

Universitat de Lleida
Escola Politècnica Superior
Grau en Enginyeria Informàtica
Estructura de dades

Laboratori 2

Programació Orientada a Objectes

Laura Haro Escoi
Guillem Mora Bea
PraLab2

Dimarts 26 d'octubre de 2021

Exercici 1: Jerarquia de classes i equals

Apartat 1a

- ***Equals de la classe Person***

En aquesta funció tenim unes certes condicions per a poder verificar si els dos objectes que estem tractant són idèntics.

La primera compara directament tots dos objectes per a veure si són iguals en tots els aspectes. Si és així, retorna true.

La segona condició que trobem comprova si l'argument és null o si la classe de l'argument és diferent a la classe que ha cridat el mètode. Si una d'aquestes es compleix, retorna false.

Si cap de les condicions anteriors es compleix, passa l'objecte de l'argument a la classe del que l'ha cridat, en aquest cas, a la classe Person.

Finalment, retorna la condició que compara els atributs d'aquests objectes. En aquest cas, mira si cap dels dos noms són nulls i llavors, els compara.

- ***Equals de la classe Employee***

En aquesta funció, podem veure moltes similituds amb el equals de la superclasse Person. Les dues primeres condicions són idèntiques.

La primera compara directament tots dos objectes per a veure si són iguals en tots els aspectes. Si és així, retorna true.

La segona condició que trobem comprova si l'argument és null o si la classe de l'argument és diferent a la classe que ha cridat el mètode. Si una d'aquestes es compleix, retorna false.

Ara, a diferència del equals de Person, tenim una nova condició. Aquesta compara els atributs heretats de la seva superclasse. En aquest cas compararia els noms dels Employees, ja que són els que venen de la classe Person. En cas de que els noms siguin diferents, retorna false.

Si cap de les condicions anteriors es compleix, passa l'objecte de l'argument a la classe del que l'ha cridat, en aquest cas, a la classe Employee.

Finalment, retorna la condició que compara els atributs d'aquests objectes. En aquest cas compara els seus salaris.

Apartat 1b

- ***Equals de la classe Person***

En aquesta funció tenim unes certes condicions per a poder verificar si els dos objectes que estem tractant són idèntics.

La primera compara directament tots dos objectes per a veure si són iguals en tots els aspectes. Si és així, retorna true.

La segona condició que trobem comprova si l'objecte que estem comparant, és una subclasse o la mateixa classe que Person. Si això no es compleix, retorna false.

Si cap de les condicions anteriors es compleix, passa l'objecte de l'argument a la classe del que l'ha cridat, en aquest cas, a la classe Person.

Finalment, retorna la condició que compara els atributs d'aquests objectes. En aquest cas, mira si cap dels dos noms són nulls i llavors, els compara.

- ***Equals de la classe Employee***

En aquesta funció, podem veure moltes similituds amb el equals de la superclasse Person. Les dues primeres condicions són idèntiques.

La primera compara directament tots dos objectes per a veure si són iguals en tots els aspectes. Si és així, retorna true.

La segona condició que trobem comprova si l'objecte que estem comparant, és una subclasse o la mateixa classe que Employee. Si això no es compleix, retorna false.

Ara, a diferència del equals de Person, tenim una nova condició. Aquesta compara els atributs heretats de la seva superclasse. En aquest cas compararia els noms dels Employees, ja que són els que venen de la classe Person. En cas de que els noms siguin diferents, retorna false.

Si cap de les condicions anteriors es compleix, passa l'objecte de l'argument a la classe del que l'ha cridat, en aquest cas, a la classe Employee.

Finalment, retorna la condició que compara els atributs d'aquests objectes. En aquest cas compara els seus salaris.

- ***Diferències entre els equals de 1a i 1b***

La diferència principal és que l'apartat 1a compara que els dos objectes siguin de la mateixa classe, sense cap excepció. En canvi, en l'apartat 1b compara que els dos objectes siguin de la mateixa classe o que estiguin relacionats per herència. Això ho podem fer gràcies al mètode *instance of*.

Apartat 1c

El mètode equals de la classe Employee es diferencia amb el equals dels apartats anteriors per lo següent:

- Quan fem el *instance of* el fem sobre la classe Person, ja que aquesta funció comprova que l'objecte donat sigui una subclasse o de la mateixa classe que li hem dit i, si ho fem sobre la classe Employee, la funció perd simetria.
- Hem afegit una altra condició en la qual si l'objecte és de la classe Person, retorna true. Això ho fem per a que si comparem dos Employees, es pugui comprovar la igualtat entre els salaris. A part, si comparem un Employee amb un Person, sol agafi els atributs heretats.

Resolen aquest apartat, hem provat varies opcions fins a trobar aquesta. Hem provat de treure el *instance of* ja que, primerament, si ho feia sobre Employee, no podíem tenir en compte les superclasses però vam veure que necessitàvem aquesta condició per a passar els testos.

Exercici 2: Iteradors i interfícies

La funció que hem implementat en aquest apartat és la següent:

```
public static <E> void removeIf (Iterator<E> it, Check<E> test)
```

Li passem com a paràmetres un iterador, amb el qual recorrerem la llista d'objectes, i una interfície anomenada Check, la qual comprovarà si l'element és parell o no.

Per a poder recórrer la llista, hem utilitzat un bucle while amb la funció it.hasNext(), la qual comprova si la llista té més elements a tractar. Una vegada dins el bucle, comprova si el nombre és parell o no. Si aquesta condició es compleix, el borra de la llista amb la funció it.remove().

Exercici 3: Comparadors i comodins

Aquesta generalització que hem fet, funciona en tots els casos, ja que quan comparem les diferents classes, indiferentment si son subclasse o la seva superclasse, utilitza el atribut heretat, el qual sempre tenen en comú.

Per exemple, si agafes una llista de Employees i li pases una Person, comparará els atributs Employee.name amb el Person.name, que és el atribut heretat de Person a Employee.

Això també funcionaria al revés i també si fossin de la mateixa classe.

Tot això es pot fer ja que hem utilitzat un comodí a la classe Comparisons, concretament a la seva funció *removeLower*, on qualsevol classe que tu li passis, la extindrà a Person, per tant sempre tindrà el atribut heretat *name*.