

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 datos de los operandos?

- 2) 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?

- 3) 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.

- 4) 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.

- 5) 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.

- 6) Definir e implementar las clases Remedio y Farmacia. De cada remedio se conoce su nombre comercial, la droga, el stock y el precio.

En un programa de aplicación (Main) crear la farmacia "Farmacity" y cargarle varios remedios. Luego simular la venta de remedios a varios clientes (usar un while). Cada cliente compra uno o más remedios. De cada remedio a comprar ingresa su nombre comercial y la cantidad de unidades deseadas. En caso de que no haya stock suficiente, disparar la excepción sinStock que imprima el mensaje "no hay stock suficiente" e implemente el manejador correspondiente. Si se concreta la compra hay que imprimir el monto total a abonar por el cliente y modificar el stock del remedio. Al finalizar el proceso informar el monto total recaudado por la farmacia.