

## Exercicis

### Exercici 1

Crea un mètode amb el nom **saludar** que en cridar-lo mostre un missatge de salutació per consola. Des de main(), crida al mètode que has creat.

### Exercici 2

Crea dos mètodes amb els noms **triangle5** i **triangle7** que en cridar-los dibuixen, utilitzant asteríscs, un triangle d'altura 5 o d'altura 7 (*segons corresponga*). Des de main(), crida als dos mètodes que has creat. Exemple d'eixida:

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

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

### Exercici 3

Crea un mètode amb el nom **quadrat** que dibuixi, utilitzant asteríscs, un quadrat de costat 5. Des de main(), crida al mètode que has creat. Exemple d'eixida:

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

### Exercici 4

Crea un mètode amb el nom **dibuixaTriangle** que sol·licitarà a l'usuari que introduïska un valor sencer (*altura del triangle*) entre 1 i 15 i dibuixarà el triangle corresponent mitjançant asteríscs. Des de main(), crida al mètode.

### Exercici 5

Crea un mètode amb el nom **llegirEnter**, la funció del qual és sol·licitar a l'usuari que introduïska un valor sencer i retorne el valor que ha introduït l'usuari. Si l'usuari no introdueix un enter vàlid, ha de tornar a sol·licitar la introducció fins que l'entrada siga un enter correcte.

Des de main(), utilitza el mètode per a sol·licitar a l'usuari dos paràmetres sencers:

**espai** i **temps**. Una vegada llegits mostrarà la velocitat. **Nota:**

$\text{velocitat} = \text{espai} / \text{temps}$ .

### Exercici 6

El programa anterior mostra el mateix missatge per a llegir l'espai que per a llegir el temps. Modifica el procediment anterior (**llegirEnter**) per a passar-li un paràmetre, que serà el missatge que mostrarà a l'usuari indicant el valor que es desitja introduir.

Exemple:

```
espai= llegirEnter ("Introdueix l'espai: ");  
temps= llegirEnter ("Introdueix el temps: ");
```

### Exercici 7

Crea un mètode amb el nom **potencia** al qual li passaràs dos paràmetres de tipus sencer (*base i exponent*). El mètode retornarà un enter llarg consistent en la base elevada a l'exponent, es a dir, la base multiplicada per ella mateixa tantes vegades com indica l'exponent. No utilitzes el mètode **pow** de la classe **Math**. Des de **main()**, crida al mètode creat per a provar el seu funcionament.

### Exercici 8

Crea un mètode amb el nom **mcm** al qual li passaràs dos paràmetres de tipus enter i retornarà un enter corresponent al mínim comú múltiple dels dos números. Prova el seu funcionament amb crides al mètode des de **main()**.

### Exercici 9

Crea un mètode amb el nom **factorial** al qual li passaràs un valor de tipus sencer i retornarà el seu factorial. Prova el seu funcionament realitzant crides al mètode des de **main()**.

Factorial (4) = 4 \* 3 \* 2 \* 1

Factorial(7) = 7 \* 6 \* 5 \* 4 \* 3 \* 2 \* 1

Factorial(n) = n \* (n - 1) \* (n - 2) \* ... \* 3 \* 2 \* 1

### Exercici 10

Crea un mètode amb el nom **mcd** al qual li passaràs 3 paràmetres i retornarà el mínim comú divisor dels tres. Prova el mètode realitzant crides des de **main()**.

mcd(6, 15, 21) tornarà 3

### Exercici 11

Crea un mètode amb el nom **aleatoriEntre** al qual li passes dos paràmetres de tipus sencer (*minim*, *maxim*) i retorne un valor de tipus sencer entre [minim , maxim]. Prova el seu funcionament realitzant crides des de main().

### Exercici 12

Crea un mètode amb el nom **tablaDel** al qual li passes un valor de tipus sencer i mostre per pantalla la seua taula de multiplicar. Exemple:

Suposem que se li ha passat el 7

```
7 x 0 = 0
7 x 1 = 7
7 x 2 = 14
7 x 3 = 21
7 x 4 = 28
7 x 5 = 35
7 x 6 = 42
7 x 7 = 49
7 x 8 = 56
7 x 9 = 63
7 x 10 = 70
```

### Exercici 13

Crea un mètode amb el nom **ascensor** al qual li passes dos paràmetres: pis actual i pis de destí. El mètode et mostrarà tots els pisos pels quals va passant fins a arribar al destí. En el cas que actual i destí siga el mateix pis, no farà res. Exemple:

Del 8 al 4	Del 3 al 6
Tancant portes	Tancant portes
Baixant.	Pujant
Baixant: pis 7	Pujant: pis 4
Baixant: pis 6	Pujant: pis 5
Baixant: pis 5	Destí: pis 6
Destí: pis 4	Obrint portes
Obrint portes	

#### Exercici 14

Crea un mètode amb el nom **mostraArray** que rep com a paràmetre un array de String i mostra el seu contingut en pantalla de manera que cada String aparega en una línia diferent.

```
String saludo = {"Avui és dilluns", "Demà serà dimarts", "La setmana és llarga"};
muestraArray(saludo);
```

La eixida serà:

```
Avui és dilluns
Demà serà dimarts
La setmana és llarga
```

#### Exercici 15

Crea un mètode amb el nom **sumaArray** que reba com a paràmetre un array d'elements de tipus sencer i retorne la suma de tots els elements del array.

#### Exercici 16

Crea un mètode amb el nom **estadistiquesArray** que reba com a paràmetre un array d'enters i retorne:

- Suma dels valors del array.
- Valor més alt dels valors del array.
- Valor més xicotet dels valors del array.
- Quantitat d'elements imparells en el array.
- Quantitat d'elements del array que són múltiples de 7

#### Exercici 17

Fes un programa que en executar-lo des de la consola li passes una adreça IP segons el protocol IPv4, i et diga si és de tipus A, B, C, D o E, i si és pública o privada.

#### Exercici 18

Crea un mètode amb el nom **aleatorisN**, que li passes un valor sencer **N** i retorna un array amb N números aleatoris entre 0 i 1.

#### Exercici 19

Crea un mètode amb el nom **aleatorisNMinMax** que rebrà 3 paràmetres: **num**, **minim**, **maxim**.

El mètode retornarà un array d'enters amb “**num**” elements, els valors dels quals estaran compresos entre “**minim**” i “**maxim**”.

#### Exercici 20

Crea un mètode amb el nom **mostraArray** que rep com a paràmetre un array d'enters i mostra el seu contingut en consola, ficant varios valors a la mateixa línia. El nombre màxim de valors mostrats en una línia serà de 15.

#### Exercici 21

Crea un mètode amb el nom **converteixIntString** que rep un array d'enters i retorna un String format pels enters separats per un espai. El String no comença ni acaba amb espais.

Rep: {10, 21, 15}

Torna: “10 21 15”

#### Exercici 22

Crea un mètode amb el nom **converteixStringInt** que rep un String com a paràmetre, format per nombres enters separats per un espai, i retorna un array d'enters format pels enters que hi ha en el String.

Rep: “10 10 11 15 18 03 06”

Torna: {10, 10, 11, 15, 18, 03, 06}

#### Exercici 23

Crea un mètode amb el nom **paraulaTrobada** que rebrà dos String's: **frase** i **paraula**. El primer paràmetre contindrà una frase i el segon una paraula (*sense espais ni cap altre signe*). El mètode retornarà true si la paraula es troba dins de la frase, en cas contrari retornarà false.

Per a aquest exercici **no pots utilitzar** el mètode **.contains()**, ni **.contentEquals()**.

#### Exercici 24

Crea un mètode amb el nom **paraulaEnArray** que reba com a paràmetres un array de String's i un String amb una paraula (*sense espais ni altres caràcters que no formen part de la paraula*).

El mètode retornarà un array d'enters amb els índexs del array que contenen la paraula rebuda com a paràmetres. Si no troba la paraula en cap String del array, retornarà **null**.

Exemple:

**Entrada:** `array`: {"Una frase bonica", "Tinc una casa a la muntanya", "La vida és bella",  
"Bon dia, princesa", "La casa dels teus somnis"}  
`paraula`: casa

**Eixida:** [1, 4]

Per a l'exercici has d'utilitzar el mètode **comptaRepeticions** que vas crear en l'exercici 23. També pots utilitzar el mètode **copyOf** de la classe **Arrays**.

### Exercici 25

Crea un mètode amb el nom **comptaRepeticions** que reba com a paràmetres un array d'enters, **llista**, i un valor sencer, **element**. El mètode retornarà un valor sencer que representa el nombre de vegades que apareix el valor **element** en el array **llista**.

### Exercici 26

Crea un mètode amb el nom **uneixCadenes** que reba com a entrada un array de String's i retorne una cadena amb la unió de totes elles. En realitzar la unió, entre cadena i cadena afeg un espai de separació.

Entrada: {"Una frase", "molt especial. ", "M'agrada programar"}

Eixida: "Una frase molt especial. M'agrada programar"

## Sobrecàrrega de mètodes

### Exercici 27

A l'exercici 5 i 6 has creat un mètode amb el nom **llegirEnter**. Sobrecarrega el mètode perquè el pugues cridar amb paràmetre i sense paràmetre. Exemple:

```
espai = llegirEnter("Introdueix l'espai recorregut: "); // crida amb paràmetre
temps = llegirEnter();                               // crida sense paràmetre
```

### Exercici 28

En l'exercici 10 has creat un mètode per a obtenir el **mcd** de 3 números. Sobrecarrega el mètode perquè pugues cridar al **mcd** de 2 números enters i d'un array d'enters.

Sobrecarrega també perquè pugues cridar-ho amb 2 números de tipus long i un array de tipus long.

### Exercici 29

En l'exercici 14 has creat el mètode **mostraArray** que mostra un array de tipus String. Sobrecarrega el mètode perquè també mostre arrays de tipus **boolean**, **char**, **int** i **double**.

### Exercici 30

En l'exercici 15 has creat el mètode **sumaArray** que retorna la suma dels elements d'un array d'enters. Sobrecarrega el mètode perquè també retorne la suma dels elements d'un array de double's.

## Atributos y Constructores

**Nota:** els següents exercicis formen part d'un mateix projecte. Crea un projecte i ves modificant-lo exercici a exercici.

### Exercici 31

Crea una classe amb el nom **Punt** amb dos atributs: “**x**” e “**y**” de tipus sencer. Crea els següents constructors:

- **Punt()**: constructor per defecte que inicialitzi x e y a 0.
- **Punt(int a, int b)**: constructor que rep dos paràmetres de tipus sencer i li assigna a “X” el valor “a” i a “Y” valor “b”.

### Exercici 32

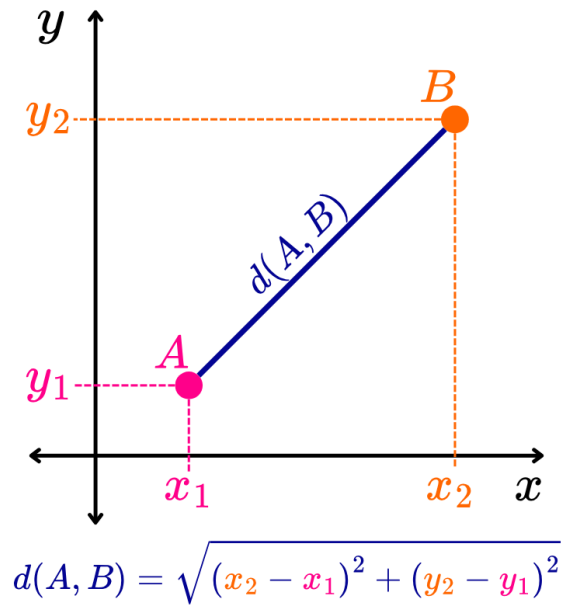
Modifica la classe Punt per a afegir dos mètodes (getters):

- **getX()**: retorna el valor de “x”
- **getY()**: retorna el valor de “y”.

### Exercici 33

Modifica la classe **Punt** per a afegir-li un mètode amb el nom **distancia**, que rep com a entrada un objecte de la classe **Punt** i retorna un valor real que representa la distància que existeix entre l'objecte punt i el punt passat com a paràmetre.

La següent imatge explica com calcular la distància entre dos punts.



### Exercici 34

El servei d'urgències té monitoritzat en tot moment les ambulàncies que estan disponibles en la comunitat valenciana. Aquesta informació es troba en l'arxiu **ambulancias.txt**, on cada línia conté la coordenada d'una ambulància lliure (*primer número representa el valor de "X" i el segon número representa el valor de "Y". Els números estan separats per una coma*).

Quan ocorre un incident, el servei d'urgències rep les coordenades del lloc al qual ha d'enviar l'ambulància i ha d'avisar a la que es trobe més pròxima.

#### Exercici 34 – A)

Crea un mètode (**llegirAmbulancias**) en la aplicació principal, que lliça els valors de l'arxiu **ambulancias.txt** i els emmagatzeme a un array d'objectes de tipus **Punt**. A l'arxiu hi ha 199 ambulancias. El mètode rebrà com entrada la ruta de l'arxiu.

```
Punt ambu[] = llegirAmbulancias("src/ambulancias.txt");
```

#### Exercici 34-B)

Si tenim un incident en la coordenada (-4946, 680), quina és la coordenada de l'ambulància més pròxima?

Solució: -4584, 520

### Exercici 35



Continuant amb l'exercici anterior, modifica el programa per a obtindre el nombre d'ambulàncies que estan a una distància inferior o igual a 1000 de la coordenada de l'incident?

Solució: 4

## Encapsulament

**Nota:** continuem treballant al mateix projecte.

### Exercici 36

Modifica la classe Punt per a que els atributs segueixin privats.

Afegeix els mètodes `setX(int valor)` i `setY(int valor)` (mètodes *setters*) que et permetran modificar els valors de "X" i "Y".

### Exercici 37-A

Modifica la classe **Punt** per a afegir-li el mètode `setAleatori(minX, maxX, minY, maxY)`, la funció del qual serà modificar els valors de **X** i **Y** per a assignar-los un valor aleatori entre min i max.

### Exercici 37-B

A la aplicació principal crea un array de 500 elements d'objectes de tipus **Punt**. Inicia els objectes del array amb valors aleatoris compresos entre 1 i 60 per a "x" i entre 1 i 20 per a "y".

### Exercici 38

Crea un mètode en la classe **Punt** que rebi com a paràmetres un objecte de la classe punt i un enter que represente una distància.

```
public boolean estaCerca(Punt punt, int distància)
```

Si la distància entre l'objecte i el punt passat com a paràmetre és inferior o igual al valor de distància, retorna `true`, en cas contrari retorna `false`.

### Exercici 39

A la aplicació principal crea un mètode amb el nom **dibuixaLinia** que rebrà els següents paràmetres:

- **array:** un array d'objectes de tipus Punt,

- **posY**: valor de la coordenada Y a representar.
- **res**: caràcter que s'utilitzarà per a representart una posició en la qual no hi ha cap objecte de la classe Punt.
- **hiha**: caràcter que s'utilitzarà per a representar una posició en la qual hi ha un o més objectes de la classe punt.

La funció del mètode és passar-li un array de punts i un valor de Y i que retorne un String indicant on hi ha punts i on no. Exemple:

```
array = {(2,2), (2,8), (3,2), (12,5), (6,2), (12, 9), (15, 2), (20, 20), (1, 1), (17, 2)}
```

```
dibuixaLinia(array, 2, "-", "x")
```

Retornarà el següent String (*suposant valors entre 1 i 20 tant per a X com per a Y*):

“-XX—X-----X-X-”

Utilitza la funció per a mitjançant un bucle, mostrar els 20 valors possibles de Y (valors entre 1 i 20). Exemple:

#### Exercici 40

Sobrecarga el mètode **dibuixaLinia** de manera que reba els següents paràmetres:

- **array**: un array d'objectes de tipus Punt,
- **objecte**: objecte de tipus Punt.
- **posY**: enter. Valor del eix Y i que correspond a la línia a dibuixar.
- **distancia**: enter
- **res**: caràcter que s'utilitzarà per a representart una posició en la qual no hi ha cap objecte de la classe Punt.
- **prop**: caràcter que s'utilitzarà per a representart una posició en la qual hi ha un o més punts a una distancia inferior a “distancia” respecte a **objecte**.
- **llun**: caràcter que s'utilitzarà per a representar una posició en la qual hi ha un o més objectes a una distancia superior a “distancia” respecte a objecte.
- **focus**: caràcter que s'utilitzarà per a representart la posició del objecte, si està a la línia indicada en posY.
- **sup**: caràcter que s'utilitzarà per a representart la posició del objecte si coincideix amb la posició d'un altre punt.

Dibuixa el mapa tenint en compte que el objecte está al punt (45, 13) y els caracters pots utilitzar: '-', 'X', 'x', 'W', 'O'

### Recursivitat

#### Exercici 41

Crea un mètode recursiu que obtinga la suma dels nombres naturals des d'1 fins a N. S'ha de passar com a paràmetre el número N

$$\text{suma}(n) = n + \text{suma}(n - 1)$$

#### Exercici 42

Crea un mètode recursiu que imprimisca els dígit des d'1 fins a N. S'ha de passar com a paràmetre el número N.

#### **Exercici 43**

Crea un mètode recursiu que imprimisca els dígit des de N fins a 1. S'ha de passar com a paràmetre el número N

#### **Exercici 44**

Crea un mètode recursiu que obtinga la quantitat de dígit d'un número N. S'ha de passar com a paràmetre el número N.

$\text{numDigits}(458458) = 1 + \text{numDigits}(45845)$

#### **Exercici 45**

Crea un mètode recursiu que obtinga el factorial d'un número N. S'ha de passar com a paràmetre el número N.

$\text{factorial}(N) = N * \text{factorial}(N - 1)$

#### **Exercici 46**

Crea un mètode recursiu que calcule el número de fibonacci a partir d'un número passat com a paràmetre.

Sèrie de fibonacci: 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, ...

Comença per 1, 1 i cada element nou és la suma dels dos anteriors.

$\text{Fibonacci}(7) = \text{fibonacci}(6) + \text{fibonacci}(5)$

#### **Exercici 47**

Crea un mètode recursiu que obtinga el resultat d'eleva un número a un altre. Tots dos números s'han de passar com a paràmetres

$\text{potencia}(\text{base}, \text{exponent}) = \text{base} * \text{potencia}(\text{base}, \text{exponent} - 1)$

#### **Exercici 48**

Crea un mètode recursiu que donat un número, l'imprimisca invertit per pantalla.

#### **Exercici 49**

Crea un mètode recursiu que imprimisca per pantalla un Rectangle a partir dels valors de la base i l'altura.

**Exercici 50**

Crea un mètode recursiu que comprove si una paraula és un palíndrom.

**Exercici 51**

Crea un mètode recursiu que comprove si un número és binari. Un nombre binari està format únicament per zeros i uns.