

RESUM BÀSIC D'EXCEPCIONS EN JAVA.

DEFINICIONES PREVIAS:

- **Excepcions:** són el mecanisme de tractament d'errors (normalment recuperables) en java, es troben totes en la jerarquia que té com base la classe Exception, que al mateix temps és derivada de Throwable.
- **Excepcions Checked:** Són aquelles que deriven directament de la classe Exception. Es denominen Excepcions comprovades per que el compilador exigeix allà on es puga produir una d'aquestes Excepcions que s'incloga el còdi para tractar-les o propagar-les. Exemple: FileNotFoundException;
- **Excepcions Unchecked:** Són aquelles que deriven de la subclasse RuntimeException. Es denominen Excepcions no comprovades perquè el compilador no obliga al seu tractament ni propagació. No obstant, si es produeixen provoquen la finalització del procés (y habitualment del programa) i, per tant, és recomanable evitar-les o en el seu cas tractar-les o propagar-les. Exemple: ArrayIndexOutOfBoundsException;
- **Captura i Tractament:** és una forma d'actuar amb les Excepcions, es pot capturar darrere d'un bloque "try" en un bloque "catch" de tal manera que l'execució no finalitza abruptament sinó que continua.
- **Propagació:** és l'altra forma d'actuar amb les Excepcions. Si no es desitja efectuar el tractament de l'excepció en el mètode on es produeix es possible "propagar-la" cap al el mètode que el va invocar prèviament, derivant-se el tractament (o la propagació en el seu cas) al eixe mètode en qüestió.
- **throw:** es la paraula reservada para indicar en que se llança una excepció ja creada o que se crea en eixe mateix moment.
- **throws:** es la paraula reservada que indica en la capçalera d'un mètode que una excepció o varies (els noms de les quals es situen continuació de la paraula) es pot produir dins del cos del mètode però no seran tractades al seu interior. En el cas de les Excepcions unchecked aquesta forma explícita de indicar-ho no és necessària, però no pot deixar-se a mode de recordatori si així es requereix.
- **try:** es la paraula reservada para indicar l'inici del bloc de codi on es pot produir una excepció.
- **catch:** es la paraula reservada que indica el inici del bloc de codi on es captura i es tracta una excepció (El nom de la qual s'escriu explícitament). S'utiliza sempre després d'un bloque try. Pot haver més d'un i ha d'haver un necessàriament si no hi ha cap bloc finally. Des de la versió 7 es possible escriure més d'una Excepció dintre del mateix catch.
- **finally:** es la paraula reservada que indica el bloc que s'executa sempre després d'un try tant si s'ha produït excepció com si no. (Noteu que s'executa inclús si hi ha un return en el try, per exemple).

EXEMPLE:

```
public class MyException extends Exception{
    public MyException(String msg){ super(msg); }
}

import java.util.*;
public class Principal {
    public static String getPos (int x, String[] a)
        throws MyException{
        try {
            return (a[x]);
        }
        catch (ArrayIndexOutOfBoundsException e) {
            throw new MyException("Índice no"+
                " válido en el array");
        }
        finally {
            System.out.println (" Mètode finalitzat ");
        }
    }
    public static void main (String[] args){
        Scanner t=new Scanner(System.in);
        String[] aux={"Verde","Amarillo","Rojo"};
        System.out.print("Introduce un indice: ");
        try{
            int indice=t.nextInt();
            System.out.println(getPos(indice,aux));
        }
        catch (MyException e) {
            System.out.println("Error, fuera de rango");
        }
        catch (InputMismatchException e) {
            System.out.println("Error, no es entero");
        }
    }
}
```

Ejemplos d'execució:

```
Introduce un indice: 32
Mètode finalitzat
Error, fuera de rango

Introduce un indice: -5
Mètode finalitzat
Error, fuera de rango

Introduce un indice: 8.4
Error, no es entero

Introduce un indice: 1
Mètode finalitzat
Amarillo
```