

Programación Orientada a Objetos

Sesión 1:

Diseño de diagramas de clases UML

Recordando la clase anterior

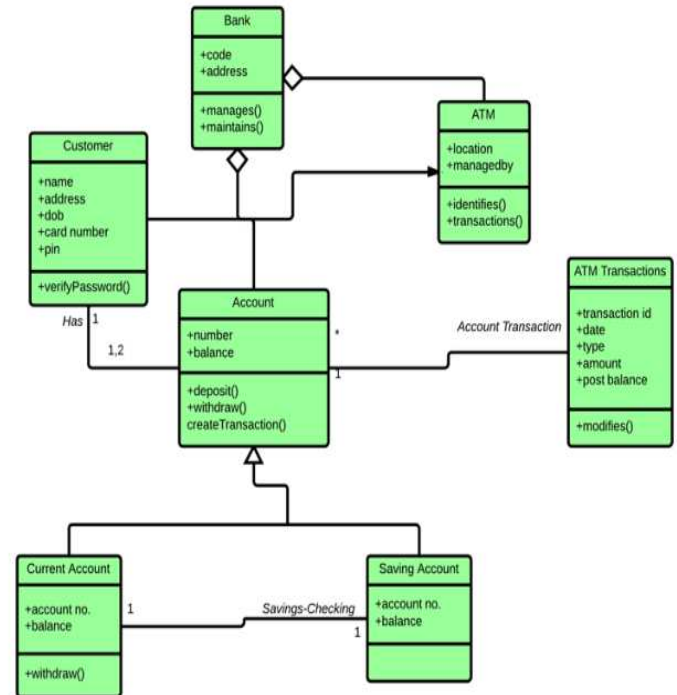
La clase anterior vimos el tema:
Diagrama de clases

Logro de la sesión

Al finalizar la sesión el alumno analiza y realiza diagramas de clase mediante el uso del Lenguaje UML realizando ejercicios en java.

Utilidad

- Expresar visualmente cualesquier necesidades específicas de un sistema.
- Comprender mejor la visión general de los esquemas de una aplicación.



Sintaxis de una clase en Java

```
public class Persona {
```

También denominadas
atributos o campos
(fields)

```
private String nombre;  
private int edad;
```

Este método sirve para
definir qué ocurre
cuando se crea un
objeto.

Métodos de la
clase

```
public void inicializar() {...}  
public void imprimir() {...}  
public void esMayorEdad() {...}  
public static void main(String[ ] ar) {...}
```

```
}
```

Ejemplo1 - Implementar la clase Persona en BlueJ

```
public class Persona {
```

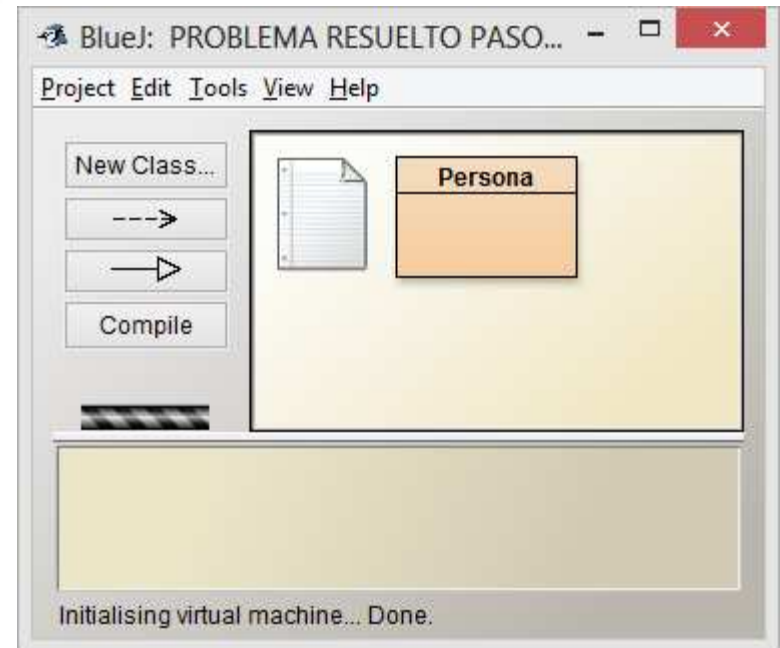
```
    private String nombre;  
    private int edad;
```

```
    public void inicializar() {...}  
    public void imprimir() {...}  
    public void esMayorEdad() {...}  
    public static void main(String[ ] ar) {...}
```

```
}
```



Creamos un nuevo proyecto en BlueJ:



Ejemplo1 - Implementar la clase Persona en BlueJ

```
public class Persona {
```

```
    private String nombre;  
    private int edad;
```

```
    public void inicializar() {...}
```

```
    public void imprimir() {...}
```

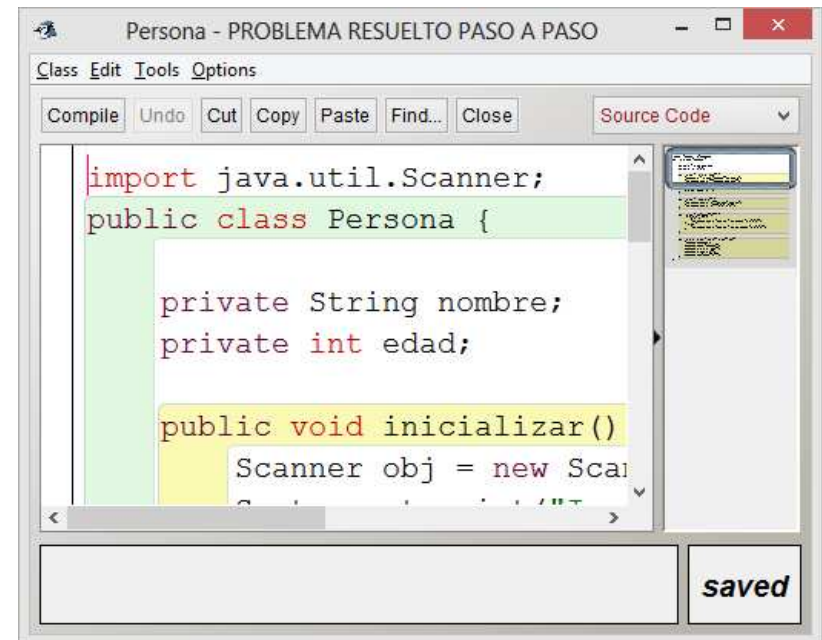
```
    public void esMayorEdad() {...}
```

```
    public static void main(String[ ] ar) {...}
```

```
}
```



Implementamos la clase Persona:



Ejemplo1 - Implementar la clase Persona en BlueJ

```
public class Persona {
```

```
    private String nombre;  
    private int edad;
```

```
    public void inicializar() {...}  
    public void imprimir() {...}  
    public void esMayorEdad() {...}  
    public static void main(String[ ] ar) {...}
```

```
}
```



Salida del Programa:

```
BlueJ: Terminal Window - PROBLEMA RESUEL... - □ ×  
Options  
Ingrese nombre:Pepe  
Ingrese edad:40|
```

```
BlueJ: Terminal Window - PROBLEMA RESUEL... - □ ×  
Options  
Ingrese nombre:Pepe  
Ingrese edad:40  
Nombre:Pepe  
Edad:40  
Pepe es mayor de edad.
```


Ejemplo1 - Implementar la clase Persona en NetBeans

The screenshot displays the NetBeans IDE 8.2 interface. The top menu bar includes Archivo, Editar, Ver, Navegar, Source, Reestructurar, Ejecutar, Depurar, Profile, Team, Herramientas, Ventana, and Ayuda. The toolbar contains various icons for file operations, running, and debugging.

The left sidebar shows the project structure for 'PersonaEnNetBeans'. It includes 'Paquetes de fuentes' (source packages) with 'personaennetbeans' containing 'Persona.java' and 'Prueba.java', and 'Paquetes de prueba' (test packages) with 'Bibliotecas' and 'Bibliotecas de pruebas'.

The main editor window shows the 'Source' view of 'Persona.java'. The code is as follows:

```
1 package personaennetbeans;
2 import java.util.Scanner;
3 public class Persona
4 {
5     private String nombre;
6     private int edad;
7     public Persona() {...7 lines }
14    public void imprimir() {...5 lines }
19    public void esMayor() {...8 lines }
27 }
```

The 'esMayor()' method is highlighted in yellow. The bottom status bar shows 'Salida' (Output) with two tabs: 'PersonaEnNetBeans (run)' and 'PersonaEnNetBeans (run) #2'. The output window displays the following text:

```
Ingresa nombre: Lucy
Ingresa edad: 24

Nombre: Lucy
Edad: 24
Lucy es mayor de edad
```

The bottom left corner shows the 'esMayor - Navegador' (Navigator) window, which lists the members of the 'Persona' class: 'Persona()', 'esMayor()', and 'imprimir()'.

¿Cómo se crea un objeto en Java?

NombreClase objeto = **new** NombreClase(parámetros);

```
public void inicializar() {  
    Scanner obj=new Scanner(System.in);  
    System.out.print("Ingrese nombre:");  
    nombre = obj.next();  
    System.out.print("Ingrese edad:");  
    edad = obj.nextInt();  
}
```

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

```
    Persona persona1;
```

```
    persona1 = new Persona();
```

```
    persona1.inicializar();
```

```
    persona1.imprimir();
```

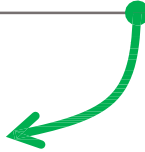
```
    persona1.esMayorEdad();
```

```
}
```

¿Qué es un constructor de una clase?

Un *Constructor* es un método de una clase, el cual es llamado automáticamente cuando se crea un objeto de esa clase.

```
public void Persona() {  
    Scanner obj=new Scanner(System.in);  
    System.out.print("Ingrese nombre:");  
    nombre = obj.next();  
    System.out.print("Ingrese edad:");  
    edad = obj.nextInt();  
}
```

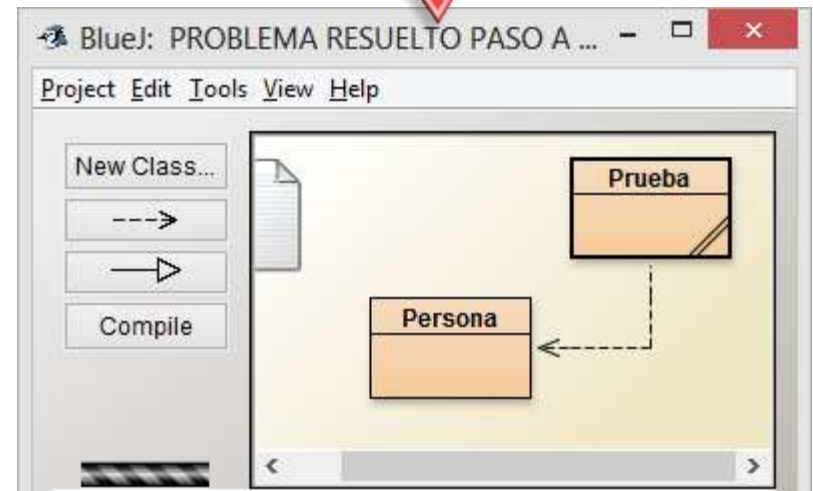


```
public static void main(String[ ] ar){  
    Persona persona1;  
  
    persona1 = new Persona();  
  
    persona1.inicializar();  
    persona1.imprimir();  
    persona1.esMayorEdad();  
}
```

Relación de Dependencia entre clases

```
public class Persona {  
    private String nombre;  
    private int edad;  
  
    public void inicializar() {...}  
    public void imprimir() {...}  
    public void esMayorEdad() {...}  
    public static void main(String[] ar) {...}  
}
```

```
public class Prueba {  
    public static void main(String[] ar) {...}  
}
```



Estructura de un archivo en java

```
class Clase1 {  
    ...  
}  
  
class Clase2 {  
    ...  
}  
  
class ClaseN {  
    public static void main(String [ ] args){  
        ...  
    }  
    ...  
}
```

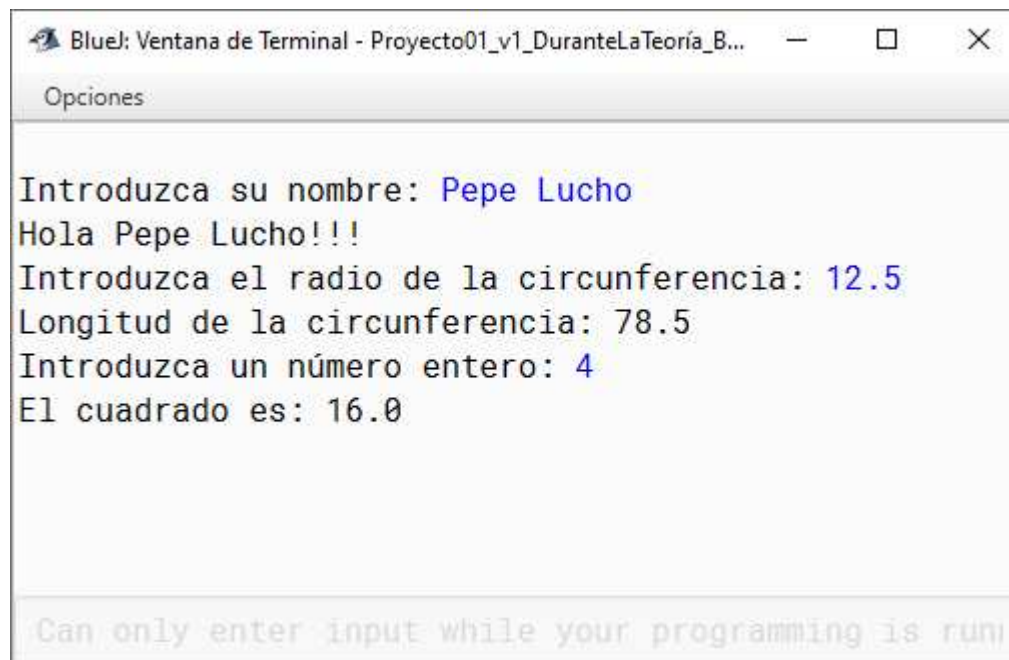
Un **programa Java** estará formado por uno o varios ficheros fuente y en cada uno de ellos habrá definida una o varias clases.

Para que un programa se *pueda ejecutar* debe contener una clase que tenga un método llamado **main**.

Ejemplo 02

Escribir un programa usando **BlueJ** y **NetBeans** que permita ingresar:

- Un nombre y muestre un saludo.
- El radio (*double*) de una circunferencia y muestre la longitud de la misma.
- Un numero *entero* y muestre el cuadrado del mismo.



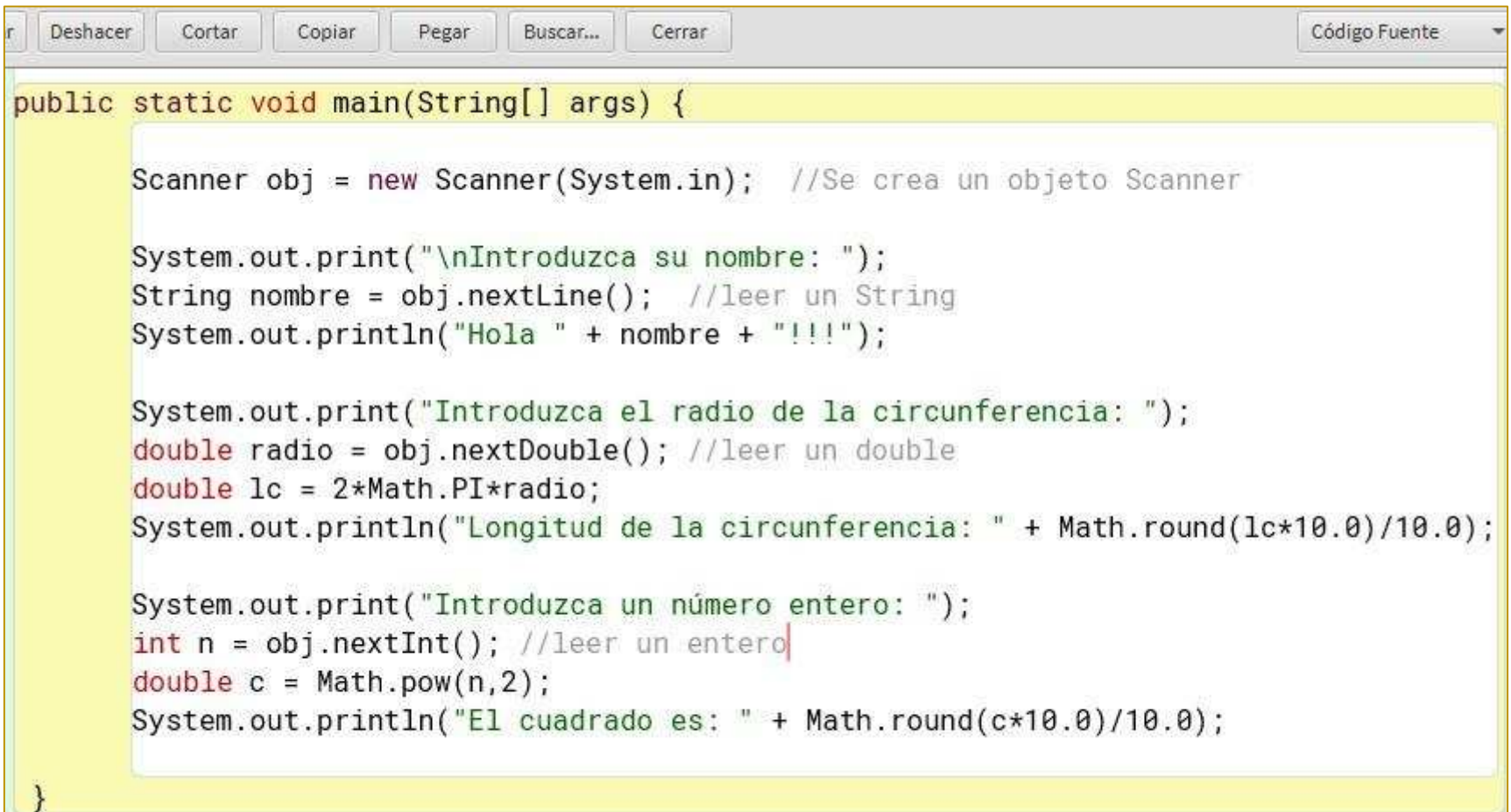
The screenshot shows a terminal window titled "BlueJ: Ventana de Terminal - Proyecto01_v1_DuranteLaTeoría_B...". The window has a menu bar with "Opciones" and standard window controls. The terminal content shows the following sequence of input and output:

```
Introduzca su nombre: Pepe Lucho
Hola Pepe Lucho!!!
Introduzca el radio de la circunferencia: 12.5
Longitud de la circunferencia: 78.5
Introduzca un número entero: 4
El cuadrado es: 16.0
```

At the bottom of the window, a greyed-out status bar reads: "Can only enter input while your programming is runi".

Ejemplo 02 - Implementación en BlueJ

Primera forma:



The screenshot shows the BlueJ IDE interface. At the top, there is a menu bar with buttons: 'Deshacer', 'Cortar', 'Copiar', 'Pegar', 'Buscar...', and 'Cerrar'. On the right side of the menu bar is a dropdown menu labeled 'Código Fuente'. The main area of the IDE displays a Java program with the following code:

```
public static void main(String[] args) {  
  
    Scanner obj = new Scanner(System.in); //Se crea un objeto Scanner  
  
    System.out.print("\nIntroduzca su nombre: ");  
    String nombre = obj.nextLine(); //leer un String  
    System.out.println("Hola " + nombre + "!!!");  
  
    System.out.print("Introduzca el radio de la circunferencia: ");  
    double radio = obj.nextDouble(); //leer un double  
    double lc = 2*Math.PI*radio;  
    System.out.println("Longitud de la circunferencia: " + Math.round(lc*10.0)/10.0);  
  
    System.out.print("Introduzca un número entero: ");  
    int n = obj.nextInt(); //leer un entero  
    double c = Math.pow(n,2);  
    System.out.println("El cuadrado es: " + Math.round(c*10.0)/10.0);  
  
}
```

Ejemplo 02 - Implementación en BlueJ

En la solución de este hemos usado la clase **Math** de Java.

La clase **Math** nos ofrece varios métodos y constantes *estáticos*, que podemos utilizar tan sólo anteponiendo el nombre de la clase, (**Math.metodo** o **Math.Constante**). Además podemos omitir el nombre de la clase si la importamos:

```
import static java.lang.Math. *;
```

Math.PI → 3.14159265358979323846

Math.pow(b,n) → devuelve b elevado a la n.

Ejemplo 02 - Implementación en BlueJ

Segunda forma:

```
import java.util.Scanner;
public class MiClase {
    private String nombre;
    private double radio;
    private int n;

    public void ingreso(){

    }

    public void saludo(){

    }

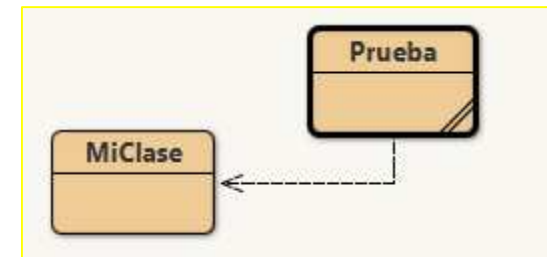
    public void longitudCircunferencia(){

    }

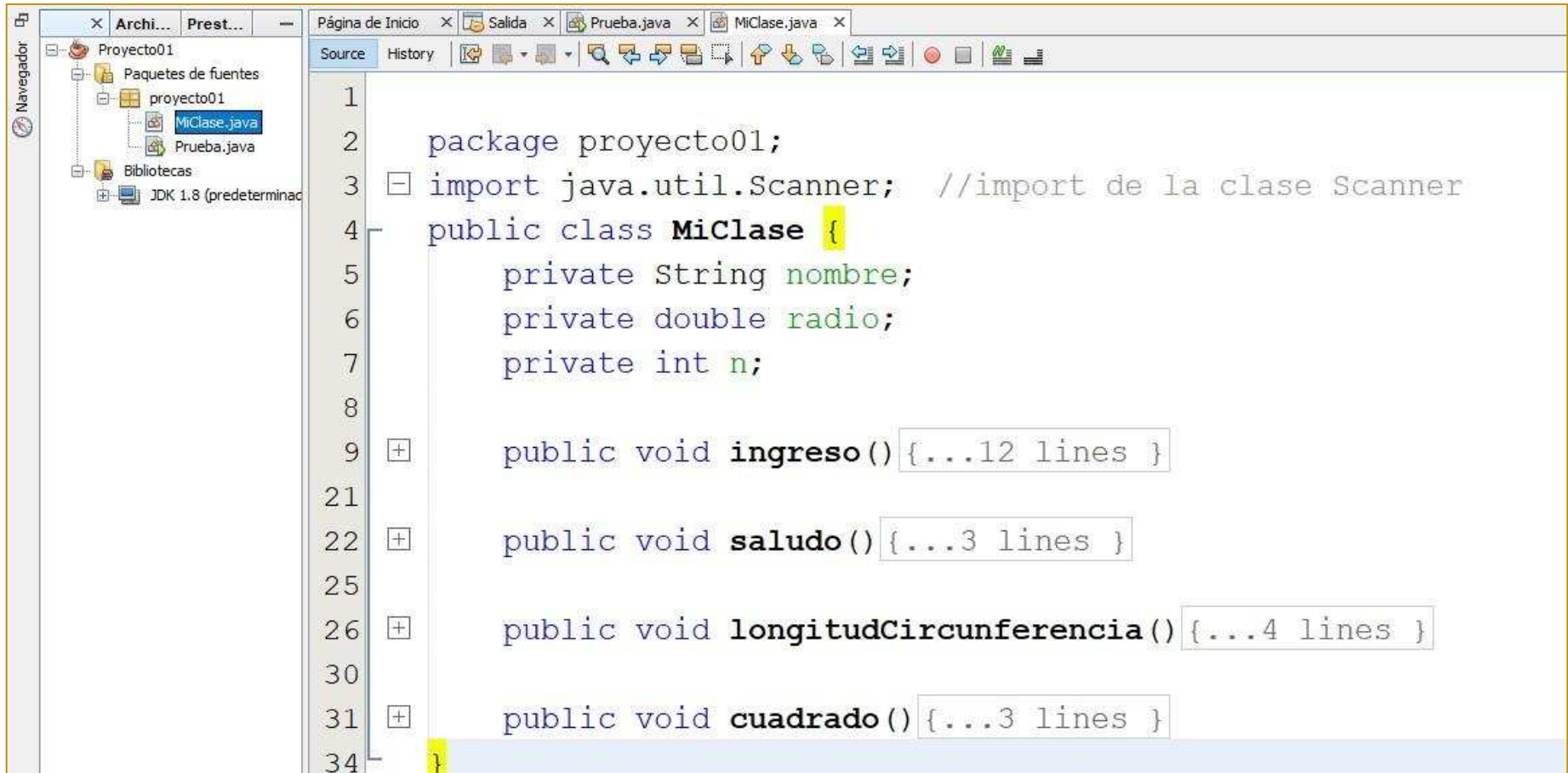
    public void cuadrado(){

    }
}
```

```
public class Prueba
{
    public static void main(String[] args) {
        MiClase obj = new MiClase();
        obj.ingreso();
        obj.saludo();
        obj.longitudCircunferencia();
        obj.cuadrado();
    }
}
```



Ejemplo 02 - Implementación en NetBeans



Ejemplo 03 - Implementación en BlueJ y NetBeans

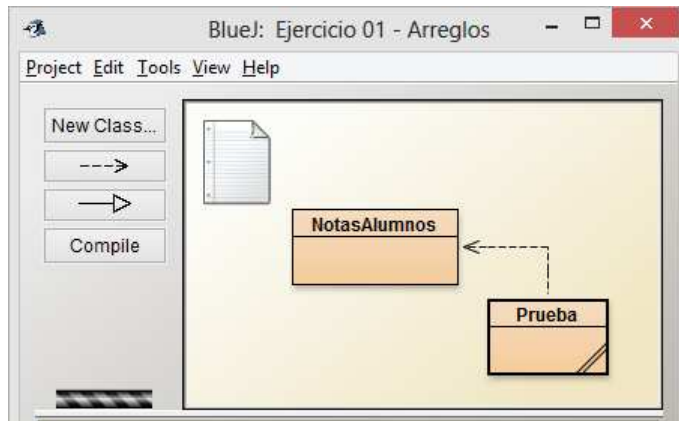
Escribir un programa que solicite la carga de un valor positivo “**n**” y nos muestre todos los números desde 1 hasta el valor ingresado (de uno en uno). *Ejemplo:*

Si ingresamos 30 se debe mostrar en pantalla los números del 1 al 30.

Ejemplo 04 - Implementación en BlueJ y NetBeans

Desarrollar un programa que lea por teclado las notas de “n” alumnos de una clase, calcule la nota media del grupo, muestre las notas superiores a la media. El valor de “n” se lee por teclado.

El Diagrama de Clases en BlueJ sería:



La salida del programa sería:

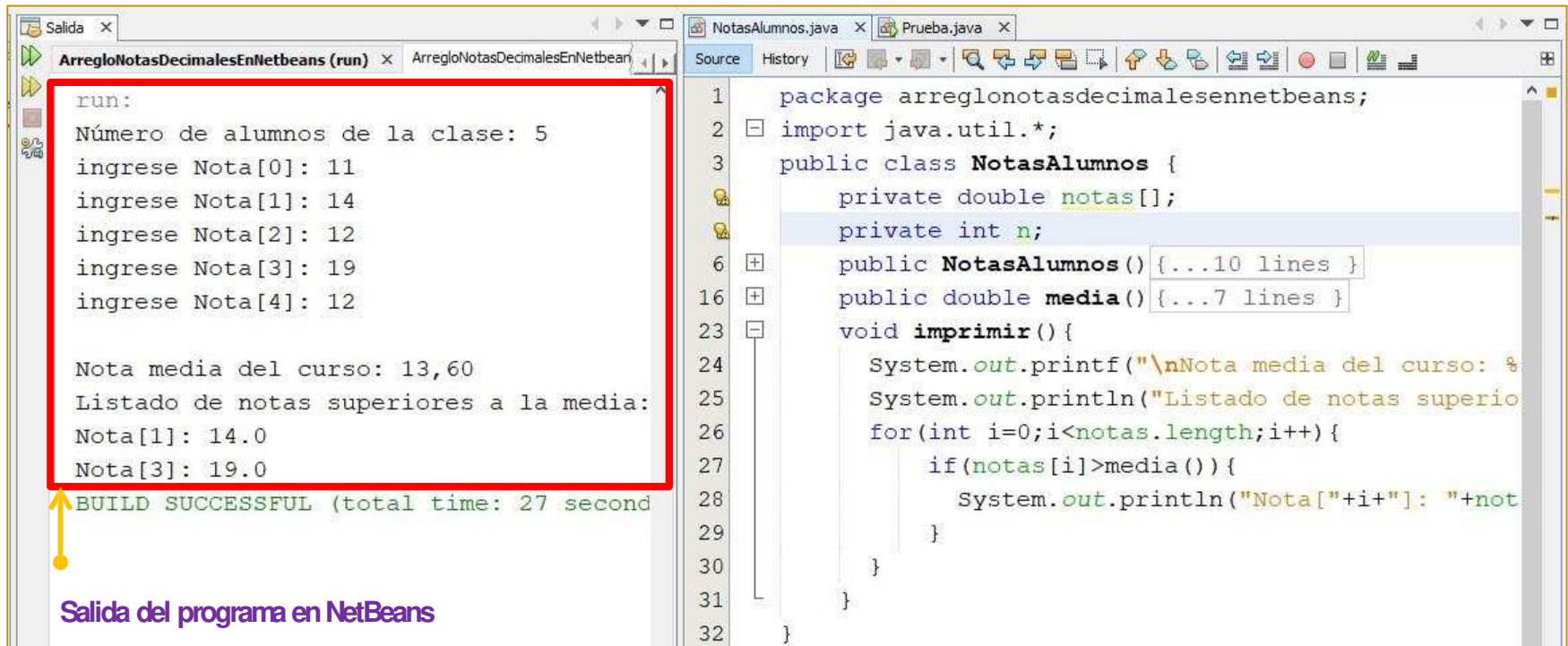
```
BlueJ: Ventana de Terminal - Arreglos de Notas Decimales en BlueJ
Opciones
Número de alumnos de la clase: 5
Ingrese Nota[0]: 11
Ingrese Nota[1]: 14
Ingrese Nota[2]: 12
Ingrese Nota[3]: 19
Ingrese Nota[4]: 12

Nota media del curso: 13.60
Listado de notas superiores a la media:
Nota[1]: 14.0
Nota[3]: 19.0

Can only enter input while your programming is running
```

Ejemplo 04 - Implementación en BlueJ y NetBeans

Desarrollar un programa que lea por teclado las notas de “n” alumnos de una clase, calcule la nota media del grupo, muestre las notas superiores a la media. El valor de “n” se lee por teclado.



The screenshot shows the NetBeans IDE with two panes. The left pane, titled 'Salida', displays the output of the program 'ArregloNotasDecimalesEnNetBeans (run)'. The right pane, titled 'Source', shows the source code of 'NotasAlumnos.java'.

Output (Left Pane):

```
run:
Número de alumnos de la clase: 5
ingrese Nota[0]: 11
ingrese Nota[1]: 14
ingrese Nota[2]: 12
ingrese Nota[3]: 19
ingrese Nota[4]: 12

Nota media del curso: 13,60
Listado de notas superiores a la media:
Nota[1]: 14.0
Nota[3]: 19.0
BUILD SUCCESSFUL (total time: 27 second)
```

Source Code (Right Pane):

```
1 package arreglonotasdecimalesennetbeans;
2 import java.util.*;
3 public class NotasAlumnos {
4     private double notas[];
5     private int n;
6     public NotasAlumnos() {...10 lines }
16    public double media() {...7 lines }
23    void imprimir() {
24        System.out.printf("\nNota media del curso: %
25        System.out.println("Listado de notas superio
26        for(int i=0;i<notas.length;i++){
27            if(notas[i]>media()){
28                System.out.println("Nota["+i+"]: "+not
29            }
30        }
31    }
32 }
```

Salida del programa en NetBeans

Cierre

¿Qué aprendimos hoy?

gracias!