

[🏠](#) > [Els Meus...](#) > [Programació-1CFSC LFP 1...](#) > [UT5. INTRODUCCIÓ A LA PROGRAMACIÓ ORIENT...](#) > [Exercicis per a p...](#)

## Exercicis per a practicar

Proposta exercicis:

- Desenvolupar un programa que carregue els costats d'un triangle i implemente els mètodes següents:
  - Constructor que inicialitze tots els atributs.
  - mètode que torna el valor del costat major
  - mètode que torna l'àrea del triangle
  - mètode que mostre quin tipus de triangle és (equilàter, isòsceles, escalè).
- Desenvolupar una classe que represente un punt en el pla i tinga els mètodes següents:
  - Constructor que inicialitze els valors de x i y (en eixe ordre)
  - Constructor buit que inicialitzarà x=0 i y=0.
  - Mètode que imprimeix per pantalla en quin quadrant es troba el punt (concepte matemàtic, primer quadrant si x i y són positives, si x<0 i y>0 segon quadrant, etc.)
- Crea la classe Pizza amb els atributs i mètodes que calguen. Sobre cada pizza volem saber el tamany (mitjana o familiar), el tipus (prosciutto, quatre formatges o funghi) i el seu estat (demanada o servida). La classe ha d'emmagatzemar informació sobre el nombre total de pizzas que s'han demanat i s'han servit. Sempre que es crea una nova pizza el seu estat serà "demanada". El següent codi del programa principal funcionarà com es mostra a continuació:

```
public class ComandesPizza {  
    public static void main(String[] args) {  
        Pizza p1 = new Pizza("prosciutto", "mitjana");  
        Pizza p2 = new Pizza("funghi", "familiar");  
        p2.servir();  
        Pizza p3 = new Pizza("quatre formatges", "mitjana");  
        System.out.println(p1);  
        System.out.println(p2);  
        System.out.println(p3);  
        p2.servir();  
        System.out.println("demanades: " + Pizza.getTotalDemanades());  
        System.out.println("servides: " + Pizza.getTotalServides());  
    }  
}
```

```
pizza prosciutto mitjana, demanada  
pizza funghi familiar, servida  
pizza quatre formatges mitjana, demanada  
eixa pizza ja s'ha servit  
demanades: 3  
servides: 1
```

- Transforma l'exercici anterior per a que funcione amb enumerats (Tipus, Tamany i Estat).
- Construeix una classe *Complex* amb dos atributs:

- `public Complex()`: Constructor que inicialitza els atributs a zero.
- `public Complex(double real, double imag)`: Constructor que inicialitza els atributs als valors indicats pels paràmetres.
- `public double getReal()`: Retorna la part real de l'objecte.
- `public double getImaginaria()`: Retorna la part imaginària de l'objecte.
- `public void setReal(double real)`: Assigna a la part real de l'objecte el valor indicat al paràmetre real.
- `public void setImaginaria(double imaginària)`: Assigna a la part imaginària de l'objecte el valor indicat en el paràmetre imaginària.
- `public String toString()`: Converteix a String el nombre complex, mitjançant la concatenació dels seus atributs i torna com a resultat la cadena de text  $3 + 4i$ , si 3 és la part real i 4 la part imaginària.
- `public Complex sumar(Complex b)`: Suma la part real amb la part real del nombre complex b, i la part imaginària amb la part imaginària del nombre complex b.
- `public Complex producte(Complex b)`: Calcula el producte del nombre complex multiplicat per b. Si un nombre té els components  $x_1$  i  $y_1$ , i el segon nombre té els components  $x_2$  i  $y_2$ : La part real del producte =  $x_1 * x_2 - y_1 * y_2$  La part imaginària del producte =  $x_1 * y_2 + x_2 * y_1$

· **Compte bancari:** Escriu un programa que simule un compte bancari. Un mètode permet fer un dipòsit al compte. La quantitat s'introdueix com a paràmetre. Un segon mètode permet fer una retirada (si l'estat és "DESCOBERT" no es permetrà fer més retirades). El DNI del titular, el saldo i l'estat del compte s'han de mostrar per pantalla en invocar al mètode "mostrar": l'estat pot ser OK o DESCOBERT. Crea una classe anomenada Compte per a representar comptes bancaris. Has de tenir els mètodes dipositar i retirar, `getSaldoActual` i `mostrar`.

· **Daus:** Dissenya i escriu una classe que actue com un dau, el qual es pot llançar per obtenir un valor de l'1 al 6. Escriu la classe de manera que al principi s'obtinga el valor 6. Escriu un programa per a crear un objecte i utilitzar-lo. La pantalla ha de mostrar una opció que faça que es llance el dau i se'n mostre el valor. Després modifiqueu la classe "Dau" de manera que retorne un valor que siga un punt més gran al que tenia l'última vegada que es va llançar; per exemple, 4 quan era un 3. Quan l'últim valor siga 6, el nou valor serà 1. Després modifiqueu la classe perquè utilitze el generador de números aleatoris de la biblioteca. Alguns jocs com el backgammon necessiten dos daus. Escriu les instruccions de Java per a crear dos instàncies de l'objecte "Dau", llançar els daus i mostrar els resultats.

· Crea la classe **Temps** amb els mètodes suma i resta. Els objectes de la classe Temps són intervals de temps i es creen de la forma `Temps t = new Temps(1, 20, 30)` on els paràmetres que es passen al constructor són les hores, els minuts i els segons respectivament. Crea el mètode `toString` per vore els intervals de temps de la forma 10h 35m 5s. Si se sumen per exemple 30m 40s i 35m 20s el resultat hauria de ser 1h 6m 0s. Realitza un programa de prova per comprovar que la classe funciona bé.

Última modificació: Friday, 6 December 2024, 6:30 PM

< Projecte "Taller d'ITV" - 1. Introducció a la POO

Salta a...

Vídeo - Proposta de solució exercicis UT5 >



---

[Avís legal](#) | [Política de Privacitat](#) | [Política de Cookies](#) | [Contacte](#) | [Accessibilitat](#)



**UNIÓ EUROPEA**  
Fons Europeu de Desenvolupament Regional