

Instituto Politécnico Nacional
Escuela Superior de Cómputo





Análisis y diseño de algoritmos, 3CV1

"Creación y Uso de la Clase Lectura Números para Lectura de Datos Numéricos"

Profesor. García Floriano Andrés
Fecha de realización: 17 de Mayo 2024

Elaborado por:

- Elizalde Hernández Alan
- Reyes Ruíz Yoselyn Estefany
- Solares Velasco Arturo Misael
- Solis Lugo Mayra
- Toral Hernández Leonardo Javier



CREACIÓN Y USO DE LA CLASE LECTURANUMEROS PARA LECTURA DE DATOS NUMÉRICOS

El objetivo de esta práctica es crear una clase `LecturaNumeros` que herede de `BufferedReader` y que añada métodos específicos para facilitar la lectura de datos numéricos (`int`, `Integer` y `double`). Además, se busca implementar una clase de prueba que utilice `LecturaNumeros` para solicitar números al usuario y desplegarlos en pantalla.

La clase `LecturaNumeros` extiende la funcionalidad de `BufferedReader` y agrega métodos adicionales para la lectura de números. Los métodos adicionales definidos son:

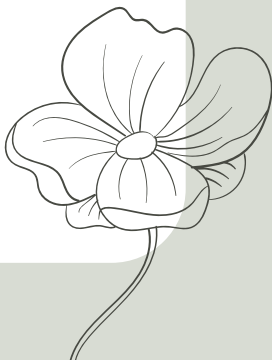
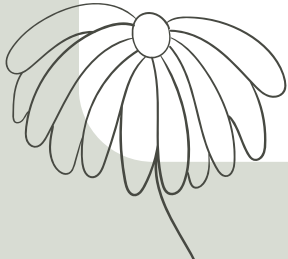
- `int readInt()`: Lee y devuelve un número entero.
- `int readInt(String mensaje)`: Muestra un mensaje antes de leer y devolver un número entero.
- `Integer readInteger()`: Lee y devuelve un número entero como un objeto `Integer`.
- `double readDouble()`: Lee y devuelve un número decimal.
- `double readDouble(String mensaje)`: Muestra un mensaje antes de leer y devolver un número decimal.


Se implementaron dos constructores:

- `LecturaNumeros()`: Prepara el objeto para leer desde el teclado utilizando `InputStreamReader` y `System.in`.
- `LecturaNumeros(Reader r)`: Permite crear un objeto `LecturaNumeros` con cualquier `Reader` especificado.

IMPLEMENTACIÓN DEL CÓDIGO

En la clase de prueba, se crea una instancia de `LecturaNumeros` y se utilizan los métodos definidos para solicitar al usuario cinco números (dos `int`, un `Integer`, y dos `double`). Los números ingresados se despliegan en pantalla.



practica 8 >  LecturaNumeros.java

```
1  import java.io.BufferedReader;
2  import java.io.InputStreamReader;
3  import java.io.IOException;
4  import java.io.Reader;
5
6  public class LecturaNumeros extends BufferedReader {
7
8      // Constructor que prepara la lectura por medio del teclado
9      public LecturaNumeros() {
10         super(new InputStreamReader(System.in));
11     }
12
13     // Constructor que recibe un objeto Reader
14     public LecturaNumeros(Reader r) {
15         super(r);
16     }
17
18     // Método que lee un int de la entrada estándar
19     public int readInt() throws IOException {
20         return Integer.parseInt(readLine());
21     }
22
23     // Método que lee un int de la entrada estándar con un mensaje
24     public int readInt(String mensaje) throws IOException {
25         System.out.print(mensaje);
26         return readInt();
27     }
28
29     // Método que lee un Integer de la entrada estándar
30     public Integer readInteger() throws IOException {
```


```

6  public class LecturaNumeros extends BufferedReader {
30      public Integer readInteger() throws IOException {
32          }
33
34      // Método que lee un double de la entrada estándar
35      public double readDouble() throws IOException {
36          return Double.parseDouble(readLine());
37      }
38
39      // Método que lee un double de la entrada estándar con un mensaje
40      public double readDouble(String mensaje) throws IOException {
41          System.out.print(mensaje);
42          return readDouble();
43      }
44
45      Run | Debug
46      public static void main(String[] args) {
47          try {
48              LecturaNumeros ln = new LecturaNumeros();
49              try {
50                  int num1 = ln.readInt(mensaje:"Introduce el primer número entero: ");
51                  int num2 = ln.readInt(mensaje:"Introduce el segundo número entero: ");
52                  Integer num3 = ln.readInt(mensaje:"Introduce el tercer número entero (como Integer): ");
53                  double num4 = ln.readDouble(mensaje:"Introduce el primer número decimal: ");
54                  double num5 = ln.readDouble(mensaje:"Introduce el segundo número decimal: ");
55
56                  System.out.println("Números ingresados:");
57                  System.out.println("Entero 1: " + num1);
58                  System.out.println("Entero 2: " + num2);
59                  System.out.println("Entero (Integer): " + num3);
60                  System.out.println("Decimal 1: " + num4);
61                  System.out.println("Decimal 2: " + num5);
62              } finally {
63                  ln.close();
64              }
65
66          } catch (IOException e) {
67              e.printStackTrace();
68          }
69      }
70  }
71

```

CLASE DE PRUEBA

En la clase de prueba, se crea una instancia de `LecturaNumeros` y se utilizan los métodos definidos para solicitar al usuario cinco números (dos int, un Integer, y dos double). Los números ingresados se despliegan en pantalla.



La ejecución del programa solicita al usuario que ingrese cinco números con los siguientes mensajes:

- "Introduce el primer número entero: "
- "Introduce el segundo número entero: "
- "Introduce el tercer número entero (como Integer): "
- "Introduce el primer número decimal: "
- "Introduce el segundo número decimal: "

Ejemplo de ejecución

```
Introduce el primer número entero: 25
Introduce el segundo número entero: 50
Introduce el tercer número entero (como Integer): 75
Introduce el primer número decimal: 10.5
Introduce el segundo número decimal: 20.75
Números ingresados:
Entero 1: 25
Entero 2: 50
Entero (Integer): 75
Decimal 1: 10.5
Decimal 2: 20.75
```

Después de que el usuario ingresa los números, el programa los muestra en pantalla correctamente

CONCLUSIÓN

Esta práctica permitió crear una clase especializada en la lectura de datos numéricos extendiendo la funcionalidad de `BufferedReader`. Los métodos adicionales facilitan la interacción con el usuario al permitir la lectura de diferentes tipos numéricos con mensajes personalizados. La implementación de una clase de prueba demostró la utilidad y correcta funcionalidad de la clase `LecturaNumeros`. Este ejercicio es útil para comprender cómo extender clases existentes y agregar funcionalidades específicas mediante la herencia en Java.

