Управљање грешкама и изузецима

Писање кода подразумева велики увид и предзнање како ће крајња апликација функционисати, али и поред тога, грешке ће се појавити и током кодирања и у свим фазама израде апликације.
Нежељено понашање програма ће се десити из раличитих разлога, од којих је већина изван контроле програмера.
Апликације морају бити спремне за откривање нежељеног начина рада и како да се поступа у таквим случајевима: или превазилажењем могућих проблема или пријављивањем кориснику разлога због којих је дошло до грешке у раду.

Грешке и хватање изузетака

Корисник апликације се понаша на непредвидив начин, хард дискови отказују, друге апликације на истом рачунару праве пометњу, недостатак оперативне меморије, губитак везе са мрежом, атмосферске прилике, нестанак струје; све то може довести до грешака у раду апликације.
Поставља се питање како да се открију и како да се отклоне такве и сличне грешке.
Да би се управљало грешкама, савремени програмски језици користе изузетке (exceptions).

Ручно додавање кода када се деси нека грешка захтева време, није ефикасна метода и није могуће реаговати на све могуће грешке на исти начин.
У C# се користи раздвајање кода за обраду грешака од основног кода, коришћењем изузетака и процедура за обраду изузетака.
За писање програма који обрађују изузетке, потребно је:

1. Написати код унутар блока try; блок се покрене и ако ниједан исказ не генерише изузетак блок се регуларно завршава; ако се деси грешка, излази се из блока try и прелазу у други део кода предвиђен за хватање и обраду изузетака – процедура catch
2. Написати једну или више процедура catch, непосредно после блока try за обраду свих могућих грешака; свака од процедура реагује на одређену врсту грешака

Пример:

```
try
{
     int levo = int.Parse(levooperand.Text);
     int desno = int.Parse(desnooperand.Text);
     int odgovor = Racunanje(levo, desno);
     rezultat.Text = odgovor.ToString();
}
catch (FormatException izuzetak)
{
     //obrada izuzetaka
}
```

У примеру је блок try који обухвата код који покушава да низ знакова, које је корисник унео, претвори у целобројне вредности.
Код позива методу за рачунање вредности и уписује резулатат.

Ако низ знакова обухвата и неки неважећи знак, метода int.Parse избацује изузетак FormatException и извршавање се преноси на одговарајућу процедуру catch; такође, промењива izuzetak се попуњава објектом који садржи детаље о изузетку.
Када се процедура заврши, програм наставља са овим исказом после процедуре.
Ако ниједна процедура не одговора изузетку, за тај изузетак се каже да је необрађен.

Тип FormatException има бројна својства која се могу користити за одређивање тачног узрока изузетка и већина тих својстава су заједничка за све изузетке.
Нпр, својство Massage садржи текстуални опис грешке која је изазвала изузетак.
Та информација се може искористити при обради изузетака, нпр записивањем детаља у log датотеку са записима о понашању програма или тако што се кориснику прикаже порука.

Необрађени изузеци
Шта се дешава ако блок try избаци изузетак а не постоји одговарајућа процедура catch?
У претходном примеру, било би могуће да levooperand садржи низ знакова који представљају исправан цео број, али је тај број изван опсега важећих који подржава C# (нпр “2147483648“).
У том случају исказ int.Parse би избацио изузетак OverflowException koji procedura catch типа FormatException не би ухватила.
Метода у којој се налази блок try се одмах прекида и извршавање се враћа до методе која је позвала ову методу и цео поступак се понавља.
Ако после прескакања уназад кроз све позивајуће методе, извршни модул не може да пронађе одговарајућу процедурусatch, програм се прекида са необрађеним изузетком.
Извршавање се наставља након што се изузетак ухвати у методи која садржи процедуру catch која је ухватила изузетак; ако се изузетак десио у другој методи, а не у онај која садржи процедуру catch, контрола се не враћа до методе која је изазвала изузетак.

Изузетке које генерише апликација можемо лако испитати (мени Debug->Start Debugging).
Појавом изузетка, појављује се оквир за дијалог, зауставља се апликација на исказу који је изазвао изузетак и враћа програмера у програм за отклањање грешака.

Вишеструке процедуре за отклањање грешака
Пример:

```
try
{
     int levo = int.Parse(levooperand.Text);
     int desno = int.Parse(desnooperand.Text);
     int odgovor = Racunanje(levo, desno);
     rezultat.Text = odgovor.ToString();
}
catch (FormatException izuzetak1)
{
     //obrada izuzetaka
}
catch (OverflowException izuzetak2)
```

```
{
     //obrada izuzetaka
}
```

Ако код у блоку try избаци изузетак FormatException, покрећу се искази у блоку catch за изузетак FormatException.
Ако код у блоку try избаци изузетак OverflowException, покрећу се искази у блоку catch за изузетак OverflowException.
Ако код у блоку catch за изузетак FormatException генерише изузетак OverflowException, тиме се не покреће блок catch за суседан изузетак OverflowException.
Уместо тога, изузетак се преноси до методе која је позвала овај код.
Пример: Код којим се покушава делити са 0.

```
using System;
public class Proba
{
    public static void Main()
    {
        int a = 10;
        int b = 0;
        int x = a / b;
        Console.WriteLine("Ostatak koda.");
    }
}
```

Даје: Unhandled Exception: System.DivideByZeroException: Attempted to divide by zero.
Код се стартује и не приказује очекивани излаз због појаве необрађеног изузетка изазваног грешком покушаја дељења са 0.
Ако се код модификује са блоковима try и catch:

```
using System;
public class Proba
{
    public static void Main()
    {
        try
        {
            int a = 10;
            int b = 0;
            int x = a / b;
        }
        catch (Exception izuzetak)
        {
            Console.WriteLine(izuzetak);
        }

        Console.WriteLine("Ostatak koda.");
    }
```

}
Даје:System.DivideByZeroException: Attempted to divide by zero.
  at Proba.Main() in C:\Users\nera\source\repos\ProjekatCS001\ProjekatCS001\Program.cs:line 10
Ostatak koda.

Сада опет долази до покушаја дељења са 0, али не долази до прекида програма.
Изузетак је ухваћен и обрађен тако што се са блоком catch припремила ситуација реакције кода на недозвољено дељење са 0.

Задаци за самосталан рад

1. а) Написати програм којим се омогућава да корисник унесе цео број а на излазу се добија тај број и његов квадрат.

   б) Модификовати програм тако да помоћу блокова try и catch на екрану се исписује порука да је дошло до грешке.

   в) Модификовати catch блок са Exception параметром којим се хвата било који тип изузетака

2. Написати програм који реагује на унос броја који се дели са бројем 100, и који приказује различите поруке у зависности од формата унетог броја.