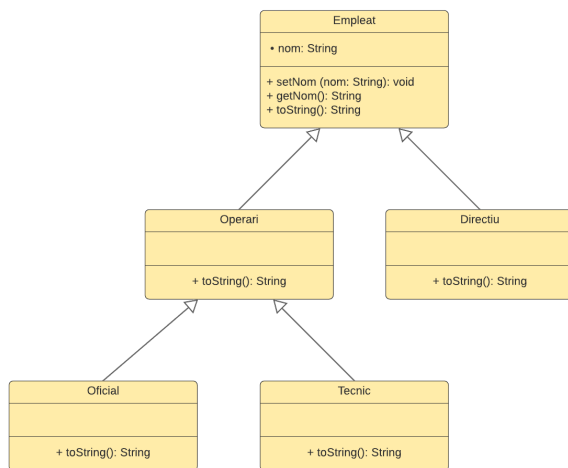


[🏠](#) > [Els Meus Cur...](#) > [Programació-1CFSC LFP 1726475...](#) > [UT6. HERÈNCIA, POLIMORFISME I INTERFÍ...](#) > [Exercicis cla...](#)

Exercicis classe

- Codifica la següent jerarquia de classes Java representada per este diagrama UML:



La resta de classes només han de sobreescriure el mètode toString() en cadascuna d'elles i declarar el constructor adequat de manera que quan l'execució de les següents instruccions:

```

Empleat E1 = new Empleat("Rafa");
Directiu D1 = new Directiu("Maria");
Operari OP1 = new Operari("Alfons");
Oficial OF1 = new Oficial("Laura");
Tècnic T1 = new Tècnic("Pau");
System.out.println(E1);
System.out.println(D1);
System.out.println(OP1);
System.out.println(OF1);
System.out.println(T1);
  
```

Donarà com a resultat:

```

Empleat Rafa
Empleat Maria -> Directiu
Empleat Alfons -> Operari
Empleat Laura -> Operari -> Oficial
Empleat Pau -> Operari -> Tècnic
  
```

- Es planteja desenvolupar un programa Java que permeta la gestió d'una empresa agroalimentària que treballa amb tres tipus de productes: productes frescos, productes refrigerats i productes congelats. Tots els productes porten esta informació comuna: data de

Hi ha tres tipus de productes congelats: congelats per aire, congelats per aigua i congelats per nitrogen. Els productes congelats per aire han de portar la informació de la composició de l'aire amb què va ser congelat (% de nitrogen, % d'oxigen, % de diòxid de carboni i % de vapor d'aigua). Els productes congelats per aigua han de portar la informació de la salinitat de l'aigua amb què es va realitzar la congelació en grams de sal per litre d'aigua. Els productes congelats per nitrogen han de portar la informació del mètode de congelació emprat i del temps d'exposició al nitrogen expressada en segons.

Crea el codi de les classes Java implementant una relació d'herència seguint estes indicacions:

a) En primer lloc realitza un esquema en paper i bolígraf on es represente com s'organitzaran les classes quan escriguem el codi. Estudia els atributs de les classes i trasllada a la classe base tot atribut que puga ser traslladat.

b) Cada classe ha de disposar de constructor i permetre establir (set) i recuperar (get) el valor dels seus atributs i tindre un mètode que permeta mostrar la informació de l'objecte quan siga procedent.

Crea una classe testHerencia amb el mètode main on es creen: dos productes frescos, tres productes refrigerats i cinc productes congelats (2 d'ells congelats per aigua, altres 2 per aire i 1 per nitrogen). Mostra la informació de cada producte per pantalla.

- Indica si el següent codi és correcte. Raona la resposta:

```
1
2  abstract class A {
3      abstract void m1();
4      void m2() {
5          System.out.println("Sóc
6          el mètode 2 d'A");
7      }
8  }
9  class B extends A {
10      void m2() {
11          System.out.println("Sóc el
12          mètode 2 de B");
13      }
14  }
15
```

- Crea una jerarquia simple "figura": una classe base anomenada Figura i unes classes derivades anomenades Cercle, Quadrat, i Triangle. En la classe base, cal fer una funció abstracta anomenada imprimirTipus(), i sobreescriure-la en les classes derivades. Declara un array d'objectes Figura que es cree en la classe principal i omple-ho amb diverses figures. Posteriorment crida a imprimirTipus(), iots els elements de l'array per a verificar que el comportament varia tot i tractar-se d'un array de tipus Figura.

Llenç.

- Lloguer de vaixells

En un port es lloguen amarraments per a vaixells de diferent tipus. Per a cada LLOGUER es guarda el nom i DNI del client, dies d'ocupació, la posició de l'amarrament i el vaixell que l'ocuparà. Un VAIXELL es caracteritza per la seua matrícula, la seua eslora en metres i any de fabricació.

El preu del lloguer es calcula multiplicant el nombre de dies d'ocupació per un factor de cada vaixell (obtingut multiplicant per 10 els metres d'eslora i afegint un valor que depén de cada tipus de vaixell).

No obstant això ara es pretén diferenciar la informació d'alguns tipus de vaixells:

- Nombre de mastelers per a velers (en este cas el valor a afegir serà el nombre de mastelers per 5)
- Potència en CV per a embarcacions esportives a motor (el valor a afegir serà la meitat de la potència en cavalls)
- Potència en CV i nombre de cabines per a iots de luxe (el valor a afegir serà la meitat de la potència en cavalls, més el nombre de cabines per 10)

Utilitzant l'herència de manera apropiada, dissenya el diagrama de classes i les seues relacions, amb detall d'atributs i mètodes necessaris. Programa els mètodes que permeten calcular el lloguer de qualsevol tipus de vaixell. Crea una classe principal amb un conjunt de "lloguers" per a comprovar que se'ls pot assignar qualsevol tipus de vaixell i calcular el seu preu de lloguer.

- Gestió d'un Zoològic.

Es vol implementar la jerarquia de classes que representen els animals d'un possible zoològic:

La classe Animal és una classe abstracta amb quatre atributs membres "protected":

- a) Una cadena indicant l'espècie (lleó, àguila, abella...)
- b) Una cadena indicant el nom de l'animal concret.
- c) Una dada numèrica real indicant el pes en kg.
- d) Una dada numèrica entera indicant el número de gàbia que s'assigna a l'animal.

A més, la classe Animal declara un mètode abstracte `getTipusAnimal()` que caldrà definir en les classes derivades.

La classe Mamífer no afegir nous atributs membre, encara que haurà d'implementar el mètode `getTipusAnimal()`.

La classe Ocell té dos nous atributs "protected":

- a) Una cadena `colorPlomatge` indicant el color predominant
- b) Una dada numèrica real indicant la altura màxima de vol (`alturaMaximaVol`).

La classe Reptil un nou atribut membre `protected` de tipus booleà anomenat `nada` que indica si el rèptil nada o no. Per a realitzar este exercici es demana el següent:

- a) Crear les quatre classes indicades, amb els corresponents constructors i sobrecàrrega de constructors
- b) Definir els mètodes anomenats `getTipusAnimal()`. Este mètode no té valor de retorn (és void) ni arguments. Ha de ser capaç de

Soc un mamífer

Em dic X

Soc de l'espècie: X

Pes X Kg

Estic en la gàbia: X

c) Crea una classe executable anomenada Zoologic. Prova la jerarquia de classes implementades creant un conjunt d'animals i mostrant la seua informació.

-

PER A PRACTICAR MÉS:

<https://sites.google.com/site/pro012iessanandres/java/ejercicios-sobre-herencia>

<https://www.discoduroderoer.es/ejercicios-propuestos-y-resueltos-programacion-orientado-a-objetos-java/>

<https://javaparakavatos.wordpress.com/category/ejercicios-de-herencia/>

<https://www.codesdope.com/practice/java-abstract-class/>

Ens demanen crear una sèrie de classes per a un programari d'un centre de formació professional. Implementa la jerarquia de classes necessària per a complir els següents criteris:

Classe 'Persona'

- Una persona té un nom, un dni, una direcció i un telèfon.
- L'atribut "dni" no podrà ser canviable, és a dir no es podrà modificar una vegada establert.
- Haurà de tindre un únic constructor amb tots els seus atributs, així com tots els "getters" i "setters" que siguen convenients.
- Haurà de tindre un mètode públic que s'encarregue de mostrar tota la informació de la persona.

Classe 'Alumne'

- Un alumne ÉS UNA persona.
- A més dels atributs que té per ser una persona, també disposa d'un número d'expedient, el nom del cicle formatiu que estudia, el curs en què es troba i un conjunt de notes que ha obtingut
- Haurà de tindre un únic constructor amb tots els seus atributs, així com els "getters" i "setters" que siguen convenients.
- Haurà de tindre un mètode públic que s'encarregue de mostrar tota la informació de l'alumne. Utilitza per a això la sobreescritura de mètode que ens proveeix l'herència en Java.

- A més dels atributs que té per ser una persona, també disposa d'un codi numèric, el nom del departament al qual pertany, l'horari que realitza (matins o vesprades) i un conjunt de mòduls que imparteix (el nom dels mòduls).
- Haurà de tindre un únic constructor amb tots els atributs, així com els "getters" i "setters" necessaris.
- Haurà de tindre un mètode públic que s'encarregue de mostrar tota la informació del professor. Utilitza per a això la sobreescritura de mètode que ens proveeix l'herència en Java.

Implementa també la classe '**TestPersona**' amb un mètode principal per a fer algunes proves:

- Declara i instància una Persona amb el nom de variable "persona1". Prova de modificar el seu telèfon i finalment imprimeix la seua informació amb el mètode creat per a eixe fi.
- Instància un nou alumne i assigna'l a la variable anterior "persona1". Prova de modificar el seu nom de cicle i finalment imprimeix la seua informació amb el mètode creat per a eixe fi.
- Declara una variable anomenada "persones" (Pots utilitzar un ArrayList o bé un array de 5 elements). Esta estructura haurà de contindre un alumne en la primera posició, 3 professors a continuació i finalment un altre alumne.
- Demostra el polimorfisme. Per a això hauràs de recórrer l'estructura "persones" i cridar per a cadascun dels seus elements al mètode que imprimeix la informació d'eixe element. Com a resultat s'haurà de mostrar la informació d'alumne o de professor segons el tipus dinàmic.

Última modificació: Tuesday, 1 March 2022, 4:35 PM

< Ejemplo Fraccion

Salta a...

Exercicis classe (Interfícies) >



© 2024 Generalitat

Conselleria d'Educació, Cultura, Universitats i Ocupació

[Avís legal](#) | [Política de Privacitat](#) | [Política de Cookies](#) | [Contacte](#) | [Accessibilitat](#)



UNIÓ EUROPEA
Fons Europeu de Desenvolupament Regional