Programació orientada a objectes (I)

Exercicis (II)

CONSTRUCTORS

El constructor és el mètode que s'executa quan s'instancia un objecte. Si no està definit, Java executarà un constructor per defecte que crearà l'objecte i inicialitzarà totes les seves variables a zero, però aquesta no és una bona pràctica. És millor definir nosaltres un constructor que controle què ha de passar quan es cree l'objecte.

En aquest apartat has de modificar els programes de l'apartat anterior (o fes una còpia del projecte si ho prefereixes) **i fer els canvis indicats.**

Exercici B1 – Punt

Afegeix a la classe Punto un constructor amb paràmetres que copiï les coordenades passades com a argument als atributs de l'objecte. Així:

```
public Punt(int x, int y){
this.x = x;
this.y = y;
}
```

Copiem els valors passats com a argument als atributs de l'objecte. Tingueu en compte que int x i int i són variables locals del mètode, NO són els atributs de l'objecte. Per fer referència als atributs de l'objecte cal utilitzar this.

Fixa't que ja no serà possible fer Punt p = new Punt(). Ara serà obligatori fer, per exemple, Punt p = new Punt(2, 7). A l'apartat A havies de recordar assignar valors a x i i després de crear un punt, cosa que no és una bona idea en projectes grans amb centenars d'objectes (és molt fàcil equivocar-se). Ara és impossible equivocar-se perquè el Java no et deixarà. Hem assegurat que tots els punts sempre tindran coordenades.

Corregeix el main i utilitza el constructor amb paràmetres per instanciar els objectes, passant-li com a argument els valors desitjats.

Exercici B2 – Persona

Afegeix a Persona el constructor de baix i corregeix el main per utilitzar-lo:

```
public Persona(String dni, String nom, String cognoms, int edat) {
        this.dni = dni;
        this.nom = nom;
        this.cognoms = cognoms;
        this.edat = edat;
}
```

Tingueu en compte que no és obligatori que els paràmetres del constructor s'anomenen igual que els atributs de l'objecte (en aquest cas no caldria utilitzar this). Podríem fer-ho així:

```
public Persona(String id, String nom, String ap, int e) {
        dni = id;
        nom = nom;
        cognoms = ap;
        edat = e;
}
```

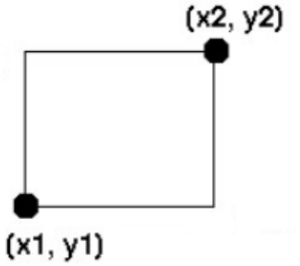
Tampoc no és obligatori passar al constructor tots els atributs de la classe. Podríem decidir per exemple que al nostre programari totes les persones han de tenir nom, cognoms i edat, però no és obligatori el DNI (nounats i nens). Aquest constructor també seria vàlid:

```
public Persona(String nom, String ap, int e) {
        nom = nom;
        cognoms = ap;
        edat = e;
}
```

Una classe pot tenir tants constructors com vulgues sempre que tinguen diferent nº i/o tipus de paràmetres (perquè no hi haja ambigüitat en què utilitzar).

Exercici B3 – Rectangle

Al nostre software necessitem assegurar-nos que la coordenada (x1,y1) representa la cantonada inferior esquerra i la (x2,y2) la superior dreta del rectangle, com en la imatge.



Afegeix a Rectangle un constructor amb els 4 paràmetres. Inclou un **if** que comprove els valors (*). Si són vàlids guardarà els paràmetres a l'objecte.

Si no ho són, mostrarà un missatge de l'estil "ERROR en instància Rectangle..." utilitzant System.err.println(…). No podrem evitar que l'objecte s'instancie però almenys avisarem per pantalla.

Corregeix el main per utilitzar aquest constructor. Hauríeu de mostrar un missatge d'error.

(*) Pista: És suficient amb un if ((condició) && (condició))

Exercici B4 – Article

Afegeix un constructor amb 4 paràmetres que assigne valors a **nom**, **preu**, **iva** i **quantsQueden**. Aquest constructor haurà de mostrar un missatge d'error si algun dels valors nom, preu, iva o **quantsQueden** no són vàlids. Quines condicions creus que podrien determinar si són vàlids o no? Raona-ho i implementa el codi.

Corregeix el main i prova a crear diversos articles. Introduïu-ne alguns amb valors incorrectes per comprovar si avisa de l'error.