

## UNITAT 10

### EXCEPCIONS

### EXERCICIS

Autors: Carlos Cacho, Raquel Torres i Luis José Sánchez González

Revisat per: Lionel Tarazon

Fco. Javier Valero

José Manuel Martí

Àngel Olmos Giner

2022/2023

#### Llicència



**[CC BY-NC-SA 3.0 ES](https://creativecommons.org/licenses/by-nc-sa/3.0/es/) Reconeixement – No Comercial – Compartir Igual (by-nc-sa)**

No es permet un ús comercial de l'obra original ni de les possibles obres derivades, la distribució de les quals s'ha de fer amb una llicència igual a la que regula l'obra original. Aquesta és una obra derivada de l'obra original de Carlos Cacho, Raquel Torres i Luis José Sánchez González

## UF10. EXCEPCIONS

### IMPORTANT

- En aquests exercicis és fonamental fer **diverses proves** per a comprovar i **comprendre** què succeeix en cada cas (segons el tipus d'excepció, quan no hi ha excepcions, etc.)
- Llevat que s'indique el contrari, al llançar una excepció hauràs d'incloure un **missatge breu** sobre l'error (*new Exception("...")*), i quan captures excepcions hauràs de mostrar informació sobre l'error

### EXERCICI 1

Implementa un programa que demane a l'usuari un valor sencer **A** i després mostre per pantalla el missatge "Valor introduït: ...".

S'haurà de tractar l'excepció ***InputMismatchException*** que llança *Scanner* quan no s'introdueix un enter vàlid. En tal cas es mostrarà el missatge "Valor introduït incorrecte".

### EXERCICI 2

Implementa un programa que demane dos valors sencers **A** i **B**, calcule **A/B** i mostre el resultat per pantalla. S'hauran de tractar de manera independent les dues possibles excepcions, mostrant en cada cas un missatge d'error diferent en cada cas.

### EXERCICI 3

Implementa un programa que cree un vector de tipus *double* de grandària 5 i després, utilitzant un bucle, demane cinc valors per teclat i els introduïska en el vector.

Hauràs de manejar la/les possibles excepcions i **continuar demanant valors fins a emplenar completament el vector**.

Finalment es mostrarà el contingut del vector mitjançant un mètode.

### EXERCICI 4

Implementa un programa que cree un vector d'enters de grandària N (nombre aleatori entre 1 i 100) amb valors aleatoris entre 1 i 10.

Després se li preguntarà a l'usuari quina posició del vector vol mostrar per pantalla, repetint-se una vegada i una altra fins que s'introduïska un valor negatiu.

Maneja totes les possibles excepcions.

---

## EXERCICI 5 - Extra

Implementa un programa amb els mètodes:

- **void imprimeixPositiu(int p):** Imprimeix el valor p. Llança una *Exception* si  $p < 0$
- **void imprimeixNegatiu(int n):** Imprimeix el valor n. Llança una *Exception* si  $n \geq 0$

Fes un bucle que és repeteix 5 vegades i que demana un valor positiu, un valor negatiu i els passa a les funcions.

Maneja totes les possibles excepcions.

## EXERCICI 6

Implementa una **classe Gat** amb els atributs nom i edat, un constructor amb paràmetres, els *getters* i *setters*, a més d'un mètode *imprimir()* per a mostrar les dades d'un gat.

El nom d'un gat ha de tindre almenys 3 lletres (i només lletres) i l'edat no pot ser negativa.

Hauràs de comprovar que els valors siguin vàlids i llançar una *Exception* en totes les parts del codi que modifiquen aquests valors.

Després, fes una classe principal per a fer proves:

- instància diversos objectes Gat, amb valors correctes i incorrectes
- modifica el nom i l'edat d'alguns gats (alguns vàlids i altres incorrectes)

En tots els casos, caldrà que el codi pugui manejar les excepcions.

## EXERCICI 7

Crea una còpia del programa anterior i modifica el *main* per a fer el següent:

- Crea una llista per a guardar Gats
- Utilitzant un bucle, demana a l'usuari que introduïska les dades de 5 gats i guarda'ls en la llista
- Finalment, imprimeix la informació dels gats

Maneja les excepcions (totes de la classe *Exception* de moment) de manera que en la llista només s'emmagatzemen objectes vàlids i el bucle es repetisca fins a crear i emmagatzemar correctament els 5 gats

## EXERCICI 8

Modifica el programa anterior per a:

- Una vegada impresa la informació de tots els gats, preguntar a l'usuari quin gat vol esborrar de la llista (demanar l'índex de la llista a esborrar) i torna a imprimir la llista resultant
  - Gestiona les possibles excepcions que puguin ocórrer en el programa de forma independent, utilitzant la classe l'excepció més adequada (les de nom i edat mantén-les com *Exception*)
-

### EXERCICI 9

Modifica el programa anterior per a:

- Crear excepcions pròpies per a la gestió dels errors en el nom i l'edat
- Les excepcions agafaran com a argument d'entrada el nom o l'edat incorrecta
- Cal sobreescrivre el *toString()* de les noves classes per a especificar l'error i utilitzar-lo quan es gestionen les excepcions
- Adapta el programa principal a les noves excepcions i elimina ja definitivament els manejadors d'*Exception*

### EXERCICI 10

Recupera l'exercici MASCOTES del tema anterior i realitza les modificacions necessàries per a llançar una excepció si s'intenta:

- Eliminar una mascota inexistent en l'inventari
- Mostrar una mascota inexistent en l'inventari
- Crear una mascota sense nom
- Crear una mascota amb edat negativa

Caldrà que modifiques també el programa principal per tal de manejar determinades excepcions. Totes les excepcions han de ser pròpies i les seues classes creades en un mateix arxiu «ExcepcionsPropies.java».

### EXERCICI 11

Recupera l'exercici BANC del tema anterior i realitza les modificacions necessàries per a manejar TOTES les excepcions que es puguin llançar (divisió per 0, error de tipus, índex fora de límits ...).

A més a més, caldrà que llances excepcions pròpies en les següents situacions:

- codi IBAN no té el format correcte
  - no hi ha suficient saldo per a fer un moviment
  - el moviment supera el valor màxim
-

## EXERCICI 12

Has de modificar el següent programa per a tractar totes les possibles excepcions que es puguin llançar. A més a més, quan ocorregui un error, es mostrarà el missatge de l'excepció i es tornarà a demanar la dada a l'usuari per tal de poder finalitzar el programa de forma correcta.

```
public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    double n;
    int posicio;
    String cadena;
    double[] valors = {9.83, 4.5, -3.06, 0.06, 2.52, -11.3, 7.60, 3.00, -30.4, 105.2};

    System.out.println("Contingut de l'array abans de modificar:");
    for (int i = 0; i < valors.length; i++) {
        System.out.printf("%.2f ", valors[i]);
    }

    System.out.print("\n\nIntrodueix la posició de l'array a modificar: ");
    cadena = sc.nextLine();
    posicio = Integer.parseInt(cadena);

    System.out.print("\nIntrodueix el nou valor de la posició " + posicio + ": ");
    n = sc.nextDouble();

    valors[posicio] = n;

    System.out.println("\nPosició a modificar " + posicio);
    System.out.println("Nou valor: " + n);
    System.out.println("Contingut de l'array modificat:");
    for (int i = 0; i < valors.length; i++) {
        System.out.printf("%.2f ", valors[i]);
    }
}
```

## EXERCICI 13

Crea un programa "generador aleatori d'excepcions".

El programa crearà un *ArrayList* on s'instanciaran 5 excepcions diferents i a continuació es llançarà una de les 5 de forma aleatòria (cada vegada que s'execute el programa, es podrà llançar una diferent).

Captura l'excepció llançada i mostra un missatge significatiu per pantalla.

---

## EXERCICI 14

Crea un programa per a "apariar gats".

Els gats estan definits pel seu nom i sexe (mascle o femella), que són uns valors que no poden canviar.

Tenen la capacitat d'apariar-se amb altres gats. L'acte d'apariar-se donarà com a resultat un nou gat sense nom i amb sexe aleatori entre mascle o femella.

En el cas que es vulga aparejar dos gats del mateix sexe, es llançarà l'excepció "ExceptionApariamentImpossible" que una vegada capturada mostrarà el missatge:

```
**** ERROR: Dos gats femella (NomDelGat1, NomDelGat2) no es poden aparejar
o
**** ERROR: Dos gats mascle (NomDelGat1, NomDelGat2) no es poden aparejar
```

Si per contra s'intenta aparejar a un gat/a amb si mateixa, **la mateixa excepció** mostrarà el missatge:

```
**** ERROR: No es pot aparejar a NomDelGat/a amb si mateix
```

En el programa principal hauràs de:

1. Crear 4 gats diferents (2 femelles i 2 mascles)
2. Aparejar-los 10 vegades entre ells de forma aleatòria. És a dir, que cada vegada la selecció dels 2 gats/es a aparejar ha de ser aleatòria entre els 4 gats/es existents
3. En cas que l'aparellament siga possible, es mostrarà el missatge:

```
L'aparellament de NomDelGat1 amb NomDelGat2 ha resultat en un gatet
o
L'aparellament de NomDelGat1 amb NomDelGat2 ha resultat en una gateta
```

---