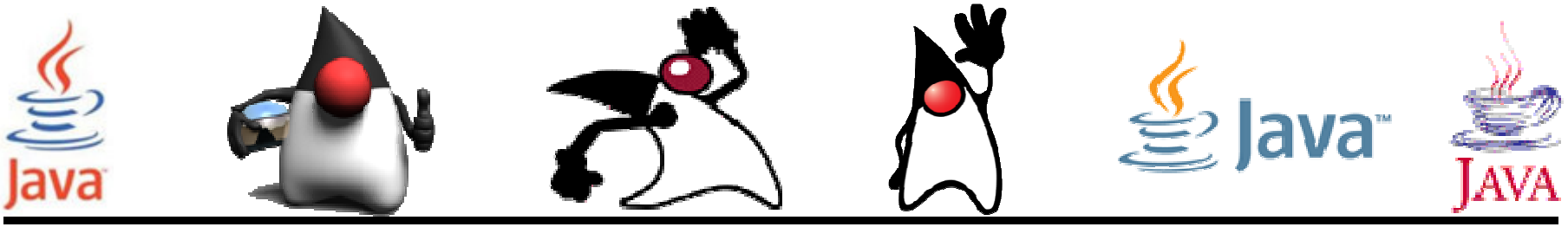


Introduciendo datos desde el teclado

Pedro Corcuera

Dpto. Matemática Aplicada y
Ciencias de la Computación

Universidad de Cantabria

corcuerp@unican.es

Objetivos

- Crear programas interactivos que lean datos desde el teclado.
- Usar la clase BufferedReader para leer datos desde el teclado usando la consola.
- Usar la clase Scanner para leer datos desde el teclado usando la consola.
- Usar la clase JOptionPane para leer datos desde el teclado usando una interfaz gráfica.

Índice

- Streams I/O
- Lectura de datos desde teclado
- Uso de la clase BufferedReader
- Uso de la clase Scanner
- Uso de la clase JOptionPane

Streams I/O

- Un stream es una secuencia de caracteres o bytes utilizados para la entrada o salida de un programa.
- Java proporciona varias clases Stream diferentes de entrada y salida en la API java.io.
- Los objetos I/O más utilizados son:
  System.in (input stream, conectado al teclado)
  System.out (output stream, resultados normales en monitor)
  System.err (output stream para mensajes de error - monitor)
- Para convertir datos String a otros tipos se usan clases wrapper.

Lectura de datos desde teclado

- Tres métodos de lectura:
  - Clase BufferedReader (consola)
  - Clase Scanner (consola)
  - Clase JOptionPane (interfaz gráfica de usuario)

Uso de la clase BufferedReader

- Clase BufferedReader
  - Se encuentra en el paquete java.io
  - Se usa para lectura de datos

Pasos para leer datos con BufferedReader

- Añadir la siguiente línea al inicio del código:

```
import java.io.*;
```

- Añadir la siguiente instrucción:

```
BufferedReader dataIn = new BufferedReader( new
        InputStreamReader(System.in) );
```

Pasos para leer datos con BufferedReader

- Declarar una variable temporal de tipo String para almacenar la entrada, e invocar el método readLine() para leer del teclado. Es necesario escribirlo dentro de un bloque try-catch:

```
try {
    String temp = dataIn.readLine();
} catch( IOException e ){
    System.out.println("Error de lectura");
}
```

Ejemplo de lectura con BufferedReader

```
import java.io.*;

public class GetInputFromKeyboard {
  public static void main( String[] args ){
    BufferedReader dataIn = new BufferedReader(new
        InputStreamReader( System.in ) );
    String name = "";
    System.out.print("Escribe tu nombre:");
    try{
      name = dataIn.readLine();
    }catch( IOException e ){
      System.out.println("Error!");
    }
    System.out.println("Hola " + name +"!");
  }
}
```

Ejemplo de lectura con BufferedReader

```
import java.io.BufferedReader;
import java.io.InputStreamReader;
import java.io.IOException;
```

import java.io.*;

```
public class GetInputFromKeyboard {
  public static void main( String[] args ){
    BufferedReader dataIn = new BufferedReader(new
        InputStreamReader( System.in) );
    String name = "";
    System.out.print("Escribe tu nombre:");
.
.
.
```

Ejemplo de lectura con BufferedReader

- Las líneas

```
import java.io.BufferedReader;
import java.io.InputStreamReader;
import java.io.IOException;
```

  indican que deseamos usar las clases BufferedReader, InputStreamReader y IOException que están dentro del paquete (package) java.io.

- Estas instrucciones también se pueden escribir como

```
import java.io.*;
```

Análisis del ejemplo: API de Java

- La interfaz de programación de aplicaciones de Java (API – Application Programming Interface) contiene cientos de clases predefinidas que se pueden usar en los programas.
- Las clases de la API se organizan en lo que se llaman paquetes (packages) que contienen clases con un propósito relacionado.
- Se llaman en un programa con

```
import nombre_package
```

Análisis del ejemplo con BufferedReader

- La sentencia

```
public class GetInputFromKeyboard {
```

declara una clase llamada GetInputFromKeyboard

- La siguiente sentencia declara el método main

```
public static void main( String[] args ){
```

- La sentencia

```
BufferedReader dataIn = new BufferedReader(new
        InputStreamReader( System.in) );
```

declara una variable dataIn con el tipo de clase BufferedReader.

Análisis del ejemplo con BufferedReader

- La sentencia

```
String name = "";
```

declara una variable name de tipo String.

- La siguiente sentencia

```
System.out.print("Escribe tu nombre:");
```

imprime la cadena "Escribe tu nombre" en pantalla.

Análisis del ejemplo con BufferedReader

- El bloque siguiente define un bloque try-catch

```
try{
    name = dataIn.readLine();
}catch( IOException e ){
    System.out.println("Error!");
}
```

Esto asegura que las posibles excepciones que pueden ocurrir en la sentencia

```
name = dataIn.readLine();
```

serán capturadas.

Análisis del ejemplo con BufferedReader

- En la sentencia

```
name = dataIn.readLine();
```

  la llamada al método dataIn.readLine(), lee la entrada del usuario y devuelve un String que se asigna a la variable name.

- El contenido de la variable name se usa para saludar al usuario en la sentencia

```
System.out.println("Hola " + name +"!");
```

Uso de la clase Scanner

- Otra manera de leer la entrada de usuario es usar la clase Scanner que se encuentra en el paquete java.util
- La clase Scanner permite leer valores de varios tipos.
- Algunos de los métodos más usados de Scanner son:

Método	Valor devuelto
nextInt()	siguiente token como un integer
nextDouble()	siguiente token como un double
next()	siguiente token como un String
nextLine()	línea entera (o el resto de la línea) como un String

Pasos para el uso de la clase Scanner

- Importar la clase Scanner que está en el paquete java.util

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

- Declarar un objecto de la clase Scanner

```
Scanner in = new Scanner(System.in);
```

- Usar métodos del objeto Scanner para leer:

```
int piezas = in.nextInt();
double volumen = in.nextDouble();
String mensaje = in.next();
```

Ejemplo de lectura con Scanner

```
import java.util.Scanner;

public class InputwithScanner
{
   public static void main(String[] args)
   {
      final double LITROS_POR_ONZA = 0.0296;
      final double VOLUMEN_DEPOS = 12 * LITROS_POR_ONZA;
      // Mensaje
      System.out.print("Ingresar numero de depositos: ");
      // Lectura del numero de depositos 1
      Scanner in = new Scanner(System.in);
      int depositos = in.nextInt();
      // Calculo del volumen total
      double Volumentotal = depositos * VOLUMEN_DEPOS;
      // Impresion de resultados
      System.out.print("Volumen del deposito: " + Volumentotal);
   }
}
```

Uso de la clase JOptionPane

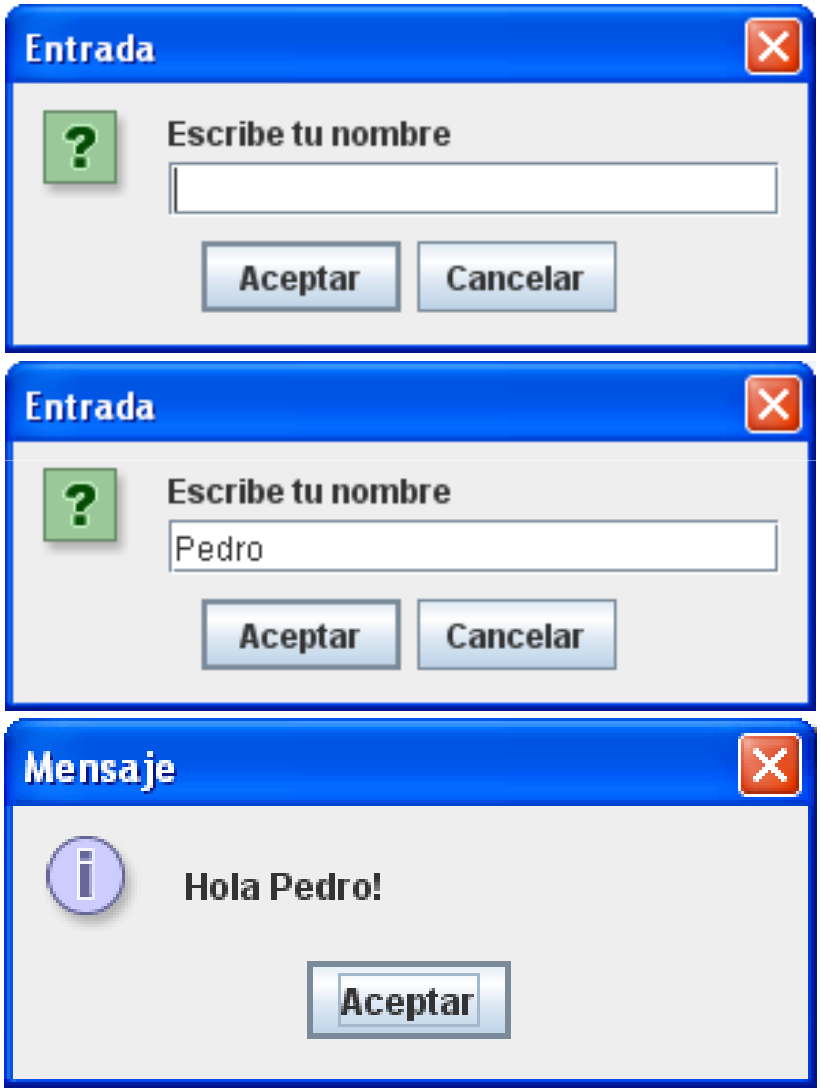
- Otra manera de leer la entrada de usuario es usar la clase JOptionPane que se encuentra en el paquete javax.swing
- Con JOptionPane es fácil crear una ventana de diálogo estándar que solicita al usuario por un valor o informa de algo.

Ejemplo de lectura con JOptionPane

```
import javax.swing.JOptionPane;

public class InputwithJOptionPane {
  public static void main( String[] args ){
    String name = "";
    name = JOptionPane.showInputDialog(
           "Escribe tu nombre");
    String msg = "Hola " + name + "!";
    JOptionPane.showMessageDialog(null, msg);
  }
}
```

Ejemplo de lectura con JOptionPane



Análisis del ejemplo con JOptionPane

- La sentencia

```
import javax.swing.JOptionPane;
```

  indica que queremos importar la clase JOptionPane del paquete javax.swing.

- También puede escribirse como:

```
import javax.swing.*;
```

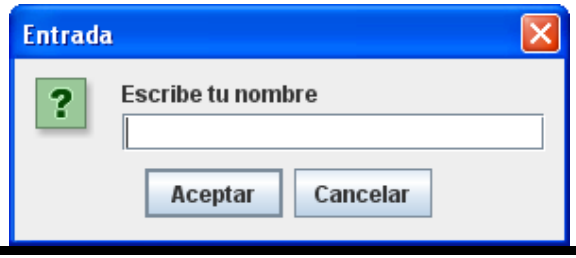
Análisis del ejemplo con JOptionPane

- La sentencia

```
name = JOptionPane.showInputDialog(
            "Escribe tu nombre");
```

  crea un diálogo de entrada JOptionPane, que muestra un diálogo con un mensaje, un campo de texto (textfield) y dos botones (Aceptar, Cancelar).

- La cadena de caracteres que escribe el usuario se devolverá y asignará a la variable name.



Análisis del ejemplo con JOptionPane

- La sentencia

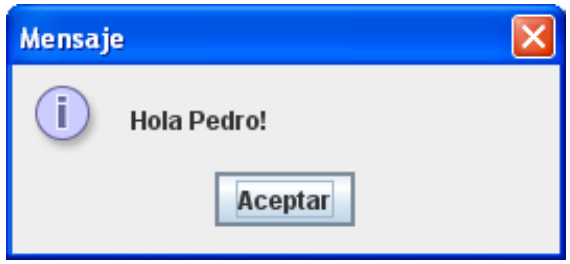
```
String msg = "Hola " + name + "!";
```

crea un mensaje de bienvenida que se almacena en la variable msg.

- La sentencia

```
JOptionPane.showMessageDialog(null, msg);
```

muestra un diálogo que contiene un mensaje y un botón de Aceptar.



Código de los ejemplos presentados

http://personales.unican.es/corcuerp/Java/Labs/codigo/GetInputFromKeyboard.java

http://personales.unican.es/corcuerp/Java/Labs/codigo/InputwithScanner.java

http://personales.unican.es/corcuerp/Java/Labs/codigo/InputwithJOptionPane.java