

Лабораторно упражнение № 8

Обработка на изключения в C#

I. Теоретична част

1. Възникване и обработка на изключителни събития (изключения)

Изключенията са събития, които настъпват при изпълнението на програмен код в резултат на възникване на извънредни ситуации. Например, опит да бъде отворен файл, който не съществува; въвеждане на данни от клавиатурата, който се са в очаквания формат; деление на нула и всякакви други подобни ситуации. При възникване на такова събитие, системата прекратява изпълнението на програмния код.

Технологията .NET предлага единен подход за обработката на изключения. Обработката на изключителни събития има за цел да не се прекратява изпълненото на програмния код при възникване на такова събитие, а да се реагира по подходящ начин, след което да се продължи с нормалното изпълнение.

2. Блок try-catch

Обектно-ориентираното програмиране предлага структурирано решение на проблемите, свързани с обработка на грешки чрез използването на блок try-catch. Идеята е физически да се раздели съществената част от алгоритъма, реализирана чрез съответните оператори и конструкции, които отговарят за нормални поток на обработка, от останалите оператори и конструкции, които обслужват обработката на грешки. Секцията на кода, който би предизвикал изключение се поставя в блок try, а кодът, който обработва изключение – в блок catch

```
try
{
    // тяло на блок try
}
catch (ИмеКласИзключение идентификатор)
{
    // тяло на блок catch
}
```

Задължително ИмеКласИзключение е от тип Exception, SystemException или производен клас. Идентификаторът е името на обекта изключение, което се използва в блок catch.

В даден блок try може да има повече от една клаузи catch. В този случай се прихващат повече от едно изключения, които би могъл да предизвика програмният код в тялото на секция try.

```
try
```

```

{
    // тяло на блок try
}
catch (ИмеКласИзключение идентификатор)
{
    // тяло на блок catch
}
catch (ИмеКласИзключение идентификатор)
{
    // тяло на блок catch
}

```

В блок try може да има секция catch, която е с общо предназначение.

```

try
{
    // тяло на блок try
}
catch
{
    // тяло на блок catch
}

```

Клауза catch с общо предназначение може да прихване всяко изключение независимо от неговия тип.

В конструнцията try-catch може да има блок finally, в който се обособяват действия, които винаги се изпълняват независимо от това дали блок try се изпълнява нормално или настъпва изключително събитие и се изпълняват действията, предвидени в клауза catch. Блок finally се използва за да предодврати дублирането на програмен код.

3. Предизвикване на изключения

Освен за обработката на системни грешки, блок try-catch може да се използва и за обработка на изключения, които са предизвикани от програмиста. Например, ако стойностите на дадена променлива трябва да са в определен интервал, програмистът може да предизвика изключение, ако въведената стойност за нея излиза извън интервала.

Предизвикването на изключение е чрез конструкция throw. Когато в кода се срещне throw, незабавно се преустановява нормалното изпълнение на програмата и управлението се прехвърля към подходящия catch блок, който да обработи възникналото изключение.

Общият вид на конструкцията е:

throw Изключение;

Пример:

```
try
```

```

{
    Console.WriteLine("Enter a minute:");
    int minute = Int32.Parse(Console.ReadLine());
    if(minute<1 || minute<=60)
    {
        string fault= minute+ " is not a valid minute";
        throw new Exception(fault);
    }
}
Catch(Exception e)
{
    Console.WriteLine(e.Message);
}

```

В посочения пример се предизвиква изключение, ако въведената стойност за минутата не е в интервала [1,60].

II. Задачи за изпълнение:

1. Да се въведат две цели числа от клавиатурата и да се намери резултатът от делението им. Да се обработят изключенията, които евентуално биха възникнали при изпълнението на програмния код. Да се изведе текст „God bye!” след завършване изпълнението на програмата.
2. Да се въведе число от клавиатурата и да се изведе корен квадратен от въведената стойност. Да се предизвика изключение, ако стойността е по-малка или равна на нула като се изведе съобщение „Invalid number”. Да се изведе низ „End of program” след приключване изпълнението на програмния код.
3. Да се напише метод, който извежда име на месец при въвеждане на номера на съответния месец. Кодът да генерира изключение, ако въведената стойност не отговаря на месец.
4. Да се декларира клас, описващ време с час, минута и секунда. В класа да се съдържа метод, който въвежда стойности за полетата. Кодът на метода да генерира изключение при въвеждане на невалидни стойности за полетата.

Относно въпроси по темата на адрес: ln_zh_st@yahoo.com