

# Algoritmos y Programación

## Práctica 7

- 1) ¿Qué salida por la consola produce el siguiente código?

```
int x=0;
try{
    Console.WriteLine(1.0/x);
    Console.WriteLine("Si es un numero real, hace la división por cero");
    Console.WriteLine(1/x);
    Console.WriteLine("Ok");
}
catch (Exception) { Console.WriteLine("No puede dividir por cero un numero entero");
}
```

¿Qué se puede inferir respecto de la excepción división por cero en relación al tipo de los operandos?

- 2) Suponiendo que está definida la clase NoEsElJefeException ¿Por qué no es atrapada la excepción en el siguiente código?

```
boolean jefe = true;
try{
    if(! jefe)
        new NoEsElJefeException();
    Console.WriteLine("Bienvenido jefe");
}
catch(NoEsElJefeException){
    Console.WriteLine("No está autorizado");
}
```

```
public class NoEsElJefeException: Exception {}
```

- 3) Compruebe el funcionamiento del siguiente programa:

```
using System;
class Ejercicio2 {
    static void Main(string[] args) {
        try{
            metodo1();
        }
        catch (Exception){
            Console.WriteLine("método 1 propagó una excepción ");
        }
        Console.ReadLine();
    }
    static void metodo1(){
        byte b=255;
        try{
```

```

        b++;    }
    finally{
        Console.WriteLine("bloque finally");
    }
}
}

```

¿Qué imprime? ¿Por qué?

4) Dado el siguiente programa:

```

using System;
class UnErrorException : Exception { }
class OtroErrorException : Exception { }
class Ejercicio4 {
    static void Main(string[] args) {
        int a = 1;
        Console.WriteLine("Entrando");
        try{
            Console.WriteLine("Hola");
            A(a);
            Console.WriteLine("Chau");
        }
        catch (UnErrorException) {
            Console.WriteLine("oh oh!");
        }
        catch (Exception) {
            Console.WriteLine("Pucha");
        }
        finally {
            Console.WriteLine("Gracias");
        }
        Console.WriteLine("Saliendo");
        Console.ReadLine();
    }
    static void A(int n){
        try{
            if(n==1)
                throw new UnErrorException();
            B(n);
        }
        catch (OtroErrorException) {
            Console.WriteLine("Vaya !");
        }
        finally{
            Console.WriteLine("OK");
        }
    }
}

```

```

static void B(int n){
    if (n == 2)
        throw new OtroErrorException();
    throw new OverflowException ();
}
}

```

Determine visualmente cual es la salida que producen el programa cuando:

- i. La variable a vale 1
- ii. La variable a vale 2
- iii. La variable a vale 3

Corrobore los resultados con la ejecución del código.

- 5) Implemente una función que reciba dos parámetros: un arreglo de enteros  $a$  y un valor entero  $v$ . Esta función debe devolver la posición dentro del arreglo donde se encuentra el valor  $v$ . Si el valor buscado no se encuentra en el arreglo entonces debe disparar una excepción definida por usted.
  
- 6) Implemente un juego de adivinanzas para dos jugadores. El primer jugador elige 5 números entre 1 y 10. Y luego el segundo jugador debe adivinar los 5 valores elegidos por el jugador 1.  
 Implemente un programa de aplicación (Main) que le permita al jugador 1 ingresar por teclado 5 números enteros del 1 al 10. Y luego le permita al jugador 2 ingresar uno a uno números hasta que adivine los 5 valores ingresados por el jugador 1. Al finalizar, el juego debe informar cuantos intentos necesitó el jugador 2 para adivinar los 5 números del jugador 1.  
 Para la implementación de este programa utilice la función programada en el ejercicio anterior.
  
- 7) Modifique el programa anterior para que se valide utilizando excepciones que por cada valor ingresado por ambos jugadores sea un número entero válido y esté en el rango de 1 a 10, en caso contrario que solicite el reingreso del dato.
  
- 8) Definir e implementar las clases Butaca y Cine. De cada butaca se conoce su nro y estado (libre o vendida). El cine tiene nombre, un cupo y el conjunto de butacas.  
 En un programa de aplicación (Main) crear el cine "San Martin" y simular la venta de entradas a varios espectadores (usar un while). De cada espectador se conoce el nro de entrada asignada (suponer que solo compra una entrada). En caso de que no hay cupo suficiente, disparar la excepción sinCupo que imprima el mensaje "no hay cupo disponible" e implemente el manejador correspondiente. Si se vende la entrada se debe modificar el estado de la butaca vendida y el cupo disponible del cine.