



UF4.NF2 - 1. Desenvolupament de programes organitzats en classes

CFGS Desenvolupament d'Aplicacions Multiplataforma

Mòdul 3: Programació

Professors: Ximo Arenós

Sergi Pérez

Curs: 2015-2016

Activitats

Índex

1. Classe CEmpleat	4
10. Membres de classe	4
11. Unió de tots els conceptes	5
2. Aplicació CTestEmpleat	4
3. Mètodes sobrecarregats	4
4. La classe CCotxe	2
5. Referència this	3
6. Constructors 1	3
7. Constructors 2	3
8. Mètode toString	4
9. Mètode equals	4

1. Classe CEmpleat

(1punt)

1. Crear una classe anomenada CEmpleat que tingui els següents atributs i mètodes:

- Tots els atributs seran públics.
- Implementar el mètode incrementar salari.

Atributs (escollir el tipus de dades més adient):

- | | |
|-----------------|------------------|
| ▪ Nom | ▪ Nombre fills |
| ▪ Primer Cognom | ▪ Data naixement |
| ▪ Segon Cognom | ▪ Salari |
| ▪ Estat civil | ▪ Categoria |

Mètodes:

- Incrementar el salari a partir de l'empleat. El mètode tindrà com a paràmetre d'entrada el percentatge que incrementarà el salari. No retornarà res.

2. La classe CEmpleat a més ha de contenir el mètode main perquè serà l'aplicació. Aquest mètode main ha de fer:

- Crear un objecte de la classe CEmpleat.
- Assignar dades per a tots els atributs de l'objecte.
- Mostrar per consola: Nom, cognoms i salari de l'objecte, fent servir una frase que tingui sentit.
- Incrementar el salari de l'objecte un 10%.
- Mostrar per consola: Nom, cognoms i salari de l'objecte, fent servir una frase que tingui sentit.

2. Aplicació CTestEmpleat

(1 punt)

1. Continuar amb la classe CEmpleat creada a l'exercici anterior. Modificar els atributs de la classe per a que tots siguin privats. Afegir els mètodes get/set de cada atribut.

2. Crear la classe CTestEmpleat amb el mètode main per a que sigui l'aplicació i per tant s'ha de treure aquest mètode de la classe CEmpleat. El mètode main ha de fer el mateix que feia a l'exercici 1.

3. Mètodes sobrecarregats

(1 punt)

1. Continuar amb les classe CEmpleat i CTestEmpleat de l'exercici anterior. Crear un nou mètode incrementar salari (classe CEmpleat) que incrementi el salari en funció del nombre de fills de l'empleat. Els mètodes han d'estar sobrecarregats:

- Incrementar salari 1: només rep com a paràmetre el percentatge de pujada i incrementa el salari segons aquest percentatge. (implementat a l'exercici 1).
- Incrementar salari 2: rep dos paràmetres el percentatge de pujada i el nombre de fills. Aquests dos paràmetres es sumen i s'incrementa el salari amb el resultat d'aquesta suma.

2. Modificar el mètode main (classe CTestEmpleat) per a que la crida al mètode sobrecarregat incrementar salari es faci en funció del nombre de fills de l'empleat.

4. La classe CCotxe

(2 punts)

1. Realitzar el model UML d'una classe anomenada CCotxe que tingui els següents atributs i mètodes:

Atributs (escollir el tipus de dades més adient).

- Marca
- Model
- Motor
- Número de marxes (1-5)
- Número de portes (3 o 5)
- Color
- Estat (true = en marxa, false = parat)
- Marxa Actual
- Velocitat Actual
- Velocitat Maxima (200 km/h)
- Velocitat Minima (0 km/h)

Mètodes:

- Posar en marxa: No té cap paràmetre d'entrada. Retorna la velocitat actual.
El mètode ha de comprovar que el cotxe no estigui en marxa. Si el cotxe no està en marxa, canvia el seu estat a true i la velocitat inicial del cotxe serà 10 km/h. Si el cotxe ja està en marxa, no canvia res.
- Accelerar: Té com a paràmetre d'entrada la força amb la que s'accelera (un número entre 0 i 1). Retorna la velocitat actual.
El mètode ha de comprovar que el cotxe estigui en marxa. Si el cotxe està en marxa, s'accelerará sumant-li a la velocitat actual el resultat de multiplicar la velocitat actual i la força amb la que s'accelera. Els canvis de marxa van de la següent forma:

Primera: velocitat < 20 km/h.

Segona: 20 km/h >= velocitat < 40 km/h.

Tercera: 40 km/h >= velocitat < 60 km/h.

Quarta: 60 km/h >= velocitat < 80 km/h.

Cinquena: velocitat >= 80 km/h.

Si el cotxe no està en marxa, es mostra un missatge d'error per pantalla.

La velocitat resultant de l'acceleració no pot ser ni més gran que la velocitat màxima de l'objecte ni més petita que la velocitat mínima.

- Frenar: Té com a paràmetre d'entrada la força amb la que es frena (un número entre 0 i 1). Retorna la velocitat actual.

El mètode ha de comprovar que el cotxe estigui en marxa. Si el cotxe està en marxa, es frenarà restant-li a la velocitat actual el resultat de multiplicar la velocitat actual i la força amb la que s'accelera. Els canvis de marxa van de la següent forma:

Primera: velocitat < 20 km/h.

Segona: 20 km/h >= velocitat < 40 km/h.

Tercera: 40 km/h >= velocitat < 60 km/h.

Quarta: 60 km/h >= velocitat < 80 km/h.

Cinquena: velocitat > = 80 km/h.

Si el cotxe no està en marxa, es mostra un missatge d'error per pantalla.

La velocitat resultant de la frenada no pot ser ni més gran que la velocitat màxima de l'objecte ni més petita que la velocitat mínima.

- Canviar de marxa: No té cap paràmetre d'entrada. Retorna la marxa actual.

El mètode ha de comprovar que el cotxe estigui en marxa. Si el cotxe està en marxa, es canviarà de marxa en funció dels valors de velocitat establerts als mètodes anteriors.

Si el cotxe no està en marxa, es mostra un missatge d'error per pantalla.

- Apagar: No té cap paràmetre d'entrada. No retorna res.

El mètode ha de comprovar que el cotxe estigui en marxa. Si el cotxe està en marxa, canvia el seu estat a false i la velocitat del cotxe serà 0 km/h. Si el cotxe no està en marxa, no canvia res.

2. Després de fer el model UML, passar-lo a codi Java, creant la classe CCotxe:

- Tots els atributs seran privats.
- Afegir els mètode get/set que siguin necessaris. Heu d'estudiar cada atribut per veure en quins casos serà necessari només el set, o només el get, o tots dos.
- Implementar tots els mètodes anteriors.

3. Crear la classe CTestCotxe que serà la classe que tingui el main.

4. El mètode main ha de fer el següent:

- Crear un objecte de la classe CCotxe.
- Donar valor a tots els seus atributs (un valor que sigui vàlid i coherent).
- Posar l'objecte CCotxe en marxa.
- Mostrar l'estat, velocitat i marxa actual per pantalla (amb una frase que tingui sentit).
- Accelerar el cotxe.
- Mostrar l'estat, velocitat i marxa actual per pantalla (amb una frase que tingui sentit).
- Frenar el cotxe.
- Mostrar l'estat, velocitat i marxa actual per pantalla (amb una frase que tingui sentit).
- Tornar a posar l'objecte CCotxe en marxa.
- Mostrar l'estat, velocitat i marxa actual per pantalla (amb una frase que tingui sentit).
- Parar l'objecte CCotxe.
- Mostrar l'estat, velocitat i marxa actual per pantalla (amb una frase que tingui sentit).

- Accelerar l'objecte CCotxe.
- Mostrar l'estat, velocitat i marxa actual per pantalla (amb una frase que tingui sentit).

5. Referència this

(1 punt)

1. Continuar amb les classe CEmpleat i CTestEmpleat de l'exercici 3. Modificar les classes per a que facin servir la referència this en tots aquells punts on sigui necessari.

6. Constructors 1

(1 punt)

1. Continuar amb les classe CEmpleat i CTestEmpleat de l'exercici anterior. Afegir a la classe CEmpleat 3 constructors:

- Constructor que tingui tants paràmetres com té la classe i els inicialitzi. S'ha de fer servir la referència this.
- Constructor sense paràmetres que cridi al constructor anterior, inicialitzant tots els atributs a valors per defecte.
- Constructor còpia. El cos d'aquest constructor serà una crida al primer constructor creat en aquest punt.

2. Modificar la classe CTestEmpleat:

- Crear un objecte de la classe CEmpleat fent servir el constructor amb paràmetres. Aquest objecte ha de tenir 0 fills.
- Crear un segon objecte CEmpleat fent servir el constructor còpia a partir de l'objecte creat al punt anterior. Aquest objecte ha de tenir 3 fills.
- Mostrar per consola: Nom, cognoms i salari dels 2 objectes, fent servir una frase que tingui sentit.
- Incrementar el salari dels 2 objectes un 10%.
- Mostrar per consola: Nom, cognoms i salari dels 2 objectes, fent servir una frase que tingui sentit.

7. Constructors 2

(1 punt)

1. Continuar amb la classe CCotxe i CTestCotxe creades a l'exercici 4. Afegir a la classe CCotxe 3 constructors (Aquests són els 3 constructors que han d'aparèixer a qualsevol classe):

- Constructor per defecte.
- Constructor que tingui tants paràmetres com té la classe i els inicialitzi. S'ha de fer servir la referència this.
- Constructor còpia. El cos d'aquest constructor serà una crida al primer constructor creat en aquest punt.

2. Estudiar la llista de mètodes get/set de la classe CCotxe i eliminar tots aquells que no siguin necessaris.
3. Modificar la classe CTestCotxe per a que l'objecte creat de la classe CCotxe es creï i inicialitzi fent servir el constructor.

8. Mètode toString

(1 punt)

1. Continuar amb les classe CEmpleat i CTestEmpleat de l'exercici anterior. Afegir a la classe CEmpleat el mètode toString que retorni la concatenació de nom, cognoms i salari de l'objecte, fent servir una frase que tingui sentit.
2. Modificar la classe CTestEmpleat per a que en compte de mostrar per consola les dades dels objectes creant la frase en el mètode println, faci una crida al mètode toString, creat a la classe CEmpleat.

9. Mètode equals

(1 punt)

1. Continuar amb les classe CEmpleat i CTestEmpleat de l'exercici anterior. Afegir a la classe CEmpleat el mètode equals que compari quan dos objectes CEmpleat són iguals. Dos objectes de la classe CEmpleat són iguals quan el valor de tots els seus atributs són iguals.
2. Modificar la classe CTestEmpleat. Fer la comparació dels dos objectes CEmpleat just després de la seva creació i mostrar per pantalla si són iguals o diferents.

10. Membres de classe

(1 punt)

1. Continuar amb les classe CEmpleat i CTestEmpleat de l'exercici anterior. Afegir a la classe CEmpleat dos atributs nous (privats):
 - Atribut de classe numTreballadors que vagi contant el nombre d'objectes CEmpleat que es van creant, inicialitzat a 1.
 - Atribut d'instància (idTreballador) de tipus String que assigni un valor únic per a cada treballador. Aquest valor estarà format pel número de treballadors concatenat amb un guió i amb la data de naixement de l'empleat calculada en milisegons (buscar el mètode adequat a la classe GregorianCalendar).
2. Modificar els constructors de la classe CEmpleat per a que assignin l'idTreballador al nou objecte i augmentin el nombre de treballadors.
3. Afegir a la classe CEmpleat dos mètodes (get) per poder obtenir l'idTreballador i el numTreballadors.

4. Modificar les cadenes que es mostren a la classe CTestEmpleat:

- idTreballador, nom, cognoms i salari dels 2 objectes, fent servir una frase que tingui sentit.
- numTreballadors al final de tot.

11. Unió de tots els conceptes

(5 punts)

1. Crear una classe CPersona:

- Atributs: nom, primer cognom, segon cognom i número secret.
- Escollir el tipus més apropiat per a cada atribut.
- El número secret està format per 4 xifres numèriques.
- Afegir 2 constructors:
 - Constructor que tingui tants paràmetres com té la classe i els inicialitzi. S'ha de fer servir la referència this.
 - Constructor còpia.
- Afegir els mètodes get/set pels atributs.

2. Crear la classe CCompteBancari:

- Atributs d'instància: número, propietari, saldo.
- Atributs de classe: següent compte i saldo inicial.
- Escollir el tipus més apropiat per a cada atribut. El propietari ha de ser de tipus CPersona.
- El següent compte ha de tenir un valor inicial: 1234.
- El saldo inicial ha de tenir un valor inicial: 100.
- El número de compte està format per 4 xifres numèriques.
- Afegir 2 constructors:
 - Constructor que tingui tants paràmetres com té la classe i els inicialitzi. S'ha de fer servir la referència this. S'han de fer servir els atributs de classe per inicialitzar els atributs que correspongui. Els atributs de classe s'han d'actualitzar correctament.
 - Constructor còpia
- Afegir els mètodes get/set per l'atribut propietari i el mètode get per l'atribut número.
- Afegir un mètode static per modificar el saldo inicial.
- Afegir el mètode consultar saldo: retorni el saldo del compte.
- Afegir el mètode comprovar número secret:
 - Paràmetres: dos nombres enters que siguin números secrets de comptes.
 - Retorn: true si els dos nombres són iguals i false si són diferents.

- Afegir el mètode ingressar:
 - Paràmetres: quantitat a ingressar i número secret.
 - Retorn: true si l'ingrés ha anat bé, false si el número secret no és correcte.
 - El mètode primer comprova que el número secret coincideix amb el del compte i després fa l'ingrés.
- Afegir el mètode extreure:
 - Paràmetres: quantitat a extreure i número secret.
 - Retorn: true si l'extracció ha anat bé, false si el número secret no és correcte.
 - El mètode primer comprova que el número secret coincideix amb el del compte i després fa l'extracció.

3. Crear la classe CAplicació:

- Crear un compte bancari (les dades les escolliu vosaltres).
- Introduir per pantalla la quantitat a ingressar i el número secret del compte (fer servir la classe CLlegir).
- Fer l'ingrés al compte i mostrar per pantalla:
 - Si l'operació ha estat correcte: el saldo final.
 - Si el número secret és erroni: un missatge que informi d'això.
- Introduir per pantalla la quantitat a extreure i el número secret del compte (fer servir la classe CLlegir).
- Fer l'extracció i mostrar per pantalla:
 - Si l'operació ha estat correcte i el saldo és positiu: el saldo final.
 - Si l'operació ha estat correcte i el saldo és negatiu: un missatge que informi d'això i el saldo final.
 - Si el número secret és erroni: un missatge que informi d'això.