

Entrada i eixida elemental

Introducció a la Informàtica i la Programació (IIP)

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 edició)"

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

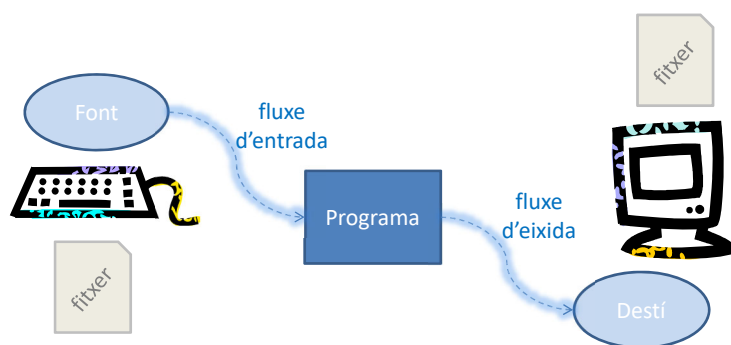


- **Descarrega** (del Tema 3 de PoliformaT) el fitxer ***exemples - Scanner.jar*** en ***W:\IIP\Tema 3***
- Des de l'opció **Projecte** de **BlueJ**, usa l'opció **Open ZIP/JAR...** per tal d'obrir aquest com un projecte **BlueJ** i prepara't per usar-lo



Introducció

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



Introducció

- Els fluxes predefinits 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, per exemple, 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, el següent a mostrar es mostra en una línia nova, mentre que amb print, 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).
```

03/09/2019



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
```

03/09/2019



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 caràcters 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, e1 %-4d, e1 %02d i e1 %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");
```

8

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.

Referència:

<https://docs.oracle.com/en/java/javase/11/docs/api/java.base/java/util/Scanner.html>

- Al començament del codi afegir: `import java.util.Scanner;`
- Declaració: `Scanner identificador = new Scanner(System.in);`
- S'usa `useLocale` per establir la configuració local d'un `Scanner`.
- Exemples d'ús:


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

```
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

- Alguns mètodes de la classe `Scanner`:

| Definició - capçalera | |
|---|--|
| <code>public Scanner(InputStream src)</code> | Crea un <code>Scanner</code> a partir d'un <code>InputStream</code> , com és <code>System.in</code> |
| <code>public Scanner(File src)</code> | Crea un <code>Scanner</code> a partir d'un <code>File</code> |
| <code>public Scanner useLocale(Locale l)</code> | Estableix la configuració local de l' <code>Scanner</code> a la configuració especificada pel <code>Locale l</code> |
| <code>public String next()</code> | Llig una paraula o token (seqüència de caràcters entre separadors -per defecte, blancs-) |
| <code>public String nextLine()</code> | Llig fins el final de la línia. Torna la línia llegida com un <code>String</code> , sense el bot de línia |
| <code>public byte nextByte()</code> | Lligen valors del tipus primitiu corresponent. Poden llançar l'excepció <code>InputMismatchException</code> en cas de tractar de llegir una dada que no corresponga amb el tipus sol·licitat |
| <code>public short nextShort()</code> | |
| <code>public int nextInt()</code> | |
| <code>public long nextLong()</code> | |
| <code>public float nextFloat()</code> | |
| <code>public double nextDouble()</code> | |
| <code>public boolean nextBoolean()</code> | |

Entrada des de teclat: la classe 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 Llibre IIP-PRG
6  * @version Curs 2019/20
7  */
8 public class TestScanner {
9     // No s'usen objectes d'aquesta classe
10    private TestScanner() { }
11
12    public static void main(String[] args) {
13        Scanner teclat = new Scanner(System.in);
14        System.out.println("Introdueix el teu nom");
15        String nom = teclat.nextLine();
16        System.out.println("Introdueix l'any del teu naixement i l'actual");
17        int a1 = teclat.nextInt();
18        int a2 = teclat.nextInt();
19        System.out.print("Et dius " + nom);
20        System.out.println(" i tens " + (a2 - a1) + " anys");
21    }
22 }

```

BlueJ: BlueJ: Finestra de terminal - exemples - Scanner

Options

Introdueix el teu nom
 Lola Rovira
 Introdueix l'any del teu naixement i l'actual
 1982 2019
 Et dius Lola Rovira i tens 37 anys

Can only enter input while your programming is running

03/09/2019



IIP - Cur

Entrada des de teclat: la classe scanner

```

1 import java.util.Scanner;
2 /**
3  * Classe TestScannerLinha: 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 Llibre IIP-PRG
7  * @version Curs 2019/20
8  */
9 public class TestScannerLinha {
10    // No s'usen objectes d'aquesta classe
11    private TestScannerLinha() { }
12
13    public static void main(String[] args) {
14        Scanner teclat = new Scanner(System.in);
15        System.out.print("Introdueix un enter: ");
16        int n = teclat.nextInt();
17        //teclat.nextLine();
18        System.out.print("Introdueix una línia: ");
19        String s1 = teclat.nextLine();
20        System.out.print("Introdueix un altra línia: ");
21        String s2 = teclat.nextLine();
22        System.out.println("\nEnter: " + n);
23        System.out.println("Línia 1: " + s1);
24        System.out.println("Línia 2: " + s2);
25    }
26 }

```

BlueJ: BlueJ: Finestra de terminal - exemples - Scanner

Options

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

Enter: 5
 Línia 1:
 Línia 2: hola

Can only enter input while your programming is running

03/09/2019



IIP -

Entrada des de teclat: la classe scanner

```

1 import java.util.Locale;
2 import java.util.Scanner;
3 /**
4  * Classe TestScannerLocale: classe programa per provar
5  * l'efecte del Locale quan es llegeix un double des de teclat
6  * @author Llibre IIP-PRG
7  * @version Curs 2019/20
8  */
9 public class TestScannerLocale {
10     // No s'usen objectes d'aquesta classe
11     private TestScannerLocale() { }
12
13     public static void main(String[] args) {
14         Scanner teclat = new Scanner(System.in);
15         System.out.print("El teclat està configurat per defecte en ");
16         System.out.println(Locale.getDefault());
17
18         System.out.print("Escriu un nombre real (amb coma decimal): ");
19         double nReal1 = teclat.nextDouble();
20         System.out.println("El valor real llegit és " + nReal1);
21
22         teclat.useLocale(Locale.US);
23         System.out.print("Escriu un nombre real (amb punt decimal): ");
24         double nReal2 = teclat.nextDouble();
25         System.out.println("El valor real llegit és " + nReal2);
26     }
27 }

```

BlueJ:exemples - Scanner

Options

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

03/09/2019

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

IIP - Curs 2

Entrada des de teclat: la classe 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 Llibre IIP-PRG
6  * @version Curs 2019/20
7  */
8 public class TestScannerChar {
9     // No s'usen objectes d'aquesta classe
10    private TestScannerChar() { }
11
12    public static void main(String[] args) {
13        Scanner teclat = new Scanner(System.in);
14        System.out.print("Introdueix un caràcter: ");
15        char c = teclat.next("\\S").charAt(0);
16        System.out.println("\nEl caràcter llegit és: " + c);
17    }
18 }

```

BlueJ:exemples - Scanner

Options

Introdueix un caràcter: k
 El caràcter llegit és: k
 Can only enter input while your p

BlueJ:exemples - Scanner

Options

Introdueix un caràcter: una cadena en lloc d'un unic caràcter
 Can only enter input while your programming is running

java.util.InputMismatchException
 at java.base/java.util.Scanner.throwFor(Scanner.java:939)
 at java.base/java.util.Scanner.next(Scanner.java:1594)
 at java.base/java.util.Scanner.next(Scanner.java:1525)
 at TestScannerChar.main(TestScannerChar.java:15)

03/09/2019

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Exercici: Entrada y Salida de datos

CAP: Entrada y Salida de datos (clau CCDGJ4ai)

- Llegir dades de l'entrada estàndard mitjançant un *Scanner* i mostrar-les en un determinat format per l'eixida estàndard (*System.out*).
- Usar alguns mètodes de la classe *String*