****

# Chapter 12

**Prof:Muzafer Shala**

**Ass:Laberion Zebica Student:Albion Burrniku**

**Kampusi:Ferizaj**

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

https://docs.microsoft.com/en-us/dotnet/api/system.io.ioexception?redirectedfrom=MSDN&view=net-5.0

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

https://docs.microsoft.com/en-us/dotnet/api/system.io.filenotfoundexception?redirectedfrom=MSDN&view=net-5.0

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

https://docs.microsoft.com/en-us/dotnet/api/system.applicationexception?redirectedfrom=MSDN&view=net-5.0

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

using System;

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");

}

}

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

using System;

class Program

{

static void ReadNumber(int start, int end)

{

int count = 1, number;

do

{

Console.Write("Number{0}: ", count);

number = Int32.Parse(Console.ReadLine());

if (number >= end || number <= start)

{

Console.WriteLine("Wrong input!");

break;

}

else

start = number;

count++;

} while (count < 11);

}

static void Main(string[] args)

{

Console.Write("Start: ");

int start = Int32.Parse(Console.ReadLine());

Console.Write("End: ");

int end = Int32.Parse(Console.ReadLine());

if (end <= start + 10)

Console.WriteLine("Wrong input");

else

ReadNumber(start, end);

}

}

6.    Explain the **advantages** when using exceptions.

using System;

using System.IO;

class Program

{

static void Main(string[] args)

{

try

{

using (StreamReader sr = new StreamReader("TestFile.txt"))

{

String line = sr.ReadToEnd();

Console.WriteLine(line);

}

}

catch (Exception e)

{

Console.WriteLine("The file could not be read:");

Console.WriteLine(e.Message);

}

}

}

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;

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");

}

}

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;

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.");

}

}

}

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;

using System.IO;

using System.Text;

class Test

{

public static void Main()

{

string path = @"C:\Users\PULSE Electronics\Music\Struktura te te dhenave\Detyra\_shtepie\Kapitulli12\albion.txt";

if (!File.Exists(path))

{

string createText = "Hello and Welcome" + Environment.NewLine;

File.WriteAllText(path, createText, Encoding.UTF8);

}

Console.WriteLine();

string appendText = "\nThis is extra text" + Environment.NewLine;

File.AppendAllText(path, appendText, Encoding.UTF8);

string readText = File.ReadAllText(path);

Console.WriteLine(readText);

}

}

10.   Write a method that takes as a parameter the name of a binary file, **reads the content** of the file and returns it as an array of bytes. Write a method that **writes the file content** to another file. Compare both files.

using System;

using System.IO;

class BinaryRW

{

static void Main()

{

char[] invalidPathChars = Path.InvalidPathChars;

MemoryStream memStream = new MemoryStream();

BinaryWriter binWriter = new BinaryWriter(memStream);

binWriter.Write("Invalid file path characters are: ");

binWriter.Write(

Path.InvalidPathChars, 0, Path.InvalidPathChars.Length);

BinaryReader binReader = new BinaryReader(memStream);

memStream.Position = 0;

Console.Write(binReader.ReadString());

int arraySize = (int)(memStream.Length - memStream.Position);

char[] memoryData = new char[arraySize];

binReader.Read(memoryData, 0, arraySize);

Console.WriteLine(memoryData);

}

}

11.   Search for information in Internet and define your own class for exception **FileParseException**. The exception has to contain the name of the processed file and the number of the row where the problem is occurred. Add appropriate constructors in the exception. Write a program that reads integers from a text file. If the during reading a row does not contain an integer throw **FileParseException** and pass it to the calling method.

using System;

public class Person

{

private string \_name;

public string Name

{

get { return \_name; }

set { \_name = value; }

}

public override int GetHashCode()

{

return this.Name.GetHashCode();

}

public override bool Equals(object obj)

{

Person p = obj as Person;

if (p == null)

return false;

else

return this.Name.Equals(p.Name);

}

}

public class Example

{

public static void Main()

{

Person p1 = new Person();

p1.Name = "John";

Person p2 = null;

Console.WriteLine("p1 = p2: {0}", p1.Equals(p2));

}

}

13.   Write a program that **downloads a file from Internet** by given URL, e.g. [https://softuni.bg/forum](https://softuni.bg/).

using System;

using System.Net;

namespace ConsoleApp4

{

class Program

{

static void Main(string[] args)

{

WebClient webClient = new WebClient();

webClient.DownloadFile("https://softuni.bg/forum", @"C:\Users\PULSE Electronics\Music\Struktura te te dhenave\Detyra\_shtepie\Kapitulli12\albion.txt");

}

}

}



