

## UNITAT 10.

### EXCEPCIONS

### EXERCICIS

PROGRAMACIÓ  
CFGS DAW

Autors:

Carlos Cacho y Raquel Torres

Revisat per:

Lionel Tarazon - [lionel.tarazon@ceedcv.es](mailto:lionel.tarazon@ceedcv.es)

Fco. Javier Valero – [franciscojavier.valero@ceedcv.es](mailto:franciscojavier.valero@ceedcv.es)

José Manuel Martí - [josemanuel.marti@ceedcv.es](mailto:josemanuel.marti@ceedcv.es)

2021/2022

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 i Raquel Torres

## UF10. EXCEPCIONS

## EXERCICIS

## IMPORTANT

- En aquests exercicis és fonamental fer **diverses proves** per a comprovar i **comprendre** què succeeix en cada cas (segons la mena d'excepció, quan no hi ha excepcions, etc.).
  - Llevat que s'indique el contrari, en llançar una excepció hauràs d'incloure un **missatge breu** sobre l'error (`new Exception("...")`), i quan captures excepcions hauràs de mostrar **la pila de cridades** (`printStackTrace()`).
1. Implementa un programa que demane a l'usuari un valor enter **A** utilitzant un **nextInt()** (de Scanner) i després mostre per pantalla el missatge "Valor introduït: ...". S'haurà de tractar l'excepció **InputMismatchException** que llança `nextInt()` quan no s'introdueix un enter vàlid. En tal cas es mostrarà el missatge "Valor introduït incorrecte".
  2. Implementa un programa que demane dos valors int **A** i **B** utilitzant un **nextInt()** (de Scanner), calcule **A/B** i mostre el resultat per pantalla. S'hauran de tractar de manera independent les dues possibles excepcions, **InputMismatchException** i **ArithmeticException**, mostrant en cada cas un missatge d'error diferent.
  3. Implementa un programa que cree un vector 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**.
  4. Implementa un programa que cree un vector d'enters de grandària N (número 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.
  5. Implementa un programa amb tres funcions:
    - **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  $p \geq 0$
    - La funció main per a realitzar proves. Pots cridar a totes dues funcions diverses vegades amb diferents valors, fer un bucle per a demanar valors per teclat i passar-los a les funcions, etc. Maneja les possibles excepcions.

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 caràcters i l'edat no pot ser negativa. Per això, tant en el constructor com en els setters, hauràs de comprovar que els valors siguin vàlids i llançar un 'Exception' si no ho són. Després, fes una **classe principal amb main** per a fer proves: instància diversos objectes Gat i utilitza els seus setters, provant diferents valors (alguns vàlids i altres incorrectes). Maneja les excepcions.
7. Crea **una còpia del programa anterior** i modifica el main per a fer el següent:
  - Crea un ArrayList<Gat>. Després, utilitzant un bucle, demana a l'usuari que introduïska les dades de 5 gats: utilitza un Scanner per a demanar les dades, instància l'objecte i guarda'l en el ArrayList. Finalment, imprimeix la informació dels gats.
  - Maneja les possibles excepcions de manera que en el ArrayList només emmagatzemem objectes Gat vàlids i el bucle se repetisca fins a crear i emmagatzemar correctament 5 gats.
8. Anem a millorar el programari del **cas pràctic DawBank** de la unitat 8 afegint excepcions i alguna cosa més. Crea un nou projecte i copia el teu propi codi de DawBank (o el de la solució proposada). Fes els següents canvis:
  1. Crea una nova classe CompteException que herete de Exception. La utilitzarem per a llançar excepcions relacionades amb comptes bancaris.
  2. Crea una nova classe AvisarHisendaException que herete de Exception. La utilitzarem per a llançar una excepció quan calga avisar a hisenda.
  3. Modifica la classe CompteBancari:
    - A. Els moviments hauran d'emmagatzemar-se en un ArrayList en lloc d'en un vector. Ja no serà necessari limitar a 100 el núm. de moviments.
    - B. No es mostrarà cap mena de missatge d'error. En el seu lloc, es llançaran excepcions.
    - C. Quan s'intente realitzar alguna cosa incorrecta o no permesa es llançarà una excepció CompteException (haurà d'incloure un missatge breu sobre l'error produït).
    - D. Quan calga avisar a hisenda es llançarà l'excepció AvisarHisendaException, que contindrà informació sobre el titular, l'IBAN i l'operació realitzada. Recorda que encara que s'avise a hisenda l'operació ha de realitzar-se de totes maneres.
  4. Modifica la classe principal que conté el main per a manejar totes les possibles excepcions (no sols les de la classe CompteBancari), mostrant els missatges d'error oportuns i els printStackTrace().