using System;

namespace Homework12.\_10.\_23

{

public class MyException1 : ApplicationException

{

string msg;

public MyException1(string msg="MyClassException1")

{

this.msg = msg;

}

public override string Message

{

get { return msg; }

}

}

class MyException2 : ApplicationException

{

public MyException2(string msg ="MyClassException2"):base(msg)

{

}

}

[Serializable]

public class MyException3 : ApplicationException

{

public MyException3():base("MyException3333333"){}

public MyException3(string message = "MyException3") : base(message) { }

public MyException3(string message, Exception inner) : base(message="MyException3", inner) { }

protected MyException3(

System.Runtime.Serialization.SerializationInfo info,

System.Runtime.Serialization.StreamingContext context) : base(info, context) { }

}

class ArrayFloat

{

float[] AF;

public ArrayFloat(int size)

{

AF = new float[size];

Random rnd = new Random();

for(int i = 0; i < size; i++)

{

AF[i]= (float)(1+rnd.NextDouble()\*9);

}

}

public float this[int index]

{

get

{

if(index<0)

throw new MyException1();

if(index>=AF.Length)

throw new MyException2();

return AF[index];

}

set

{

if (index < 0)

throw new MyException1();

if (index >= AF.Length)

throw new MyException2();

}

}

public void Set(float value,int index)

{

if (index < 0 || index >=AF.Length)

throw new MyException3();

AF[index]= value;

}

}

internal class Program

{

static void Main(string[] args)

{

int size = 5;

ArrayFloat AF = new ArrayFloat(size);

for(int i = 0; i < size; i++)

{

Console.Write(AF[i]+" ");

}

Console.WriteLine();

try

{

AF.Set(10, 10);

/\* Console.WriteLine(AF[10]);

Console.WriteLine(AF[-1]);\*/

}

catch (Exception obj)

{

Console.WriteLine(obj.Message);

}

}

}

}