Pt11.1 Activitats Herència

1.Objecte Geomètric.

Crea la classe ObjecteGeometric que té:

atributs: coord_x, coord_y, color

mètodes: constructor, toString()

Crea la classe Cercle que hereda de ObjecteGeometric i, a més a més, té:

atributs: radi

mètodes: constructor, toString(), perimetre() i area()

Crea la classe Quadrat que hereda de ObjecteGeometric i, a més a més, té:

atributs: costat

mètodes: constructor, toString(), perimetre() i area()

També haureu de crear el pograma principal que comprova que tot funcioni correctament.

Recordatori:

Àrea cercle = pi * radi * radi
Perímetre cercle = 2 * pi * radi
Àrea quadrat = costat * costat
Perímetre quadrat = 4 * costat

2. Vehicles

Es pretén desenvolupar una aplicació que permet calcular els preus de lloguer d'una empresa de lloguer de vehicles.

Cada vehicle s'identifica per la seva matrícula.

La empresa lloga diferents tipus de vehicles, tant per transport de persones com de càrrega. En la actualitat, els vehicles llogats per la empresa són: cotxes, microbusos, furgonetes de càrrega i camions.

El preu del lloguer de qualsevol vehicle té una component base que depèn dels dies de lloguer a raó de 10€ per dia.

En el cas de lloguer d'un cotxe, al preu base se li suma la quantitat de 1.5€ per plaça i dia.

El preu de lloguer del microbusos és igual que el dels cotxes, però se li afegeix una quantitat de 2€ per plaça independentment dels dies de lloguer.
El preu dels vehicles de càrrega és el preu base més 20€ * PMA (on PMA És el pes màxim autoritzats en tones).
A més a més, en cas dels camions, al preu se li suma un fixe de 40€ independentment dels dies de lloguer.

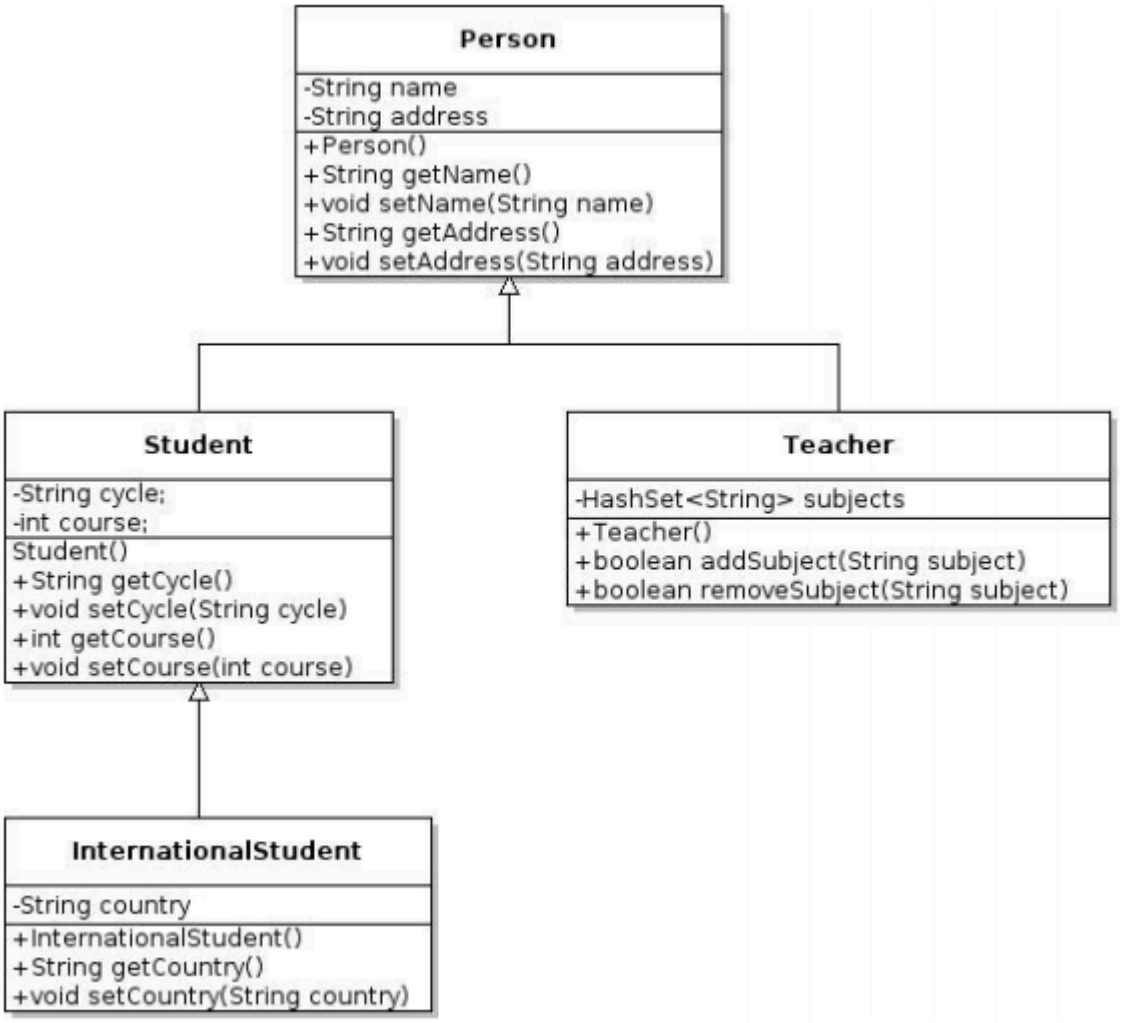
Les operacions que el empleat de la empresa de lloguer ha de poder realitzar són les següents:

1. Afegir un vehicle.
2. Obtenir el preu del vehicle.

3. School.

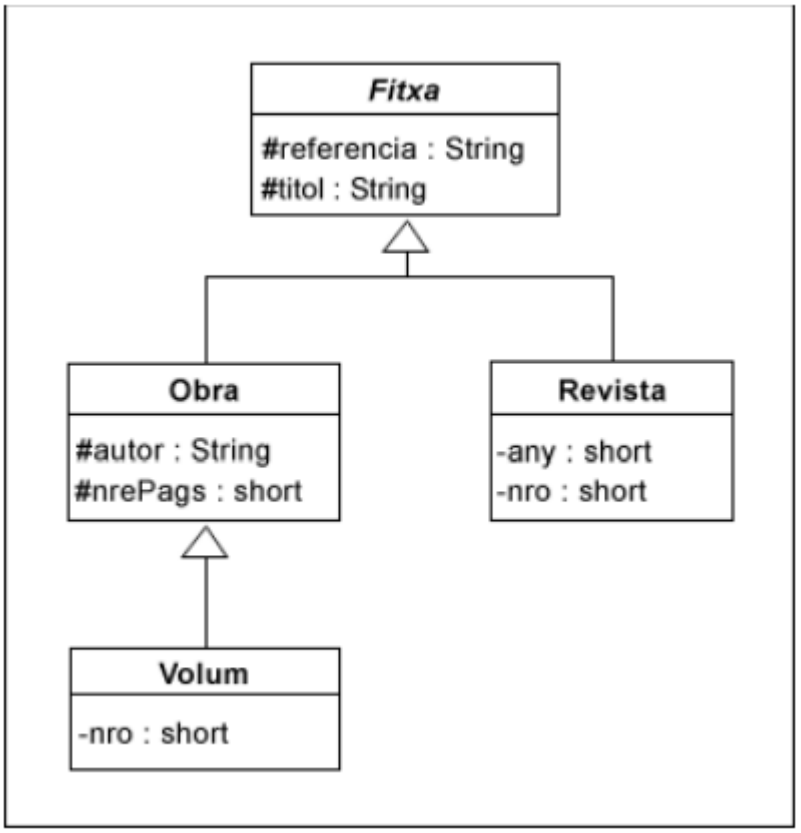
Defineix les classes corresponents al següent diagrama UML.

Afegeix constructors a cada classe de manera que el constructor cridi al constructor de la classe mare i seguidament escrigui per pantalla "Sóc el constructor de la classe xxx". Després crea una classe executora (amb un main) dins el mateix package, crea objectes de cada classe i observa què es mostra per pantalla en cada creació.



4- Biblioteca IImplementació herència

L'objectiu d'aquesta activitat és implementar un seguit de classes en Java relacionades mitjançant herència. Implementeu les classes Fitxa, Obra, Volum i Revista corresponents a la figura següent, pensades per poder gestionar alguns dels elements que ens podem trobar en la gestió d'una biblioteca.



El significat dels atributs que puguin induir a confusió són:

nrePags: nombre de pàgines de l'obra
nro: número de revista o número de volum

Aquestes classes han de contenir, com a mínim, els mètodes següents:

Constructors adequats, que han de permetre crear objectes de les classes corresponents.
Mètodes per obtenir els valors dels atributs.
Mètodes per modificar els valors dels atributs.
El mètode toString(), que ha de mostrar la informació de l'objecte actiu per la consola.
La sobreescriptura del mètodes equals() heretat de la classe Object.

Cal considerar que dues fitxes són iguals si tenen la mateixa referència. Les classes han de residir en un paquet anomenat biblioteca. Desenvolupeu una altra classe, anomenada ProvaFitxes, que contingui un mètode main() que comprovi la gestió correcta de les classes Fitxa i derivades implementades. La presència d'aquesta classe no prohibeix que cada classe tingui el propi mètode main() per comprovar-ne el funcionament.

5. Biblioteca II –Gestió d'Objectes dins una jerarquia d'Herència

L’objectiu d’aquesta activitat és veure com es gestionen diferents objectes quan aquests pertanyen a diferents classes vinculades mitjançant herència. Considereu la implementació de la jerarquia de classes corresponent a l'activitat anterior.

Dissenyeu una classe anomenada Biblioteca que permeti la gestió d’una biblioteca a partir d’una taula d’objectes de les classes derivades de la classe Fitxa.

Els requeriments són els següents:

En la creació de la biblioteca cal indicar la seva dimensió.
En una biblioteca no hi pot haver dues fitxes amb la mateixa referència.
Ha de proporcionar mètodes per:

1. Conèixer la capacitat de la biblioteca.
2. Conèixer el nombre d’elements que hi ha en la biblioteca.
3. Afegir una fitxa a la biblioteca.
4. Extreure una fitxa a partir de la seva referència.
5. Proporcionar la fitxa que es troba en una posició determinada en la biblioteca.
6. Visualitzar el contingut de la biblioteca.

Desenvolupeu una altra classe, anomenada ProvaBiblioteca, que contingui un mètode main() que comprovi la gestió correcta de la classe Biblioteca implementada.

6. Empleats

Tenim un sistema on controlem els treballadors que hi ha a la nostra empresa. De l'Empleat guardarem el nom, el cognom, l'edat i el salari. A més de la constructora per defecte i la constructora per paràmetres i dels seus setters&getters, un empleat també tindrà:

boolean plus(int plusSalarial). Aquest mètode ha de ser capaç de sumar una quantitat que entrarem com a paràmetre si l'empleat té més de 40 anys. Retorna una variable booleana per tal d'informar si s'ha apujat el seu salari o no.

boolean comprovaNom(): ens comprova que el nom no estigui buit.

Després tindrem un tipus d'empleat que és un Comercial, d'un comercial guardarem la comissio que s'endú. Crea igualment constructores i gettersSetters.

A més tenim un empleat que serà el Repartidor. Aquest empleat en guardarem la zona per la qual reparteix el producte. Crea igualment constructores i gettersSetters.

Fes una classe executora, comprova la creació de diferents tipus d'objectes diferents empleat, comercial i repartidor. Compara classes, utilitza el getClass, instanceOf, prova la igualtat entre dos Objectes (equalsTo).