

Laboratorio de Software

Práctica nº 6

Temas

- Conceptos y uso de excepciones
- Try/catch/finally
- Definición de Excepciones

1.- Determine si el siguiente código es correcto. Si produce un error, observe de qué tipo es y soluciónelo.

```
class Excepcion1 extends Exception{}
class Excepcion2 extends Excepcion1{}
public class Test1 {
    public static void main(String[] args) {
        try {
            throw new Excepcion2();
        } catch(Excepcion1 e1) {
            System.out.println("Se capturó la Excepción1");
        } catch( Excepcion2 e2) {
            System.out.println("Se capturó la Excepción2");
        }
    }
}
```

2.- Ejecute el siguiente código. ¿Cuál es el resultado?.
Elimine los comentarios y vuelva a ejecutarlo. ¿Cuál es el resultado?.

```
public class Test2 {
    public int unMetodo(){
        // try {
            System.out.println("Va a retornar 1");
            return 1;
        // } finally {
            System.out.println("Va a retornar 2");
            return 2;
        // }
    }
    public static void main(String[] args) {
        Test2 res = new Test2();
        System.out.println(res.unMetodo());
    }
}
```

3.- Ejecute el siguiente código. ¿Cuál es la salida del programa?

```

public class Test3 {
    public static void main(String[] args) {
        System.out.println("Test3");
        try {
            System.out.println("Primer try");
            try {
                throw new Exception();
            } finally {
                System.out.println("Finally del 2° try");
            }
        } catch (Exception e) {
            System.out.println("Se capturó la Excepción ex del 1° Primer try");
        } finally {
            System.out.println("Finally del 1° try");
        }
    }
}

```

4.- Analice el siguiente código y determine si es correcto. Si hay errores, escriba el motivo de cada uno y proponga una solución.

```

class FutbolException extends Exception{}
class Falta extends FutbolException{}
class EquipoIncompleto extends
    FutbolException{}
class ClimaException extends Exception{}
class Lluvia extends ClimaException{}
class Mano extends Falta{}
class Partido {
    Partido() throws FutbolException{}
    void evento() throws FutbolException{}
    void jugada() throws EquipoIncompleto,
        Falta{}
    void penal(){}
}
interface Tormenta {
    void evento() throws Lluvia;
    void diluvio() throws Lluvia;
}

```

```

public class Encuentro extends Partido
    implements Tormenta {

    Encuentro() throws Lluvia,
        FutbolException{..}
    Encuentro (String fecha) throws Falta,
        FutbolException{..}

    void penal() throws Mano{..}
    public void evento() throws Lluvia{..}
    public void diluvio() throws Lluvia{..}
    public void evento(){..}
    void jugada() throws Mano{..}

    public static void main (String[] args) {
        try {
            Encuentro enc = new Encuentro();
            enc.jugada();
        } catch(Mano e) {
        } catch(Lluvia e) {
        } catch(FutbolException e) {
        }
        try {
            Partido par = new Encuentro();
            par.jugada();
        } catch(EquipoIncompleto e) {
        } catch(Falta e) {
        } catch(Lluvia e) {
        } catch(FutbolException e) {}
        }
    }
}

```

5.- Analice el siguiente código:

```
public class Suma {  
    public static void main(String[] args){  
        int suma=0;  
        for(int i=0;i<=args.length;i++){  
            suma+= Integer.parseInt(args[i]);  
            System.out.print("La suma es:"+suma);  
        }  
    }  
}
```

- a)** Ejecútelo ingresando al menos 2 valores.
- b)** Ahora ejecútelo ingresando: **2 3 four**. ¿Qué pasó?. Solucione el problema de manera que los datos no numéricos sean impresos en la consola con un mensaje y descartados antes de ser sumados.
- c)** ¿Por qué no fue necesario capturar la excepción en el inciso **a)** ?