Pràctica 3 (P3): Composicions seqüencials, alternatives i iteratives bàsiques

Laboratori de Programació I - Curs 18-19
Grau d'Enginyeria Informàtica
Departament de Matemàtiques i Informàtica
Facultat de Matemàtiques i Informàtica

IMPORTANT: Abans d'escriure el codi, heu de pensar en els diferents *tests* de prova que especifiquen com s'ha de comportar el vostre programa. Això vol dir, donada una *Entrada* quina sortida s'espera del vostre programa. Una vegada definits els tests de prova, escriviu el programa Java i comproveu que la *Sortida Esperada* coincideixi amb la sortida obtinguda pel vostre programa. Més endavant, fareu tests unitaris (tests de prova de forma automàtica) amb *JUnit*. Fixeu-vos en els exemples de tests de prova que podeu trobar en els primers exercicis.

En alguns exercicis us proposem treballar amb dues classes, cada classe definida en un fitxer font .java diferent. Tingueu en compte que aquests dos fitxers font han d'estar emmagatzemats en el mateix directori. Aquest directori serà el contenidor (també anomenat package) [1] de les vostres classes, i així tindreu assegurat l'accès a atributs i mètodes d'una classe i de l'altre.

1. El següent codi Java defineix la classe **Rectangle** i l'utilitza a la classe principal **Exercici1** definint objectes (o instàncies) de la mateixa. Escriviu el codi corresponent als mètodes `getArea()` (línies 17-19) i `getPerimetre()` (línies 21-24) de la classe **Rectangle**.
   Les fórmules per calcular l'àrea i el perímetre són:

$$A = bh$$
$$L = 2b + 2h$$

   Prèviament a implementar el codi, heu de definir un conjunt de tests de prova. Per exemple, pel mètode `getArea()`:

Test	Entrada	Sortida Esperada
1	amplada 4.0 alçada 20.0	Àrea: 80.0
2	amplada 3.5 alçada 4.2	Àrea: 14.70

[1] Per a **crear una nova classe** en un projecte Netbeans, feu clic amb el botó dret del ratolí a l'icone que sembla una caixa groga a la finestra de Projectes. Aquesta nova classe i la classe principal (la del `main()`) pertenyen per defecte a un mateix paquet (package) que s'anomena igual que el projecte. Si treballeu per consola i no declareu cap paquet, les dues classes, a l'estar al mateix directori, pertenyen al paquet per defecte (default package).

Nota: En aquest exercici suposem que els valors d'amplada i alçada que s'utilitzen al construir els rectangles són positius majors que 0.
A continuació es mostra una plantilla de com heu d'especificar la taula de tests:
(`Exercici1.java`)(`Rectangle.java`)

```java
/*
 * Author:
 * Data:
 */

  /* Tests metode getArea()
 * Entrada       | Sortida Esperada
 * ----------------------------------
 * Amplada: 4.0   | 80.0
 * Alcada: 20.0   |
 * ----------------------------------
 * Amplada: 3.5   | 14.70
 * Alcada:  4.2   |
 * ----------------------------------
 */
 //Fer el mateix per els tests del metode getPerimetre()
 /* Tests metode getPerimetre()
 * Entrada       | Sortida Esperada
 * ----------------------------------
 *               |
 *               |
 * ----------------------------------
 *               |
 *               |
 * ----------------------------------
 */

public class Exercici1 {
  public static void main(String[] args) {
    Rectangle meu = new Rectangle(4.0, 20.0);
    System.out.println("L'area d'un rectangle d'amplada " +
      meu.amplada + " i alcada " +  meu.alcada + ": " + meu.getArea());
    System.out.println("El perimetre: " +  meu.getPerimetre());

    Rectangle teu = new Rectangle(3.5, 4.2);
    System.out.println("\nL'area d'un rectangle d'amplada " +
      teu.amplada + " i alcada " + teu.alcada + ": " + teu.getArea());
    System.out.println("El perimetre: " + teu.getPerimetre());
  }
}
```

```java
class Rectangle {
  // Atributs
  double amplada, alcada;

  // Constructor per defecte
  Rectangle() {
     amplada  = 1.0;
     alcada = 1.0;
  }

  // Constructor amb parametres
  Rectangle(double novaAmplada, double novaAlcada) {
    amplada = novaAmplada;
    alcada = novaAlcada;
 }
  //  Metode per obtenir l'area del rectangle
  double getArea() {
    //Introduiu solucio aqui
  }

  // Metode per obtenir el perimetre del rectangle
```

```
22      double getPerimetre() {
23       //Introduiu solucio aqui
24     }
25   }
```

2. Escriviu el mateix programa de l'exercici anterior però aquesta vegada que sigui l'usuari qui introdueixi per teclat valors d'amplada i alçada de dues instàncies (objectes) de la classe **Rectangle**. Utilitza la classe **Scanner**. (`Exercici2.java`)

3. Donat un radi r, escriviu un programa qua calculi l'àrea i el volum d'una esfera de radi r assegurant-se que r no sigui negatiu. Les fórmules per calcular l'àrea i el volum són:

$$A = 4\pi r^2$$

$$V = \frac{4}{3}\pi r^3$$

Escriviu el programa tenint en compte els següents tests:

Test	**Entrada**	**Sortida**
1	$r = 2.0(m)$	$A = 50.3(m^2)$ $V = 33.5(m^3)$
2	$r = 0.0(m)$	$A = 0.0(m^2)$ $V = 0.0(m^3)$
3	$r = -1.0(m)$	No es pot calcular l'àrea i el volum

No oblideu especificar la taula de tests al `Esfera.java` , tal com s'ha fet també a l'exercici 1 (ídem per a la resta d'exercicis d'aquesta llista).
(`Exercici3.java`) (`Esfera.java`).

```
1  /*
2     * Author:
3     * Data:
4     *
5     */
6
7     /* Tests metode getArea()
8    *
9    *
10   */
11   /* Tests metode getVolum()
12   *
13   *
14   */
15
16   public class Exercici3 {
17     public static void main(String[] args) {
18       Esfera meva;
19       meva = new Esfera(2);
20
21       // Printeu per pantalla un missatge que indiqui a l'usuari el radi de l'esfera,
22       //el volum i l'area
23     }
24   }
```

```
class Esfera {
  // Definiu aqui els atributs de l'esfera

  // Constructor amb parametres
  Esfera(double nouRadi) {
    radi = nouRadi;
  }
  //  Metode per obtenir l'area de l'esfera
  double getArea() {
    //Introduiu solucio aqui
  }

  // Metode per obtenir el  volum de l'esfera
  double getVolum() {
    //Introduiu solucio aqui
  }
}
```

- Penseu com usar PI, podeu usar la classe **java.lang.Math** https://docs.oracle.com/javase/7/docs/api/java/lang/Math.html, si el declaréssiu com la seva pròpia entitat on podríeu fer-ho?

4. El següent codi defineix i utilitza la classe **Persona**. Trobareu en aquesta classe el mètodes `calcularEdat()` i `calcularHoroscop()` que retornen l'edat de la persona, tenint en compte que estem a l'any 2018, i el seu horòscop respectivament. El mètode `calcularParitat()` ha de retornar si l'any en que va nèixer l'usuari era parell o senar.

   (a) Implementeu el mètode `int calcularEdat()`.

   (b) Implementeu el mètode `String calcularHoroscop()`. Per a la implementació, tingueu en compte la següent informació:

   ```
   21/03 - 20/04 - Aries
   21/04 - 21/05 - Tauro
   22/05 - 1/06 - Geminis
   22/06 - 22/07 - Cancer
   23/07 - 22/08 - Leo
   23/08 - 22/09 - Virgo
   23/09 - 22/10 - Libra
   23/10 - 22/11 - Escorpio
   23/11 - 21/12 - Sagitario
   22/12 - 20/01 - Capricornio
   21/01 - 19/02 - Acuario
   20/02 - 20/03 - Piscis
   ```

   (c) Implementeu el mètode `String calcularParidad()`.

   (d) Els mètodes `calcularEdat()`, `calcularHoroscop()` i `calcularParidad()` són mètodes de classe o mètodes d'objecte? Perquè? Contesteu aquesta pregunta cap al final del fitxer .java entre comentaris /* */.

   (e) Modifiqueu el codi de la classe `Exercici4.java` per tal de crear un nou objecte de la classe **Persona**, demaneu per teclat les dades necessàries per crear aquest nou objecte. La informació d'una persona s'ha de mostrar així:

   ```
   ** Hola "nom" **
   ```

```
La teva data de naixement es: "dd/mm/aaaa"
Tens "edat" anys
El teu signe es: "horoscopo"
Vas neixer en un any: "parell / senar"
```

Prèviament a implementar el codi, heu de definir un conjunt de tests de prova. Per exemple, pel mètode `calcularEdad()`:

Test	Entrada	Sortida
1	Any de naixement 1993	Edat: 25
2	Any de naixement 1990	Edat: 28

(Exercici4.java)(Persona.java)

```java
public class Exercici4 {

  public static void main(String[] args) {
        Persona p = new Persona("Anna",1994,22,9);
        System.out.println(p.nom + ",␣tens␣" + p.calcularEdat() + "␣anys");
        System.out.println("Horoscop:␣" + p.calcularHoroscop());
        System.out.println("Vas␣neixer␣un␣any:␣" + p.calcularParitat());

  }
}
```

```java
/*
*
* CLASSE Persona
* Author: Nom Cognom
* Data:
*
*/

/*
* Tests metode calcularEdat()
* Entrada      |  Sortida
* ----------------------------
* Any: 1993    |    24
* ----------------------------
* Any: 1990    |    27
* ----------------------------
*/
// Fer el mateix pels tests dels metodes calcularHoroscop() i calcularParitat()

public class Persona {
        // Atributs
        String nom;
        int naixement, mes, dia;

        // Constructor
        Persona(String nom, int naixement, int dia, int mes){
                this.nom = nom;
                this.naixement = naixement;
                this.mes = mes;
                this.dia = dia;
        }

        int calcularEdat(){
                //Introduiu solucio aqui
        }

        String calcularHoroscop(){
                //Introduiu solucio aqui
```

```
39          }
40
41          String calcularParitat(){
42                  //Introduiu solucio aqui
43          }
44
45  }
```

5. Donat el següent codi que defineix i utilitza la classe **Calendari**, escriu el codi que es demana a continuació.

   (a) Implementeu el mètode `String calcularDiumengePasqua(int any)`. Aquest mètode, donat un any, retorna un String amb la data del diumenge de Pasqua tenint en compte la següent informació:

   El diumenge de pasqua correspon al primer diumenge després de la primera lluna plena de la primavera. L'any 1800, el matemàtic Carl Friederich Gauss va presentar un algoritme per calcular la data del diumenge de pasqua i va anar fent modificacions fins a la seva proposta final l'any 1816. L'algorisme és el que s'indica a continuació:

   - := indica assignació
   - *div* indica divisió entera
   - *mod* indica mòdul
   - Y és l'any
   - M és el mes
   - D és el dia de pasqua de l'any indicat (Y).
   - $k := Y div 100$
   - $x := Y mod 19$
   - $b := Y mod 4$
   - $c := Y mod 7$
   - $q := k div 4$
   - $p := (13 + 8k) div 25$
   - $y := (15 – p + k – q) mod 30$
   - $z := (19x + y) mod 30$
   - $n := (4 + k – q) mod 7$
   - $e := (2b + 4c + 6z + n) mod 7$
   - Si $z + e \leq 9 \Rightarrow D := 22 + z + e$ i $M := 3$.

- D'altra banda, si $z = 29$ i $e = 6 \Rightarrow D := 19$ i $M := 4$.
- D'altra banda, si $z = 28$ i $e = 6$ i $x > 10 \Rightarrow D := 18$ i $M := 4$.
- Finalment, si no és cap de les anteriors, $D := z + e–9$ i $M := 4$.

(b) Implementeu el mètode `String comVaigDeTemps(int horaAra, int minutAra, int horaQuedada, int minutQuedada)`. Si l'hora actual supera l'hora de la quedada el mètode retorna "Vas tard", en cas contrari retorna "Vas bé de temps". *Nota:* la classe **Time** de Java et pot ajudar a resoldre aquest exercici! A continuació teniu l'enllaç a la API de Java: https://docs.oracle.com/javase/8/docs/api/java/time/package-summary.html

- Per crear un objecte de tipus Time amb una hora, minuts i segons exactes, heu de fer: `Time temps = new Time(hora, minuts, segons);`
- Podeu utilitzar funcions com `temps.before(hora)` i `temps.after(hora)` per saber si *temps* és inferior o superior a *hora* respectivament.

(c) Implementeu el mètode `boolean haPassatAniversari(int dia, int mes)`. Aquest mètode retorna true si el dia i el mes de l'aniversari superen el dia i el mes que té com a atributs la classe **Calendari.java**, ja que es considera que ha passat l'aniversari. En cas contrari, retorna false.

(d) Donat el número de dia de la setmana, implementeu el mètode `String diaDeLaSetmana(int numero)` perquè retorni el nom del dia de la setmana. Es considera que 1 fa referència a Dilluns i 7 a Diumenge. En el cas que l'usuari introdueixi un nombre fora d'aquest rang, s'ha d'indicar per pantalla.

(e) Implementeu el mètode `int diesPelCapDeSetmana(int num)`. Aquest mètode retorna quants dies queden perquè arribi el cap de setmana, tenint en compte la numeració comentada a l'exercici anterior.

Prèviament a implementar el codi, has de definir un conjunt de tests de prova, de la mateixa manera que els has definit en els exercicis anteriors.
(`Exercici5.java`) (`Calendari.java`)

```
1  public class Exercici5 {
2          public static void main(String[] args) {
3                  Calendari c = new Calendari(12,11,1993);
4          int horaAra = 16;
5                  int minutAra = 45;
6                  int horaQuedada = 16;
7                  int minutQuedada = 30;
8
9                  System.out.println("Al␣2020,␣diumenge␣de␣pasqua␣serà␣el␣" +
                       c.calcularDiumengePasqua(2020));
10
11                 System.out.println("Son␣les␣" + horaAra + ":" + minutAra);
12                 System.out.println("Has␣quedat␣a␣les␣" + horaQuedada + ":" +
                       minutQuedada);
13                 System.out.println("Per␣tant..." + c.comVaigDeTemps(horaAra,
                       minutAra, horaQuedada, minutQuedada));
14
15                 int diaAniversari = 12;
16                 int mesAniversari = 11;
17                 System.out.println("Ha␣passat␣el␣teu␣aniversari?␣" +
                       c.haPassatAniversari(diaAniversari,mesAniversari));
```

```
18
19                  int diaSetmana = 2;
20                  System.out.println("Som " + c.diaDeLaSemana(diaSemana));
21
22                  System.out.println("Queden " + c.diesPelCapDeSetmana(diaSetmana) + "
                        dies perque sigui cap de setmana");
23
24          }
25  }
```

```
1   public class Calendari {
2           // Atributs
3           int d, m, any;
4
5           // Constructor
6           Calendari(int d, int m, int any){
7                   this.d = d;
8                   this.m = m;
9                   this.any = any;
10          }
11
12          String calcularDiumengePasqua(int any){
13                  //Introduiu solucio aqui
14          }
15
16          String comVaigDeTemps(int horaAra, int minutAra, int horaQuedada, int
                minutQuedada){
17                  //Introduiu solucio aqui
18          }
19
20          boolean haPassatAniversari(int dia, int mes){
21                  //Introduiu solucio aqui
22          }
23
24          String diaDeLaSetmana(int num){
25                  //Introduiu solucio aqui
26          }
27
28          int diesPelCapDeSetmana(int num){
29                  //Introduiu solucio aqui
30          }
31  }
```

6. Donat el codi de la classe **Exercici6**, implementeu els mètodes que es demanen a continuació. No oblideu definir un conjunt de tests de prova.

   (a) `String dibuixaLinia(int n)`, de manera que dibuixi per pantalla una línia de n asteriscs.

   (b) `String dibuixaLiniaDiscontinua(int n)`, de manera que dibuixi per pantalla una línia de n asterics amb espais a les posicions senars.

   (c) `void dibuixaTriangle(int n)`, de manera que dibuixi per pantalla un triangle de n pisos. Per exemple, `dibuixaTriangle(3)` retorna:
   *
   **
   ***

   (d) `void dibuixaTriangleInvertit(int n)`, de manera que dibuixi per pantalla un triangle invertit de n pisos. Per exemple, `dibuixaTriangleInvertit(4)` retorna:

```
****
***
**
*
```

(e) `void dibuixaNFigures(int mètode, int n)`. Aquest mètode mostra per pantalla 'n' vegades una figura. El primer paràmetre indica quina figura mostrar: 1 indica la figura del mètode `dibuixaTriangle()`, 2 indica la figura del mètode `dibuixaTriangleInvertit()`. Per exemple, si es crida a la funció amb `dibuixaNFigures(2,3)`, cridarà al mètode `dibuixaTriangleInvertit()` 3 vegades.

(f) Els mètodes `dibuixaLinia()`, `dibujaLiniaDiscontinua()`, `dibuixaTriangle()`, `dibuixaTriangl` i `dibuixaNFigures()` són mètodes de classe o mètodes d'objecte? Perquè? Contesteu aquesta pregunta cap al final del fitxer .java entre comentaris /* */.

(`Exercici6.java`)

```
1         public class Exercici6 {
2             public static void main(String[] args) {
3
4                 System.out.println("UNA␣LINIA");
5                 System.out.println(dibuixaLinia(10));
6                 System.out.println("");
7
8                 System.out.println("UNA␣LINIA␣DISCONTINUA␣EN␣NOMBRES␣PARELLS");
9                 System.out.println(dibujaLiniaDiscontinua(10));
10                System.out.println("");
11
12                System.out.println("UN␣TRIANGLE");
13                dibuixaTriangle(4);
14                System.out.println("");
15
16                System.out.println("UN␣TRIANGLE␣INVERTIT");
17                dibuixaTriangleInvertit(4);
18                System.out.println("");
19
20                System.out.println("DIBUIXA␣FIGURES");
21                dibuixaNFigures(2, 4);
22            }
23            static String dibuixaLinia(int n){
24                //Introduiu solucio aqui
25            }
26
27            static String dibuixaLiniaDiscontinua(int n){
28                //Introduiu solucio aqui
29            }
30
31            static void dibuixaTriangle(int alt){
32                //Introduiu solucio aqui
33            }
34
35            static void dibuixaTriangleInvertit(int alt){
36                //Introduiu solucio aqui
37            }
38
39            static void dibuixaNFigures(int metode, int cops){
40                //Introduiu solucio aqui
41            }
42        }
```

7. Hem perdut el comandament de la televisió i com no volem aixecar-nos a buscar-lo decidim que programarem amb el nostre portàtil una classe **TV** i uns mètodes per encendre i apagar la

televisió `onOff()`, canviar de canal `canviarCanal()`, pujar el volum `pujarVolum()` i abaixar el volum `baixarVolum()`.
Suposeu que encara que crideu el mètode `pujarVolum()` per pujar el volum, si la televisió està apagada no es farà res. En canvi, si canviem el canal encara que estigui apagada l'encendrem. Suposeu que a l'hora d'introduir canals no podeu introduir ni 0 ni cap negatiu, a més la memòria del vostre televisor no permet més de 50 canals (del 0 al 49), de manera que si posem canal 51 es posarà el canal 1. (*Pista:* Recordeu l'operador mòdul? %). Observeu el codi i realitzeu els punts que s'indiquen a continuació:

(a) Completeu els atributs i mètodes de la classe **TV**, afegiu un atribut que indiqui la marca de la televisió i canvieu els constructors que creieu necessaris.

(b) Us ha encantat la idea de tenir el control remot de la televisió al portàtil i decidiu que voleu també afegir al programa la televisió de l'habitació del vostre germà petit, Pau, per tal de poder-li apagar la televisió quan faci els deures. Comproveu si la televisió d'en Pau és encesa.

(c) Després de tantes hores de programar, caieu adormits sobre el teclat. Per prevenir que no caigueu cridant al mètode de pujar el volum infinitament poseu la protecció de què no es pugui pujar el volum més de 100dB ni menys de -100dB.

(d) Escriviu el programa i feu els tests que creieu necessaris per tal de complir les necessitats de l'enunciat.

(e) En la classe **TV**, quins atributs són atributs de classe i quins són atributs d'objecte? Raoneu la vostra resposta. Contesteu aquesta pregunta cap al final del fitxer .java entre comentaris /* */.

(`Exercici7.java`)(`TV.java`)

```
1  /*
2  * Author:
3  * Data:
4  *
5  */
6
7  /* Alguns exemples de tests metode onOff() (afegiu mes)
8
9  * *             Entrada                          | Sortida Esperada
10 * ------------------------------------------------
11 * La tele esta apagada i cridem onOff()          |       La tele esta encesa
12 * ------------------------------------------------------
13 * La tele esta apagada i cridem pujaVolum()     | La tele esta apagada i no s'ha pujat
     el volum
14 *-------------------------------------------------------------------
15 */
16   //Fer tests per a tots el metodes
17
18   public class Exercici7 {
19     public static void main(String[] args) {
20       TV salaTV = new TV("Sala␣d'estar");
21
22       //enceneu la tele aqui
23
24
25       System.out.println("La␣" + salaTV.nom + "esta␣" + "encesa?␣"+ salaTV.encesa);
26       //canvieu de canal al canal9 i
27       //printeu en quin canal esteu
28
29       //pujeu el volum 20 dB i
30       //printeu el volum actual
```

```
31
32        //apagar de nou la televisio
33
34        //intenteu pujar el volum amb la tele apagada
35
36        //canvieu de canal amb la tele apagada
37
38        // canvieu de canal al canal 102
39        //printeu l'estat de la tele, teniu en compte els detalls a donar segons si
              esta o no encesa.
40
41      }
42    }
```

```
1     class TV {
2           String nom;
3       // Defineix els atributs de canal, volum i encesa
4
5       // Constructor per defecte
6       TV(){
7           nom = "␣";
8           canal = 1; // la tele esta al canal 1
9           volum = 1; // quan es crea la tele te poquet volum, 1dB
10          encesa = false; // quan es crea la tele esta apagada
11      }
12          // Constructor amb parametre de nom de la tele
13      TV(String nomNou) {
14          nom = nomNou;
15          canal = 1; // la tele esta al canal 1
16          volum = 1; // quan es crea la tele te poquet volum, 1dB
17          encesa = false; // quan es crea la tele esta apagada
18          }
19       // Constructor amb 4 parametres
20       TV(String nouNom, int canalNou, int volumNou, boolean encesaInici) {
21          nom = nouNom;
22          canal =  canalNou;
23          volum = volumNou; // es crea la tele amb els parametres per defecte que
                defineix l'usuari al constructor
24          encesa = encesaInici;
25        }
26       // Metode per apagar la tele si esta encesa i encendre-la si esta apagada i
             mostra missatge de l'accio que s'ha realitzat (Pista: ! )
27       void onOff(){
28         //Introduiu codi aqui
29       }
30
31       //  Metode pujar el volum a la tele 1dB
32      void pujaVolum(){
33        //Introduiu codi aqui
34      }
35
36       // Metode per baixar el volum 1dB
37      void baixaVolum(){
38                  //Introduiu codi aqui
39      }
40
41       /// Metode per canviar el canal rep el canal de la tele a canviar. Si es 50 o
             superior canvia al modul del numero.
42       void canviaCanal(int canalNou){
43          // Introdueix codi aqui
44       }
45    } // fi de la classe TV
```

8. Escriviu un programa per a la simulació d'un taximètre. Suposeu que la baixada de bandera té un cost (constant) BANDERA = €5.0 i que hi ha 3 zones amb tarifa diferent:

   - ZONA A = €60 a la hora

- ZONA B = €90 a la hora
- ZONA C = €2 al kilòmetre

A l'usuari se li demanarà la zona, i la quantitat de *minuts* (ZONA A i ZONA B) o *kilòmetres* (ZONA C); el programa imprimirà l'import de la factura. (`Exercici8.java`)(`Taximetre.java`)

```
/*
   * Author:
   * Data:
   *
   */

   /* Tests metode calculaPreu()
  *
  *
  */

  public class Exercici8 {
    public static void main(String[] args) {
     Taximetre taxisBarna;
     taxisBarna = new Taximetre(60, 90, 2);
     // Demaneu per consola la introduccio de la zona (en majuscules o minuscules) i
         els minuts o km, demaneu pel calcul de l'import a pagar i mostreu-lo

    }
  }
```

```
  class Taximetre {
    // atributs
    double zonaA;
    double zonaB;
    double zonaC;

    // Constructor amb parametres
    Taximetre(double preuZonaA, double preuZonaB, double preuZonaC) {
      zonaA = preuZonaA;
      zonaB = preuZonaB;
      zonaC = preuZonaC;
        }
    //  Metode per obtenir el preu a pagar segons la zona, rep la zona i els minuts.
    double calcularPreuAB(char zona, int minuts) {
      //Introduiu solucio aqui
    }
    double calcularPreuC(double km){
        //Introduiu solucio aqui
    }
  }
```

9. Feu una versió amb Enum de l'exercici anterior.

10. Esteu fent un pastís d'aniversari per un amic i ja l'esteu introduint al forn. A la recepta indica que l'heu de deixar 15 minuts, després obrir el forn i veure si introduint un ganivet surt tacat. Si surt tacat, l'heu de deixar 10 minuts més i tornar a fer la comprovació i si no surt tacat, heu d'apagar el forn i deixar-lo reposar 3 minuts més.
    La classe anomenada **Cronometre** té:

    - Els atributs tempsInicial i tempsFinal.
    - Un constructor sense paràmetres que inicialitza el tempsInicial amb el moment actual, useu la utilitat del paquet **java.lang.Object** el mètode `currentTimeMillis()` de la classe **System**. https://docs.oracle.com/javase/7/docs/api/java/lang/System.html

- Un mètode anomenat `updateTime()` que posa el tempsFinal amb l'hora actual.
- Un mètode anomenat `tempsTranscorregut()` que ens mostra quant de temps ha passat des de que s'ha creat l'objecte fins que es crida `updateTime()`.
- Afegiu un mètode anomenat `msToMin()` conversor de mil · lisegons a minuts, afegiu els paràmetres que cregueu que rep i el tipus que retorna.

Feu que el mètode principal **main()** esperi els 15 min de cocció del pastís. Una vegada transcorregut aquest temps, pregunteu a l'usuari si el ganivet surt tacat o no [reposta: SI o NO]. Si la resposta es SI surt tacat, el programa ha de tornar a comptar 10 minuts més i després tornar a preguntar a l'usuari com surt el ganivet. Si la resposta es NO surt tacat, el programa espera 3 minuts més de repòs i, una vegada transcorreguts aquests minuts, anunciar que ja el podeu treure del forn. Una possible sortida podria ser:

```
El pastis es al forn cronometre iniciat 15min.
Ja han passat els 15min! Comproveu la coccio del pastis ara.
Ha sortit moll el ganivet?[SI/NO]
NO
Perfecte!, deixeu-lo al forn reposar 5min, cronometre iniciat 5min.
Pi, pi, pi, pi ! Ja podeu treure el pastis de forn i deixar-lo refredar sobre una reixe
```

Altre sortida possible:

```
El pastis es al forn cronometre iniciat 15min.
Ja han passat els 15min! Comproveu la coccio del pastis ara.
Ha sortit moll el ganivet?[SI/NO]
SI
El deixarem 10min més, cronometre iniciat 10min.
Ja han passat els 10min! Comproveu la coccio del pastis ara.
Ha sortit moll el ganivet?[SI/NO]
NO
Pi, pi, pi, pi ! Ja podeu treure el pastis de forn i deixar-lo refredar sobre una reixe
```

(Exercici10.java)(Cronometre.java)

```java
public class Exercici10 {
    public static void main(String[] args) {
      Cronometre cronoCake = new Cronometre();

                // aneu cridant updateTime() i comprovant el tempsTranscorregut()
                    mentre que no haguin passat 15min
                // quan haguin passat els 15 min pregunteu a l'usuari per l'estat del
                    pastis i inicieu el comptador altre cop depenent de la resposta.
    }
```

```java
class Cronometre {
    //atributs
    long tempsInicial, tempsFinal;
    //no oblidis consultar la documentacio per tal de saber quin tipus retorna el
        metode currentTimeMillis() de la classe System de Java (enllac a l'enunciat)

    // Constructor de la classe Cronometre, tan bon punt es crea el cronometre el
        pastis ja ha de ser al forn (guarda el temps actual en milisegons)
    Cronometre() {
        tempsInicial = System.currentTimeMillis(); //guarda l'hora del sistema en
            milisegons
```

```
    }

    // Metode que guarda a la variable tempsFinal el temps actual.
    public void updateTime() {
       // Escriviu codi aqui
    }

    // Metode que retorna el temps transcorregut en milisegons
    long tempsTranscorregut() {
     // Calcula tempsTranscorregut entre tempsInicial i tempsFinal i retorna'l
    }
}
```

11. Definiu la classe **Minim** i dos mètodes:

    (a) El mètode `minim3()` que calcula el mínim de tres nombres naturals, fent servir com a màxim dues instruccions `if`.

    (b) El mètode `minimUnlimited()` que calcula el mínim de tots els nombres naturals que introdueix l'usuari fins que tecleja -1 per sortir.

    (`Minim.java`)

```
/*
   * Author:
   * Data:
   *
   */

 /* Test metode minim3()
  *
  *
  */

  public class Minim {
    public static void main(String[] args) {


    }

    static int minim3( ){
        // Definiu i inicialitzeu un scanner i aneu preguntant i calculant el minim
            dels nombres i el retorneu
    }
    static int minimUnlimited(){
        // Completeu el codi
    }
  }
```

12. Definiu la classe **Ordre** que tingui el mètode `ordenar()` que demana tres nombres i els escriu ordenats de major a menor, fent servir com a màxim tres instruccions `if`. Segueix la mateixa estructura de l'exercici anterior. (`Ordre.java`)

13. La classe **Alumne** emmagatzema el nom i les notes dels tres exàmens que ha realitzat un alumne, els punts extra que ha obtingut, i la nota final que li ha quedat. Aquesta classe també emmagatzema el màxim de punts extra que un alumne pot obtenir. La nota final d'un alumne és la mitjana de les notes dels tres examens més els punts extra. Totes les notes estan a l'interval $[0, 10]$, per tant si la nota final, una vegada sumats els punts extra fossi major a 10, aquesta ha de quedar amb valor 10.

    (a) Implementeu el mètode `calcularNotaFinal()` de la classe **Alumne**. Aquest mètode retorna la nota final de l'alumne.

(b) Implementeu el mètode `veureNotaFinalFormatText()` de la classe **Alumne**. Aquest mètode retorna una cadena amb l'equivalent en text de la nota numèrica; només en el cas que la nota sigui d'Insuficient, retornarà dues coses: un String amb el text de la nota i la nota numèrica.

Nota	**Equivalent en text**
0 a 5	Insuficient
5 a 6	Suficient
6 a 7	Notable Baix
7 a 8	Notable Alt
9 a 10	Excel · lent

(c) Escriviu la taula de tests dels dos mètodes anteriors. (Veure codi avall).

(d) Creieu un objecte de la classe **Alumne** i sol · liciteu a l'usuari el nom i les notes dels tres exàmens de l'alumne. Suposa que l'usuari introdueix les notes a l'interval $[0, 10]$.

(e) En la classe **Alumne**, quins són atributs de classe i quins són atributs d'objecte? Raoneu la vostra resposta. Contesteu aquesta pregunta cap al final del fitxer .java entre comentaris /* */.

(f) Els mètodes `veureNotaFinalFormatText()` i `modificarMaxPuntsExtra()`, a quina classe pertanyen? són mètodes de classe o mètodes d'objecte? Per què? Contesteu aquesta pregunta cap al final del fitxer .java entre comentaris /* */.

(`Exercici13.java`)(`Alumne.java`)

```java
/*
 * Author:
 * Data:
 */


public class Exercici13 {

  public static void main(String[] args) {
      Alumne meuAlumne;
      // Creeu un alumne, demaneu les dades de l'alumne a l'usuari, i crideu el
          metode calcularNotaFinal() i veureNotaFinalFormatText()

  }
}
```

```java
class Alumne{
   static double maxPuntsExtra = 1.5;
   String nom;
   double notaExamen1;
   double notaExamen2;
   double notaExamen3;
   double puntsExtra;
   double notaFinal; // La nota final es calcula fent la mitjana de les notes dels
       examens + puntsExtra

   Alumne (String nom){
     this.nom = nom;
     // La resta de paràmetres s'inicien per defecte a 0.0
   }

   //Constructor amb parametres
   Alumne(String nom, double notaExamen1, double notaExamen2, double notaExamen3,
       double puntsExtra) {
       this.nom = nom;
```

```
18          this.notaExamen1 = notaExamen1;
19          this.notaExamen2 = notaExamen2;
20          this.notaExamen3 = notaExamen3;
21          this.puntsExtra = puntsExtra;
22      }
23
24      static void modificarMaxPuntsExtra(double max){
25        maxPuntsExtra = max;
26      }
27      /* Tests del metode calcularNotaFinal()
28    *
29    *
30    *
31    *
32    */
33      void calcularNotaFinal() {
34          // Escriu el codi que falta
35      }
36
37      /* Tests del metode veureNotaFinalFormatText()
38    *
39    *
40    *
41    *
42    */
43       // Mostar la nota en el seu equivalent en text; en el cas que la nota sigui
             Insuficient mostrar el text i la nota numerica
44      String veureNotaFinalFormatText(){
45        // Escriu el codi que falta
46
47      {
48    }
```

14. Definiu la classe **DiaSetmana** que contingui:

    - El mètode principal `main()`: on es demanarà a l'usuari un número de dia de la setmana de 1 a 7.
    - El mètode static `getNomDia()` que rebrà el número del dia de la setmana (introduït per l'usuari) com a paràmetre i retornarà l'String corresponent al nom del dia. En el mètode `main` es farà la crida a aquest mètode. Com heu implementat aquest mètode, com a mètode d'objecte o mètode de classe? Raoneu la teva resposta. Contesteu aquesta pregunta cap al final del fitxer .java entre comentaris /* */..
      (`DiaSetmana.java`)

15. Es tracta de simular n $(0 < n \leq 5)$ tirades del JOC DEL KIRIKI. En aquest joc juguen dos **Jugador**s, cadascun tira dos daus (cada dau pren valor de l'1 a 6) i obté una puntuació. El jugador que guanya la tirada és el que aconsegueix la puntuació més alta.

    Puntuacions: La puntuació màxima és la corresponent a un kiriki, és a dir, quan surt un 1 i un 2; li donarem el valor de 50 punts. Després venen les parelles (les que valen més són les de 6, i les que menys les de 1); el seu valor serà la seva suma més 20 punts. Finalment, si no ha sortit cap dels casos anteriors, s'han de sumar els valors dels dos daus.

    Un cop feta la tirada, el programa mostrarà per pantalla la jugada obtinguda per cada jugador, analitzarà la jugada de cada jugador i imprimirà quin jugador és el guanyador, i si hi ha algun kiriki ho indicarà.

    Exemple ($n = 2$):

```
primer jugador: 1 2
punts: 50
segon jugador: 3 3
punts: 26
KIRIKI del primer jugador!

primer jugador: 5 5
punts: 30
segon jugador: 5 5
punts: 30
Heu empatat!

primer jugador: punts: 80
segon jugador: punts: 56
Guanya el primer jugador
```

Per simular una jugada, generareu nombres pseudoaleatoris. Utilitzeu el mètode `nextInt` de la classe `Random` https://docs.oracle.com/javase/7/docs/api/java/util/Random.html : si li passem l'argument n, retorna un nombre enter m, amb $0 \leq m < n$.

Per exemple per generar un nombre que sigui 0, 1, o 2:

```
int n;
Random aleatori = new Random();
n = aleatori.nextInt(3);
System.out.println("El␣nombre␣aleatori␣es:␣" + n);
```

Codi a emplenar:
(`Kiriki.java`)(`Jugador.java`)

```
  /*
   * Author:
   * Data:
   */

  public class Kiriki {
        Jugador jugador1 = new Jugador();
        Jugador jugador2 =  new Jugador();
    public static void main(String[] args) {
                // feu que n vegades ambdos jugadors facin les seves tirades i
                    decidiu els guanyadors
        // Decidiu qui finalment guanya
    }

  }
```

```
  class Jugador{
        String nom;
    Random aleatori;
        // defineix atribut de puntuacio aqui

    //Constructor amb parametre
    Jugador(String nom){
        this.nom = nom;
        //Introduiu codi aqui
    }

    // Metode que obte dos numeros pseudoaleatoris de l'1 al 6 (Random) i assigna la
        puntuacio corresponent al jugador. Retorna els punts de la jugada
    int tirarDaus(){
```

```
14            // Codi aqui
15        }
16    }
```