текст програми, розробленої згідно індивідуального завдання лабораторної роботи №3, додавши 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).

КОД ПРОГРАМИ

```
#include <iostream>
#include <math.h>
using namespace std;

class negativeTime : public exception
{
public:
    void what()
    {
        cout << "Error!!!!!! Time cannot be negative" << endl;
    }
};

class Date
{
private:
    int day, month, year;
public:
    Date() {}
    Date(int day, int month, int year)
    {
        this->day = day;
        this->month = month;
        this->year = year;
    }
    int getDay() { return day; }
    int getMonth() { return month; }
    int getYear() { return year; }
    void setDay(int day) { this->day = day; }
    void setMonth(int month) { this->month = month; }
    void setYear(int year) { this->year = year; }
    Date operator +(Date& date)
    {
        Date temp;
        try
        {
            if (this->getDay() + date.getDay() < 0 || this->getMonth() +
date.getMonth() < 0 || this->getYear() + date.getYear() < 0)
            {
                negativeTime nT;
                throw nT;
            }
        }
    }
};
```

```

        else
        {
            temp.setDay(this->getDay() + date.getDay());
            temp.setMonth(this->getMonth() + date.getMonth());
            temp.setYear(this->getYear() + date.getYear());
        }
    }
    catch (negativeTime e)
    {
        e.what();
        exit(1);
    }
    return temp;
}
Date& operator -(Date& date)
{
    Date temp;
    try
    {
        if (this->getDay() - date.getDay() < 0 || this->getMonth() -
date.getMonth() < 0 || this->getYear() - date.getYear() < 0)
        {
            negativeTime nT;
            throw nT;
        }
        else
        {
            temp.setDay(this->getDay() - date.getDay());
            temp.setMonth(this->getMonth() - date.getMonth());
            temp.setYear(this->getYear() - date.getYear());
        }
    }
    catch (negativeTime e)
    {
        e.what();
        exit(1);
    }
    return temp;
}
bool operator <(Date& date)
{
    if (this->getDay() + this->getMonth() * 31 + this->getYear() * 365 <
        date.getDay() + date.getMonth() * 31 + date.getYear() * 365)
        return true;
    else
        return false;
}
bool operator >=(Date& date)
{
    if (this->getDay() + this->getMonth() * 31 + this->getYear() * 365 >=
        date.getDay() + date.getMonth() * 31 + date.getYear() * 365)
        return true;
    else
        return false;
}
~Date() {}
};
ostream& operator <<(ostream& s, Date& date)

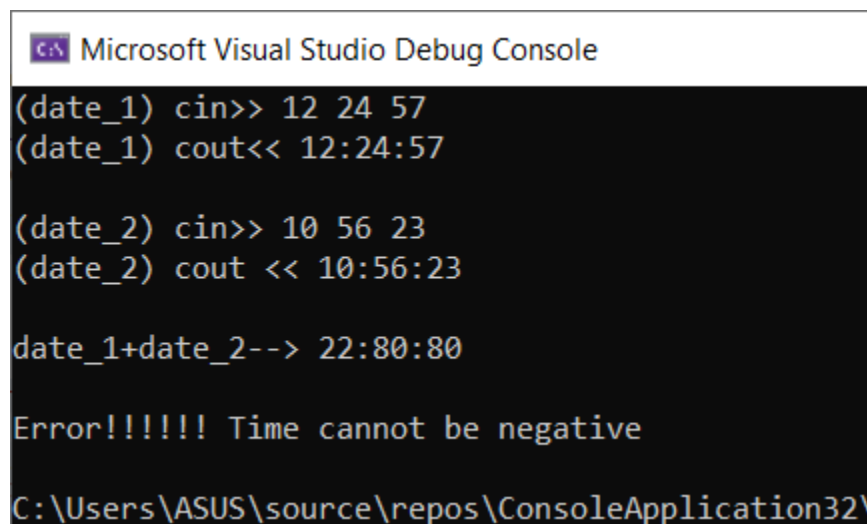
```

```

{
    s << date.getDay() << ":" << date.getMonth() << ":" << date.getYear() << endl;
    return s;
}
istream& operator >>(istream& i, Date& date)
{
    int day, month, year;
    cin >> day; date.setDay(day);
    cin >> month; date.setMonth(month);
    cin >> year; date.setYear(year);
    return i;
}
int main()
{
    cout << "(date_1) cin>> ";
    Date date1;
    cin >> date1; cout << "(date_1) cout<< ";
    cout << date1; cout << endl;
    Date date2;
    cout << "(date_2) cin>> ";
    cin >> date2; cout << "(date_2) cout << ";
    cout << date2;
    Date date3 = date1 + date2;
    cout << endl << "date_1+date_2--> " << date3 << endl;
    Date date4 = date1 - date2;
    cout << endl << "date_1-date_2--> " << date4 << endl;
    bool l = date1 < date2;
    bool m = date1 >= date2;
    if (l == 1) cout << "date_1 < date_2 --> true" << endl;
    else cout << "date_1 < date_2 --> false" << endl;
    if (m == 1) cout << "date_1 >= date_2 --> true" << endl;
    else cout << "date_1 >= date_2 --> false" << endl;
    return 0;
}

```

РЕЗУЛЬТАТ ВИКОНАННЯ ПРОГРАМИ



```

Microsoft Visual Studio Debug Console

(date_1) cin>> 12 24 57
(date_1) cout<< 12:24:57

(date_2) cin>> 10 56 23
(date_2) cout << 10:56:23

date_1+date_2--> 22:80:80

Error!!!!!! Time cannot be negative

C:\Users\ASUS\source\repos\ConsoleApplication32\

```

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