Fatlum Maliqi / Kampusi Ferizaj

Chapter 12. Exception Handling

1. Find out all exceptions in the System.IO.IOException hierarchy.

```
using System;

namespace ex1
{
    class Program
    {
        public class IOException : SystemException { }
        static void Main(string[] args)
        {
            try
            {

            }
            catch (IOException e)
            {
                Console.WriteLine(
                    "{0}: The write operation could not " +
                    "be performed because the specified " +
                    "part of the file is locked.",
                    e.GetType().Name);
            }
            Console.ReadKey();
        }
    }
}
```

2. Find out all standard exceptions that are part of the hierarchy holding the class System.IO.FileNotFoundException.

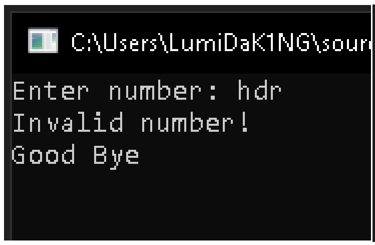
- public FileNotFoundException (string? message, string? fileName, Exception? innerException);
- public class FileNotFoundException : System.IO.IOException
- public FileNotFoundException (string? message, string? fileName);
- public FileNotFoundException (string? message, Exception? innerException);
  protected FileNotFoundException (System.Runtime.Serialization.SerializationInfo info, System.Runtime.Serialization.StreamingContext context);
- public FileNotFoundException (string? message);
- public FileNotFoundException ();

3. Find out all standard exceptions from System.ApplicationException hierarchy.

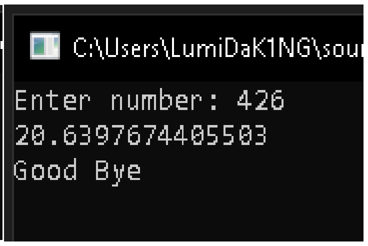
- public ApplicationException (string? message);
- protected ApplicationException (System.Runtime.Serialization.SerializationInfo info,
- System.Runtime.Serialization.StreamingContext context);
  public ApplicationException (string? message, Exception? innerException);

4. Explain the concept of exceptions and exception handling, when they are used and how to catch exceptions.

```
using System;

namespace ex4
{
   class Program
   {
      static void Main(string[] args)
      {
         Console.Write("Enter number: ");
         string input = Console.ReadLine();
         int n = -1;
         bool invalidNumber = false;

         try
         {
            n = Convert.ToInt32(input);
         }
         catch (FormatException e)
         {
            Console.WriteLine("Invalid number!");
            invalidNumber = true;
         }
         finally
         {
            if (n < 0)
            {
               if (!invalidNumber) Console.WriteLine("Invalid number!");
            }
            else Console.WriteLine(Math.Sqrt(n));
         }

         Console.WriteLine("Good Bye");
         Console.ReadKey();
      }
   }
}
```



```
C:\Users\LumiDaK1NG\sour
Enter number: hdr
Invalid number!
Good Bye
```



```
C:\Users\LumiDaK1NG\sou
Enter number: 426
20.6397674405503
Good Bye
```

5. Explain when the statement try-finally is used. Explain the relationship between the statements try-finally and using.

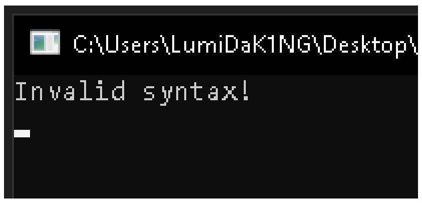
Duke përdorur një bllok më në fund, ju mund të pastroni të gjitha burimet që janë alokuar në një bllok try dhe mund të ekzekutoni kodin edhe nëse ndodh një përjashtim në bllokun try. Në mënyrë tipike, deklaratat e një blloku të ekzekutuar përfundimisht kur kontrolli lë një deklaratë try. Transferimi i kontrollit mund të ndodhë si rezultat i ekzekutimit normal, të ekzekutimit të një deklarate pushimi, vazhdimi, shkimi ose kthimi, ose përhapjes së një përjashtimi nga deklarata try.

6. Explain the advantages when using exceptions.

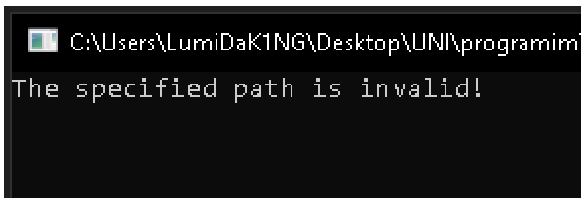
exceptions ju mundësojnë të shkruani rrjedhën kryesore të kodit tuaj dhe të merreni me raste të jashtëzakonshme diku tjetër. ... 2: Përhapja e gabimeve deri në pirgun e thirrjeve: - Një avantazh i dytë i exceptions është aftësia për të përhapur gabimin në raportimin e metodave të grupit të thirrjeve.

7. Write a program that takes a positive integer from the console and prints the square root of this integer. If the input is negative or invalid print "Invalid Number" in the console. In all cases print "Good Bye".

```
using System;

namespace ex7
{
   class Program
   {
      static void Main(string[] args)
      {
         try
         {
            using (StreamReader sr = new StreamReader(@"C:\Users\\Desktop\text.txt"))
            {
               String line = sr.ReadToEnd();
               Console.WriteLine(line);
            }
         }
         catch (FileNotFoundException e)
         {
            Console.WriteLine(e.Message);
         }
         catch (DirectoryNotFoundException e)
         {
            Console.WriteLine("The specified path is invalid!");
         }
         catch (IOException e)
         {
            Console.WriteLine("Invalid syntax!");
         }
         Console.ReadKey();
      }
   }
}
```



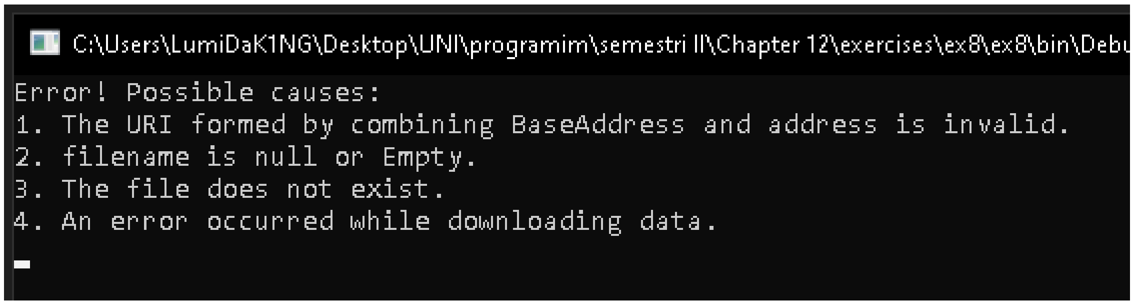
```
C:\Users\LumiDaK1NG\Desktop\
Invalid syntax!
```



```
C:\Users\LumiDaK1NG\Desktop\UNI\programim
The specified path is invalid!
```

8. Write a method ReadNumber(int start, int end) that reads an integer from the console in the range [start…end]. In case the input integer is not valid or it is not in the required range throw appropriate exception. Using this method, write a program that takes 10 integers a1, a2, …, a10 such that 1 < a1 < … < a10 < 100.

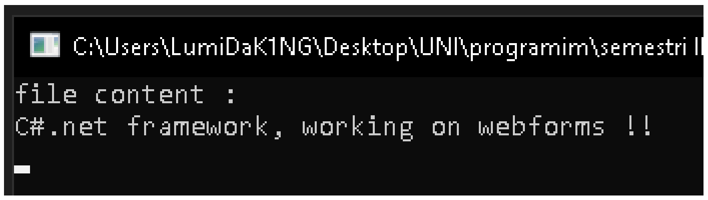
```
using System;
using System.Net;

namespace ex8
{
   class Program
   {
      static void Main(string[] args)
      {
         WebClient Client = new WebClient();

         try
         {
            Client.DownloadFile("http://3.bp.blogspot.com/-
qXtmJRAlJcA/U413iy_YzKI/AAAAAAAAOn8/Ajr4B8h9TcE/s1600/google-logo-high-res.png",
@"C:\Users\Ivan\Desktop\image.png");
         }
         catch (ArgumentException)
         {
            Console.WriteLine("The address or fileName parameter is null!");
         }
         catch (WebException)
         {
            Console.WriteLine("Error! Possible causes:\n1. The URI formed by combining
BaseAddress and address is invalid.\n2. filename is null or Empty.\n3. The file does not
exist.\n4. An error occurred while downloading data.");
         }
         catch (NotSupportedException)
         {
            Console.WriteLine("The method has been called simultaneously on multiple threads.");
         }
         Console.ReadKey();
      }
   }
}
```



```
C:\Users\LumiDaK1NG\Desktop\UNI\programim\semestri II\Chapter 12\exercises\ex8\ex8\bin\Debu
Error! Possible causes:
1. The URI formed by combining BaseAddress and address is invalid.
2. filename is null or Empty.
3. The file does not exist.
4. An error occurred while downloading data.
```

9. Write a method that takes as a parameter the name of a text file, reads the file and returns its content as string. What should the method do if and exception is thrown?

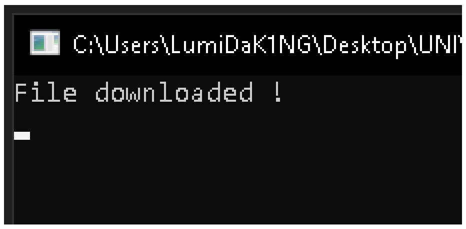
```
using System;

namespace ex9
{
   class Program
   {
      static void GetName(string name)
      {
         string newName = Console.ReadLine();
         return;
      }
      static void Main(string[] args)
      {
         Console.WriteLine("file content : ");
         try
         {
            Console.WriteLine("C#.net framework, working on webforms !!");
         }
         catch
         {
            Console.WriteLine("File couldnt replace");
         }
         Console.ReadKey();
      }
   }
}
```



```
C:\Users\LumiDaK1NG\Desktop\UNI\programim\semestri I
file content :
C#.net framework, working on webforms !!
```

10. Write a program that downloads a file from Internet by given URL, e.g. https://softuni.bg/forum.

```
using System;

namespace ex13
{
  class Program
  {
    static void Main(string[] args)
    {
      var client = new System.Net.WebClient();
      client.DownloadFile(url, "c:\\path-to-save-file-on-server\\test.pdf");

      Console.WriteLine("File downloaded !");
      Console.ReadKey();
    }
  }
}
```


C:\Users\LumiDaK1NG\Desktop\UNI

```
File downloaded !
```