

IIP Primer Parcial - ETSInf

9 de Novembre de 2016. Duració: 1 hora i 30 minuts.

Nota: L'examen s'avalua sobre 10 punts, però el seu pes específic en la nota final d'IIP és de 2,4 punts

1. 1 punt Es disposa de la classe **Punt** que defineix un punt en un espai bidimensional real mitjançant els atributs **x** i **y** (representant abscissa i ordenada, respectivament), amb la funcionalitat que, en part, es mostra, a continuació, en la seua documentació:

Constructor Summary	
Constructors	
Constructor and Description	
Punt()	Crea un Punt amb coordenades (0.0, 0.0).
Punt(double abs, double ord)	Crea un Punt amb coordenades (abs, ord).

Method Summary	
All Methods	Instance Methods
Concrete Methods	
Modifier and Type	Method and Description
double	distancia(Punt p) Torna la distancia del Punt en curs a p.
boolean	equals(java.lang.Object o) Torna true si o es un Punt amb les mateixes coordenades que el Punt en curs i torna false en cas contrari.
java.lang.String	toString() Torna un String amb les dades del Punt en curs en el format (x, y).

Es demana: Implementar el mètode **distancia** tal que torne la distància del **Punt** en curs a un altre **Punt p** donat, **arrodonint** el resultat a **quatre** decimals. Es recorda que la distància entre dos punts (a_1, b_1) i (a_2, b_2) es calcula com $\sqrt{(a_1 - a_2)^2 + (b_1 - b_2)^2}$.

Solució:

```
public double distancia(Punt p) {  
    double abs = this.x - p.x;  
    double ord = this.y - p.y;  
    double dist = Math.sqrt(abs * abs + ord * ord);  
    return Math.round(dist * 10000) / 10000.0;  
    // o també: return Math.round(dist * Math.pow(10, 4)) / Math.pow(10, 4);  
}
```

2. 6.5 punts Utilitzant la classe **Punt** descrita anteriorment, es volen representar els edificis del campus de Vera de la UPV. Per a això, es vol dissenyar una classe Tipus de dades anomenada **Edifici** que contindrà informació física sobre la construcció (coordenades GPS, codi d'identificació en un pla), així com de l'ús assignat a l'edifici (tipus d'ús i nom de l'entitat que està fent ús del mateix).

Es demana: implementar la classe **Edifici** amb els atributs i mètodes que s'indiquen a continuació:

a) (0.5 punts) Atributs de classe públics i constants de tipus enter:

- **DEPARTAMENT**, amb valor 0 que representa el tipus d'edifici departamental.
- **ESCOLA**, amb valor 1 que representa el tipus d'edifici dedicat a docència com aularis, escoles o laboratoris.
- **SERVICIS**, amb valor 2 que representa el tipus d'edifici per a altres activitats com cafeteries, oficines, etc.

Aquestes constants han de ser utilitzades sempre que es requereixca (tant a la classe `Edifici` com a la classe `GestorEdificis`).

- b) (0.5 punts) Atributs d'instància privats `codi` (`String`), `entitat` (`String`), `tipus` (`int`) i `coordenades` (`Punt`).
- c) (1.5 punts) Dos constructors:
- Un constructor general amb els paràmetres apropiats (un d'ells de tipus `Punt`) per inicialitzar tots els atributs d'instància. Supposeu que les dades són correctes.
 - Un constructor per defecte que creï un edifici departamental usat per l'entitat `DSIC`, amb codi `1F` i en les coordenades `(39.4625, -0.3472)`.
- d) (0.5 punts) Un mètode consultor i un mètode modificador de l'atribut `coordenades`. Supposeu que el valor del paràmetre del modificador és correcte.
- e) (1 punt) Un mètode `equals` (que sobreesciu el d'`Object`) per comprovar si dos edificis són iguals tenint en compte les dades de la construcció i independentment de l'ús, és a dir, si tenen el mateix codi i les mateixes coordenades. Cal notar que un edifici pot usar-se al mateix temps per dues entitats, per exemple, l'edifici `1G` per l'`ETSINF` i el `DISCA`.
- f) (1 punt) Un mètode `toString` (que sobreesciu el d'`Object`) perquè, usant obligatòriament una instrucció `switch`, torne el resultat amb un format com el mostrat en els següents exemples (per a les coordenades GPS es disposa del mètode `toString` de la classe `Punt`):
- ```
Edifici departamental 1F (DSIC), GPS: (39.4625, -0.3472)
Edifici departamental 1G (DISCA), GPS: (39.4826, -0.3470)
Edifici docent 1G (ETSINF), GPS: (39.4826, -0.3470)
Edifici de servicis 3N (Cafeteria BBAA), GPS: (39.4841, -0.3443)
```
- g) (1.5 punts) Un mètode `mesPropDeRectorat` que, donat un `Edifici` passat com paràmetre `e`:
- Si la distància de l'edifici `this` a les coordenades de rectorat `(39.4823, -0.3457)` és menor que la de l'edifici `e`, torna `-1`.
  - Si, pel contrari, la distància de l'edifici `this` a rectorat és major que la de l'edifici `e`, torna `1`.
  - Si ambdues distàncies són iguals es considera que:
    - Si els tipus d'edifici són diferents es consideren més prop els de tipus servici, després docents i, finalment, departamentals, tornant `-1` o `1` segons el cas. Per exemple, si l'edifici `this` és el `DISCA` i l'edifici `e` és l'`ETSINF` (edificis departamental i docent, respectivament, a la mateixa distància de rectorat), el mètode tornarà `1` indicant que l'`ETSINF` està més prop de rectorat que el `DISCA`. Però si `this` és l'`ETSINF` i `e` és el `DISCA`, tornarà `-1`.
    - Si els tipus són iguals, torna `0`.

### Solució:

```
public class Edifici {
 public static final int DEPARTAMENT = 0;
 public static final int ESCOLA = 1;
 public static final int SERVICIS = 2;

 private String codi, entitat;
 private int tipus;
 private Punt coordenades;
```

```

public Edifici(String c, String e, int t, Punt p) {
 codi = c;
 entitat = e;
 tipus = t;
 coordenades = p;
}

public Edifici() {
 this("1F", "DSIC", DEPARTAMENT, new Punt(39.4625, -0.3472));
}

public Punt getCoordenades() { return coordenades; }

public void setCoordenades(Punt p) { coordenades = p; }

public boolean equals(Object o) {
 return o instanceof Edifici
 && codi.equals(((Edifici) o).codi)
 && coordenades.equals(((Edifici) o).coordenades);
}

public String toString() {
 String res = "Edifici ";
 switch (tipus) {
 case DEPARTAMENT:
 res += "departamental "; break;
 case ESCOLA:
 res += "docent "; break;
 case SERVICIS:
 res += "de servicis "; break;
 }
 res += codi + " (" + entitat + "), GPS: " + coordenades;
 return res;
}

public int mesPropDeRectorat(Edifici e) {
 Punt rectorat = new Punt(39.4823, -0.3457);
 double distThis = coordenades.distancia(rectorat);
 double distE = e.coordenades.distancia(rectorat);
 int resultat = 0;
 if (distThis < distE) { resultat = -1; }
 else if (distThis > distE) { resultat = 1; }
 else if (tipus < e.tipus) { resultat = 1; }
 else if (tipus > e.tipus) { resultat = -1; }
 return resultat;
}
}

```

3. 2.5 punts **Es demana:** implementar la classe Programa GestorEdificis amb un mètode main que realitzi les següents accions:

- a) (1 punt) Crear un objecte e1 de tipus Edifici usant el constructor general, per a representar a un edifici docent usat per l'entitat ETSINF, amb codi 1G i en les coordenades (39.4826, -0.3470).
- b) (0.5 punts) Crear un objecte e2 de tipus Edifici usant el constructor per defecte.

- c) (1 punt) Invocar al mètode `mesPropDeRectorat` per a comparar `e1` amb `e2` i, a continuació, mostrar per pantalla l'edifici més proper a rectorat després del literal "L'edifici més proper a rectorat és " o, si estan igual de prop, el missatge "Els dos edificis estan igual de prop de rectorat".

### Solució:

```
public class GestorEdificis {
 public static void main(String[] args) {
 Edifici e1 = new Edifici("1G", "ETSINF", Edifici.ESCOLA,
 new Punt(39.4826, -0.3470));
 Edifici e2 = new Edifici();

 int comp = e1.mesPropDeRectorat(e2);
 String res = "L'edifici més proper a rectorat és ";
 if (comp == -1) { res += e1; }
 else if (comp == 1) { res += e2; }
 else { res = "Els dos edificis estan igual de prop de rectorat"; }
 System.out.println(res);
 }
}
```