

Exercicis de Repàs

Exercici 1

Suposem que tenim una classe Empresa que te un array d'objectes Empleat. A aquesta classe podrem tenir mètodes per a contractar EmpleatS (afegir un nou objecte al array), despedir-lo (llevar-lo del array) u obtenir el nom a partir del número de empleat. La classe podria ser així com la següent:

```
public class Empresa {
    String nombre;
    Empleado [] listaEmpleados;
    int totalEmpleados = 0;
    ...
    Empresa(String n, int maxEmp) {
        nombre = n;
        listaEmpleados = new Empleado [maxEmp];
    }
    ...
    void nuevoEmpleado(String nombre, int sueldo) {
        if (totalEmpleados < listaEmpleados.length ) {
            listaEmpleados[totalEmpleados++] = new Empleado(nombre,sueldo);
        }
    }
}
```

Observa en el mètode nuevoEmpleado que comprova que hi ha lloc al array per a emmagatzemar la referència al nou empleat. Si està es crearà el objecte, però si no hi ha lloc el mètode no fa res més. No dona ninguna indicació de si la operació ha tingut èxit o no. Genera una excepció pròpia que controle açò.

Exercici 2

Indica que línies de codi executa el següent programa i quina seria la eixida per pantalla. Intenta solucionar-lo sense executar-lo i després comprova el resultat:

```
class ThrowDemo {
    static void demoproc() {
        try {
            throw new NullPointerException("demo");
        } catch(NullPointerException e) {
            System.out.println("Captura dentro de demoproc.");
            throw e;
        }
    }
    public static void main(String args[]) {
        try {
            demoproc();
        } catch(NullPointerException e) {
            System.out.println("Nueva captura: " + e);
        }
    }
}
```

Exercici 3

Indica que línies de codi executa el següent programa i quina seria la eixida per pantalla. Intenta solucionar-lo sense executar-lo i després comprova el resultat:

```
class FinallyDemo {
    static void procA() {
        try {
            System.out.println("Dentro de procA");
            throw new RuntimeException("demo");
        } finally {
            System.out.println("Sentencia finally de procA");
        }
    }

    static void procB() {
        try {
            System.out.println("Dentro de procB");
            return;
        } finally {
```

```

        System.out.println("Sentencia finally de procB");
    }
}
static void procC() {
    try {
        System.out.println("Dentro de procC");
    } finally {
        System.out.println("Sentencia finally de procC");
    }
}
public static void main(String args[]) {
    try {procA();
    } catch (Exception e) {
        System.out.println("Excepcion capturada");
    }
    procB(); procC();
}
}

```

Exercici 4

Indica que línies de codi executa el següent programa i quina seria la eixida per pantalla. Intenta solucionar-lo sense executar-lo i després comprova el resultat:

```

class MyException extends Exception {
    private int detail;
    MyException(int a) {
        detail = a;
    }
    public String toString() {
        return "MyException[" + detail + "]";
    }
}
class ExceptionDemo {
    static void compute(int a) throws MyException {
        System.out.println("Ejecuta compute(" + a + ")");
        if(a > 10)
            throw new MyException(a);
        System.out.println("Finalizacion normal");
    }
    public static void main(String args[]) {
        try {
            compute(1);
            compute(20);
        } catch (MyException e) {
            System.out.println("Captura " + e);
        }
    }
}

```

Exercici 5. Es correcte el següent codi?

```

try {

} finally {

}

```

Exercici 6 Què tipus d'excepcions poden capturar amb el següent codi?

```

catch (Exception e) {

}

```

Exercici 7. Es correcte el següent control d'excepcions? Compilarà aquest codi?

```

try {

} catch (Exception e) {

} catch (ArithmeticException a) {

}

```

Exercici 8

Explica que fan els següents mètodes al control d'errades:

Método
<code>getMessage()</code>
<code>printStackTrace()</code>
<code>toString()</code>

Exercici 9

Realitza un programa, amb el següent menú:

1. Afegir URL
2. Carregar URL
3. Eixir

Opció 1: que pregunte per un nick, una data i per una URL. Fes que la informació es guarde a un fitxer. Deuràs demanar tantes vegades la informació, com l'usuari vulga i guardar totes les indicades.

Opció 2: es preguntarà per el nick i una data, es buscarà al fitxer i una vegada trobat, es carregarà la URL al navegador.

A més, deurà controlar una possible errada de entrada/eixida(en el cas en que el fitxer no puga ser creat), en el que es mostrarà la pila de errades per les que ha passat i les errades possibles de entrada incorrecta de dades per pantalla.