

OOP

Obrada iznimaka

U ovom poglavlju naučit ćete

- Izbacivanje i hvatanje iznimki: try - catch
- Iskaz finally
- Objekti klase Exception
- Prilagođene iznimke
- Ponovno izbacivanje iznimki (throw)

MI1 - summary

OOP	2017	2018	2019	2020	2021	2022	2023	2024
Broj studenata	79	77	108	109	112	111	118	114
Broj izašlih na MI1	55	49	77	80	81	85	83	92
Postotak izašlih na M1	69,62%	63,64%	71,30%	73,40%	72,32%	76,58%	70,33%	80,70%
Prosjek bodova izašlih (od 55 mogućih)	44,68	46,29	44,56	46,54	46,63	46,03	45,13	47,93

Iznimke

- Kao i mnogi drugi objektno orijentirani jezici i programski jezik C# pogreškama i neuobičajenim uvjetima upravlja s pomoću *iznimki*.
- Iznimka (engl. *exception*) je objekt u kojem su ućahurene informacije o neobičnoj pojavi u programu.

Klasa Exception

- U programskom jeziku C# kao iznimke se mogu izbacivati samo objekti klase System.Exception ili objekti koji su izvedeni iz tog tipa
- U imenskom prostoru System možemo koristiti razne tipove iznimki kao što su InvalidCastException, FormatException i mnogi drugi).

Primjer try-catch:

```
using System;

namespace IskazCatch
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Ulaz u Main...");
            f1();
            Console.WriteLine("Izlaz iz Main.");
        }
        static void f1()
        {
            Console.WriteLine("Ulaz u f1...");
            f2();
            Console.WriteLine("Izlaz iz f1.");
        }
        static void f2()
        {
            Console.WriteLine("Ulaz u f2...");
            try
            {
                Console.WriteLine("Ulaz u try blok...");
                throw new System.Exception();
                Console.WriteLine("Izlaz iz try bloka...");
            }
            catch
            {
                Console.WriteLine("Iznimka uhvaćena i obrađena.");
            }
            Console.WriteLine("Izlaz iz f2.");
        }
    }
}
```

finally

- **U nekim situacijama izbacivanje iznimke može uzrokovati problem.**
 - Npr. otvaranje datoteke ili zauzimanje nekog dijeljenog resursa....može nam biti važna mogućnost zatvaranja datoteke ili pražnjenja privremene memorije.
- **Izvođenje te akcije, bez obzira na izbacivanje iznimke, možemo osigurati koristeći finally blok.**

Finally..struktura

```
Osoba o;  
try  
{  
    // Inicijalizacija objekta unutar try bloka  
    o = new Osoba();  
}  
catch  
{  
    // objekt o sada možemo dohvatiti i testirati njegovo stanje  
}  
finally  
{  
    // objekt o sada možemo dohvatiti i otpustiti njegove eventualne "skupe resurse"  
}
```


Hvala na pažnji!