

# Glossari de termes del Bloc 2

## (Programació orientada a objectes)

- **Abstracció** *f* Capacitat de concebre un concepte sense pensar en cap exemple específic.
- **Àlies** *f* Dos o més referències que es refereixen al mateix objecte
- **Array** *m* Agrupació fitada d'elements d'un mateix tipus.
- **Asserció** *f* Sentència de codi verdader-fals per comprovar una suposició en temps d'execució.
- **Atribut** *m* Descripció d'una característica de totes les instàncies d'una classe. En general, representa una qualitat dels elements del món real dins d'una classe.
- **Bloc condicional** *m* Conjunt d'instruccions que s'executen o no depenent d'una certa condició.
- **Bloc iteratiu** *m* Conjunt d'instruccions que s'executen reiterativament fins que es compleix una certa condició.
- **Booleà** *m* Tipus de dada que representa els valors de certesa o falsedat.
- **Capçalera d'un mètode** *f* Descripció d'un mètode: nom, paràmetres i tipus del retorn.
- **Casting** *m* Tècnica de conversió de dades que ens permet utilitzar una instància d'una classe o tipus bàsic de dades com si es tractés d'una instància d'un altre tipus o classe. Permet conversions d'ampliació i reducció.
- **Classe** *f* Estructura que permet definir entitats amb propietats comunes mitjançant l'abstracció i que representa un element del món real.
- **Classe abstracta (o classe diferida)** *f* Classe que queda definida, però que no accepta instàncies concretes.
- **Classe embolcall (Wrapper class)** *f* Classes que encapsulen l'accés o utilització d'un altra classe o tipus de dades.
- **Classe final** *f* Classe de la qual ja no es pot estendre més la jerarquia d'herències.
- **Classe parametritzada** *f* Classe la descripció de la qual depèn d'un o diversos paràmetres.
- **Classe sincronitzable** *f* Classe amb mètodes als que no es pot accedir al mateix temps des de diferents tasques evitant problemes de sobreescritura.
- **Constructor** *m* Mètode especial que inicialitza un objecte en el moment de la seva creació.
- **Delegació** *f* Acció de demanar a un altre objecte la resolució, total o parcial, d'un mètode.
- **Destructor** *m* Mètode que realitza les tasques prèvies a l'eliminació de l'objecte.
- **Encapsulament** *m* Ocultació de la implementació concreta d'una classe en què s'ofereixen només aquelles dades que es vol que les altres classes puguin utilitzar. És una característica dels mòduls que permet abstroure la gestió de la informació d'aquests de la representació interna de la informació que contenen.
- **Especialització** *f* Acció de crear una classe a partir d'una altra afegint-li significat.
- **Excepció** *f* Tipus de classe que serveix per a millorar el tractament d'errors.
- **Flux de control d'una invocació** *m* Ordre en el que s'executen les instruccions d'un programa, sent les pròpies instruccions les que determinen o controlen dit flux.
- **Genericitat** *f* Característica d'un element de programació que permet que s'utilitzi amb tipus de dades diferents.

## Programació 2.

Grau d'Enginyeria Informàtica. Facultat de Matemàtiques. UB

Laura Igual

- **Herència** *f* Mecanisme amb què es defineix una classe a partir d'unes altres, descrivint només les característiques que la diferencien de les altres.
- **Herència múltiple** *f* Forma no restringida d'herència de manera que es poden tenir tantes superclasses com facin falta.
- **Herència simple** *f* Herència en què només es pot tenir una única superclasse.
- **Instanciació** *f* Acció que crea una instància d'una classe.
- **Interfície** *f* Col·lecció de declaracions de mètodes abstractes que defineixen un comportament, però no l'implementa. Permet implementar herència múltiple en Java.
- **Invocació d'un mètode** *f* Crida d'un mètode.
- **Java** Llenguatge de programació d'alt nivell, orientat a objectes.
- **Llibreria de classes** *f* Col·lecció de classes, implementades per tercers, que podem utilitzar per desenvolupar programes i que permeten reutilització del codi.
- **Matriu** *m* Conjunt d'elements, generalment del mateix tipus, als quals es pot accedir per mitjà d'un índex. Poden tenir d'1 a  $n$  dimensions.
- **Mètode** *m* Bloc de codi que permet realitzar una tasca o algorisme concret. Es pot entendre com un símil en la programació orientada a objectes d'una acció que es pot realitzar sobre l'element del món real representat per una classe.
- **Mètode consultor** *m* (*getter*) Mètode que retorna el valor actual d'una variable.
- **Mètode modificador** *m* (*setter*) Mètode per establir el valor d'una variable.
- **Mètode de classe** *m* Mètode que és el mateix per a tota una classe d'objectes i no depèn de les dades concretes de cap instància. No cal que existeixi cap objecte per a poder-lo aplicar.
- **Mètode d'objecte** *m* Mètode que pot ser el mateix per a tota una classe d'objectes i depèn de les dades concretes d'una instància. Cal que existeixi un objecte al qual poder-lo aplicar.
- **Modificadors de visibilitat** *m* Paraula reservada Java que especifica característiques particulars d'un mètode o de les dades. Exemples, public, private, protected.
- **Objecte** *m* Instància d'una classe d'objectes.
- **Operador** *m* Element del llenguatge que permet definir una operació sobre les dades a les quals afecta.
- **Polimorfisme** *m* Capacitat que permet substituir el comportament d'un mètode dins una jerarquia de classes.
- **Programació orientada a objectes** *f* Enfocament per a desenvolupar aplicacions en què el món real es representa mitjançant mòduls de compilació separada, denominats *classes*, que es relacionen entre ells.
- **Programació procedimental** *f* Enfocament del desenvolupament d'aplicacions basat en la descomposició funcional d'aquestes.
- **Recursivitat** *f* Definició d'un procediment en termes de si mateix.
- **Referència** *f* Forma de referir-se a un objecte.
- **Reutilització** *f* Propietat per la qual el mateix mòdul (peça de programari) es pot utilitzar diverses vegades en diferents aplicacions.
- **Signatura d'un mètode** *f* Allò que identifica un mètode de forma única: la classe en la que es defineix, el nom, el número, ordre i tipus dels arguments.
- **Sobrecàrrega** *f* Capacitat perquè el nom d'un mètode o atribut faci referència a dos o més elements.
- **Tipus estàtic** *m* Tipus d'una variable que es determina en temps de compilació.
- **Tipus dinàmic** *m* Tipus d'una variable que només es pot conèixer en temps d'execució i pot canviar.

## Programació 2.

---

Grau d'Enginyeria Informàtica. Facultat de Matemàtiques. UB

Laura Igual

- **Vector**  $m$  Matriu d'una dimensió. La majoria de llenguatges de programació tenen un vector com a tipus de dada.