



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA

Escola Tècnica Superior d'Enginyeria Informàtica  etsinf

# Entrada i eixida elemental

Introducció a la Informàtica i la Programació (IIP)

Curs 2018/19

Departament de Sistemes Informàtics i Computació



# Continguts

1. Introducció
2. Eixida per pantalla
  - `System.out.println` i `System.out.print`
  - Eixida formatejada amb `printf`
3. Entrada des de teclat
  - La classe `Scanner`

## Bibliografia:

“Empezar a programar usando Java (3ª edición)”

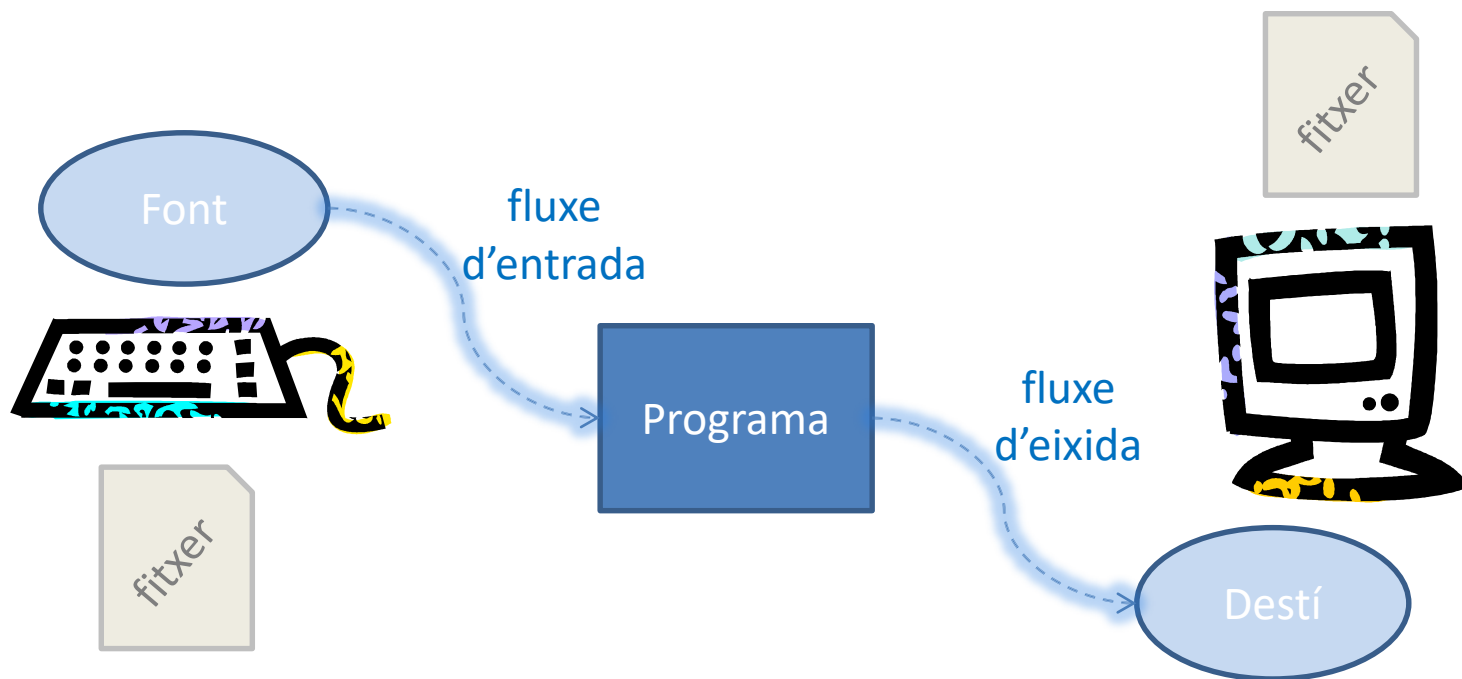
Professors d'IIP i PRG. Editorial UPV, 2016. **Capítol 6**



- **Descarrega** (del Tema 3 de PoliformaT) el fitxer ***exemples - Scanner.jar*** en ***W:\IIP\Tema 3***
- Des de l'opció **Projecte** de **BlueJ**, usa l'opció **Obre Projecte...** per tal d'obrir aquest com un projecte **BlueJ** i prepara't per usar-lo

# Introducció

- En Java l'entrada/eixida es realitza utilitzant fluxes (**streams**), que són seqüències d'informació que tenen una font (**fluxes d'entrada**) o un destí (**fluxes d'eixida**).



# Introducció

- Els fluxes predefinitos en Java, a la classe `System` del paquet `java.lang`, que permeten realitzar l'entrada/eixida estàndard són:

| Entrada estàndard      | Eixida estàndard        |
|------------------------|-------------------------|
| <code>System.in</code> | <code>System.out</code> |

- Hi ha un tercer fluxe predefinit `System.err`, idèntic a `System.out`, usat per mostrar missatges d'error.
- L'habitual és que aquests fluxes estàndard es troben associats per defecte al **teclat** (el d'entrada) i a la **pantalla** (el d'eixida).
- Es poden definir altres fluxes perquè un programa pugui llegir des de o escriure en fitxers. Una vegada definits, s'utilitzen de manera semblant als fluxes `System.in` i `System.out`.

# Eixida per pantalla:

## `System.out.println` i `System.out.print`

- La sintaxi de la instrucció que mostra una línia per pantalla és:

```
System.out.println(Elem_1 + Elem_2 + ... + Elem_n);
```

sent *Elem\_i* cada un dels elements a mostrar.

- Sense arguments mostra una línia en blanc.
- L'única diferència entre `System.out.println` i `System.out.print` és que amb `println`, la següent sortida a mostrar es mostra en una línia nova, mentre que amb `print`, la sortida es mostra en la mateixa línia.

```
double r = 5.5; String c = "roig";  
System.out.println("Cercle de radi " + r + ", color " + c);  
System.out.println();  
System.out.println(" i centre (" + 6 + ", " + 3 + ").");  
System.out.print("Cercle de radi " + r + ", color " + c);  
System.out.println(" i centre (" + 6 + ", " + 3 + ").");
```

Cercle de radi 5.5, color roig

i centre (6, 3).

Cercle de radi 5.5, color roig i centre (6, 3).

# Eixida per pantalla:

## Eixida formatejada amb `printf`

- La sintaxi de la instrucció que presenta l'eixida en un format específic és:

```
System.out.printf(String amb format, Llista d'arguments a formatejar);
```

- String amb format** és un `String` format per text normal i **especificadors de format** que indiquen com es van a formatejar els **valors de la llista d'arguments a formatejar**.
- Per exemple, es pot mostrar per pantalla el valor de `Math.PI` amb tres dígits decimals utilitzant:

```
System.out.printf("El valor de Math.PI és %.3f \n", Math.PI);
```

que mostraria:

```
El valor de Math.PI és 3,142
```

- Mentre que utilitzant:

```
System.out.println("El valor de Math.PI és " + Math.PI);
```

s'obtindria:

```
El valor de Math.PI és 3.141592653589793
```

# Eixida per pantalla:

## Eixida formatejada amb `printf`

- Un **especificador de format** comença amb el caràcter **%** i finalitza amb un caràcter que indica el tipus de conversió a realitzar, anomenat **indicador de conversió**. A més, pot contindre altres valors que controlen el disseny del valor a convertir.

**%** [índex\_argument\$] [flags] [amplària] [.precisió] indicador de conversió

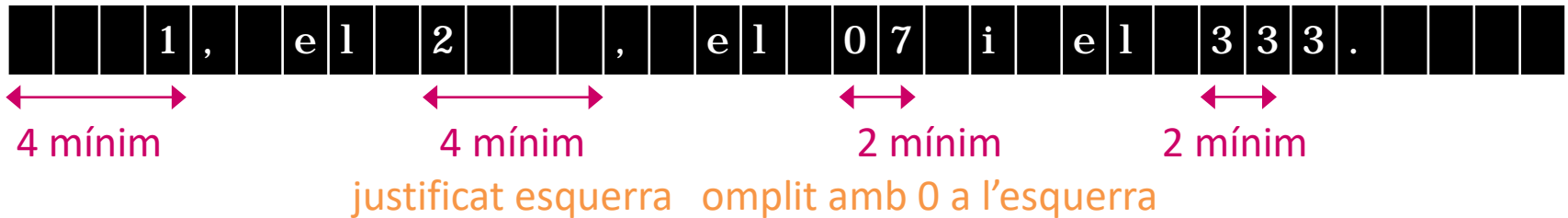


- Son **opcionals** i depenen de l'**indicador de conversió**:
- índex\_argument** és un enter que indica la posició de l'argument a la **llista d'arguments a formatjar**.
- flags** permeten, entre altres coses, la justificació a l'esquerra o a la dreta.
- amplària** indica el número mínim de caràcteres que apareixeran en l'eixida (útil per alinear columnes de dades).
- precisió** indica, en la conversió de valors reals, el número de xifres decimals que han d'apareixer.

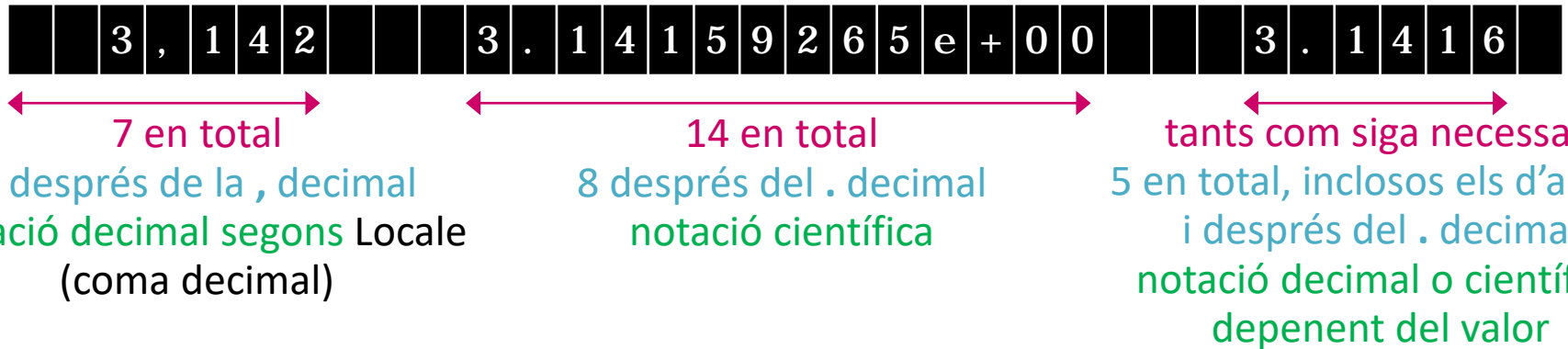
- És un caràcter que especifica el tipus del valor que es va a mostrar i el seu format.
- Pera `byte`, `short`, `int` i `long`:
  - d** --- format decimal
  - o** --- format octal
  - x, X** --- format hexadecimal
- Per a `float` i `double`:
  - e, E** --- notació científica informatizada (p.e., 3.142e+00)
  - f, F** --- notació decimal (p.e., 3,142)
  - g, G** --- notació científica general
- Per a conversions a `String`: **s, S**

# Eixida per pantalla: Eixida formatejada amb printf

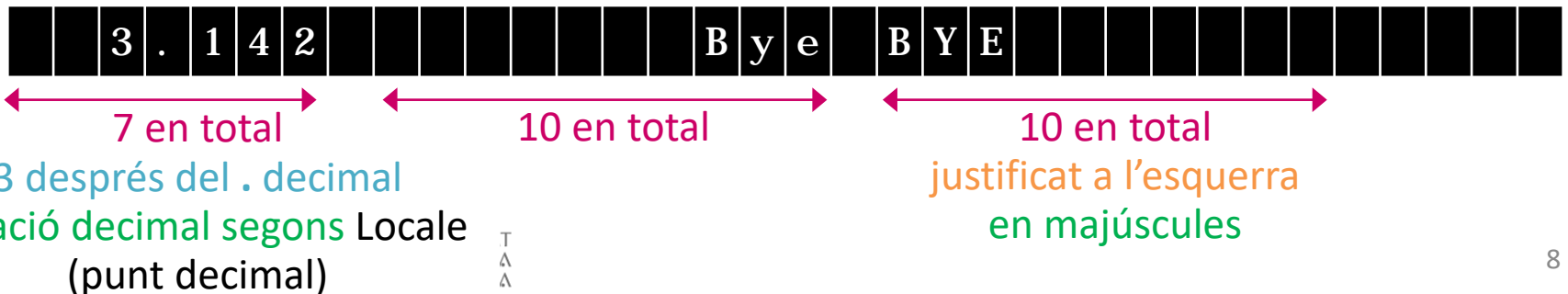
```
System.out.printf("%4d, el %-4d, el %02d i el %2d.\n", 1, 2, 7, 333);
```



```
System.out.printf("%7.3f %1$14.8e %1$1.5g", Math.PI);
```



```
System.out.printf(Locale.US, "%7.3f %10s %2$-10S", Math.PI, "Bye");
```





# Entrada des de teclat: la classe Scanner

- L'objecte `System.in` permet llegir dades introduïdes per l'usuari des de teclat però el seu ús no és tan senzill com el de l'objecte `System.out`.
- La classe `Scanner` forma part del paquet `java.util` i permet llegir valors introduïts pel teclat d'una forma molt còmoda per al programador.

- Al començament del codi afegir: `import java.util.Scanner;`

- Declaració: `Scanner identificador = new Scanner(System.in);`

- El mètode `useLocale` permet establir la configuració local d'un `Scanner`. La definició d'aquest mètode és: `public Locale useLocale(Locale l)`

```
import java.util.Scanner;
import java.util.Locale;
```



- Exemples d'ús: `Scanner teclat = new Scanner(System.in);`  
`teclat.useLocale(Locale.US);`

o directament,

```
Scanner teclat = new Scanner(System.in).useLocale(Locale.US);
```



# Entrada des de teclat: la classe Scanner

Referència: <http://docs.oracle.com/javase/8/docs/api/java/util/Scanner.html>

- Alguns mètodes de la classe Scanner:

## Definició - capçalera

```
public String next()
```

```
public String nextLine()
```

```
public byte nextByte()
```

```
public short nextShort()
```

```
public int nextInt()
```

```
public long nextLong()
```

```
public float nextFloat()
```

```
public double nextDouble()
```

```
public boolean nextBoolean()
```

## Definició - capçalera

```
public boolean hasNext()
```

```
public boolean hasNextLine()
```

```
public boolean hasNextByte()
```

```
public boolean hasNextShort()
```

```
public boolean hasNextInt()
```

```
public boolean hasNextLong()
```

```
public boolean hasNextFloat()
```

```
public boolean hasNextDouble()
```

```
public boolean hasNextBoolean()
```

# Entrada des de teclat: la classe Scanner



BlueJ:exemples - Scanner

```
1 import java.util.Scanner;
2 /**
3  * Classe TestScanner: classe programa per provar
4  * la lectura de dades des de teclat amb Scanner
5  * @author IIP
6  * @version Curs 2015/16
7  */
8 public class TestScanner {
9     public static void main(String[] args) {
10         Scanner teclat = new Scanner(System.in);
11         System.out.println("Introdueix el teu nom");
12         String nom = teclat.nextLine();
13         System.out.println("Introdueix l'any del teu naixement i l'actual");
14         int a1 = teclat.nextInt();
15         int a2 = teclat.nextInt();
16         System.out.print("Et dius " + nom);
17         System.out.println(" i tens " + (a2 - a1) + " anys");
18     }
19 }
```

A terminal window titled "BlueJ: BlueJ: Finestra de terminal - exemples - Scanner" showing the output of the Java program. The window has a title bar with standard OS controls (minimize, maximize, close) and a tab labeled "Opcions". The text inside the terminal is as follows:

```
Introdueix el teu nom
Lola Rovira
Introdueix l'any del teu naixement i l'actual
1982 2015
Et dius Lola Rovira i tens 33 anys
```

# Entrada des de teclat: la classe Scanner



BlueJ:exemples - Scanner

```
1 import java.util.Scanner;
2 /**
3  * Classe TestScannerLinea: classe programa per provar
4  * què ocorre quan es llegeix des de teclat un valor numèric
5  * i a continuació un String amb nextLine()
6  * @author IIP
7  * @version Curs 2015/16
8  */
9 public class TestScannerLinea {
10     public static void main(String[] args) {
11         Scanner teclat = new Scanner(System.in);
12         System.out.print("Introdueix un enter: ");
13         int n = teclat.nextInt();
14         //teclat.nextLine();
15         System.out.print("Introdueix una línia: ");
16         String s1 = teclat.nextLine();
17         System.out.print("Introdueix un altra línia: ");
18         String s2 = teclat.nextLine();
19         System.out.println("\nEnter: " + n);
20         System.out.println("Línia 1: " + s1);
21         System.out.println("Línia 2: " + s2);
22     }
23 }
```

BlueJ: BlueJ: Finestra de terminal - exemples - Scanner

Opcions

```
Introdueix un enter: 5
Introdueix una línia: Introdueix un altra línia: hola

Enter: 5
Línia 1:
Línia 2: hola
```

# Entrada des de teclat: la classe Scanner



BlueJ:exemples - Scanner

```
1 import java.util.Scanner;
2 import java.util.Locale;
3 /**
4  * Classe TestScannerLocale: classe programa per provar
5  * l'efecte del Locale quan es llegeix un double des de teclat
6  * @author IIP
7  * @version Curs 2015/16
8  */
9 public class TestScannerLocale {
10     public static void main(String[] args) {
11         Scanner teclat = new Scanner(System.in);
12         System.out.print("El teclat està configurat per defecte en ");
13         System.out.println(Locale.getDefault());
14
15         System.out.print("Escriu un nombre real (amb coma decimal): ");
16         double nReal1 = teclat.nextDouble();
17         System.out.println("El valor real llegit és " + nReal1);
18
19         teclat.useLocale(Locale.US);
20         System.out.print("Escriu un nombre real (amb punt decimal): ");
21         double nReal2 = teclat.nextDouble();
22         System.out.println("El valor real llegit és " + nReal2);
23     }
24 }
```

```
BlueJ: BlueJ: Finestra de terminal - exemples - Scanner
Opcions
El teclat està configurat per defecte en es_ES
Escriu un nombre real (amb coma decimal): 6,75
El valor real llegit és 6.75
Escriu un nombre real (amb punt decimal): 8.5
El valor real llegit és 8.5
```

# Entrada des de teclat: la classe Scanner



BlueJ:examples - Scanner

```
1 import java.util.Scanner;
2 /**
3  * Classe TestScannerChar: classe programa per provar
4  * com es llegeix un char des de teclat
5  * @author IIP
6  * @version Curs 2015/16
7  */
8 public class TestScannerChar {
9     public static void main(String[] args) {
10         Scanner teclat = new Scanner(System.in);
11         System.out.print("Introdueix un caràcter: ");
12         char c = teclat.next("\\S").charAt(0);
13         System.out.println("\nEl caràcter llegit és: " + c);
14     }
15 }
```

