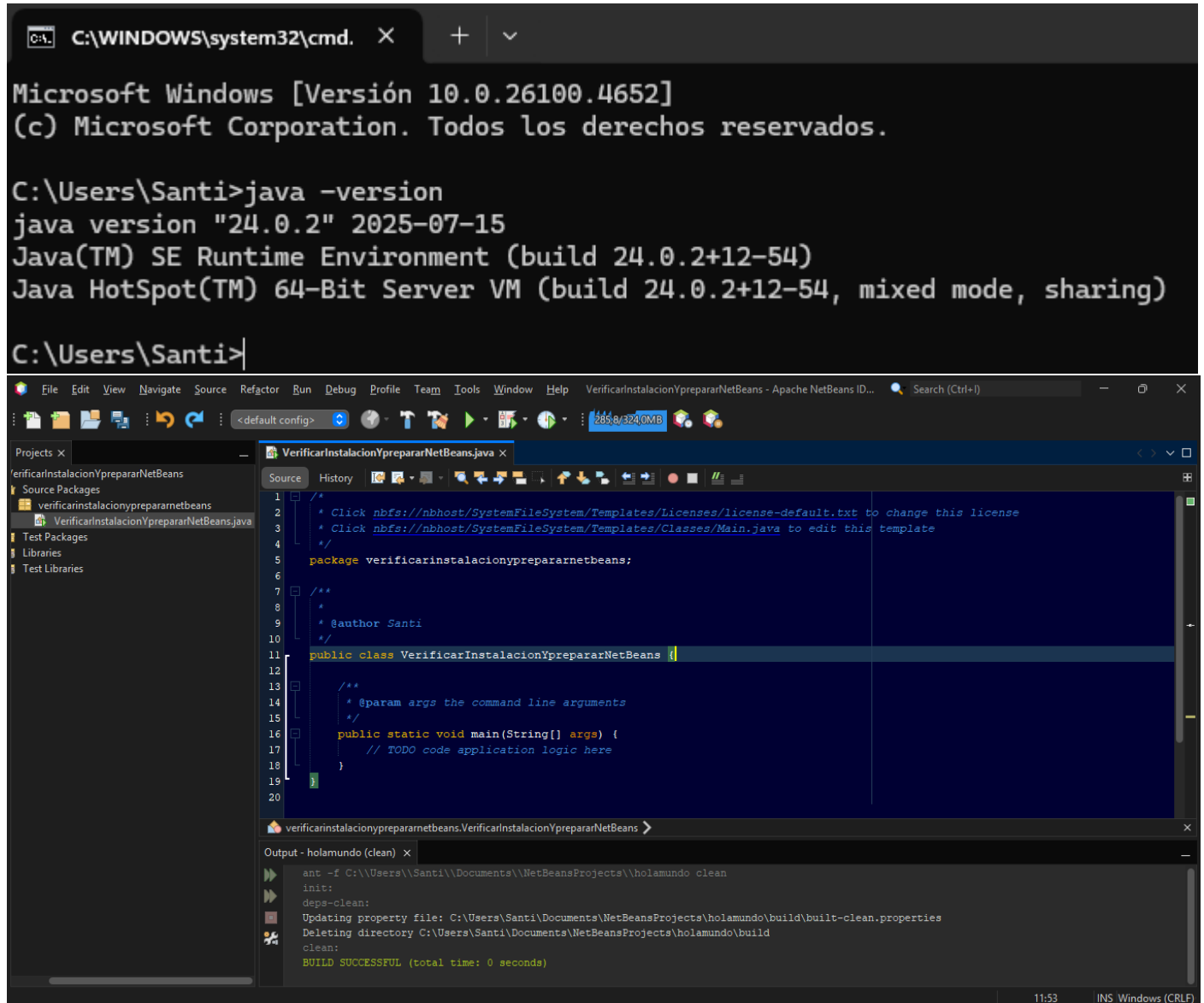


Programación II

TP – Introducción a Java

Santiago Nicolas Nievas

Ejercicio 1



The screenshot shows a Windows command prompt window with the following text:

```
C:\WINDOWS\system32\cmd. X + v
Microsoft Windows [Versión 10.0.26100.4652]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\Santi>java -version
java version "24.0.2" 2025-07-15
Java(TM) SE Runtime Environment (build 24.0.2+12-54)
Java HotSpot(TM) 64-Bit Server VM (build 24.0.2+12-54, mixed mode, sharing)

C:\Users\Santi>
```

Below the command prompt is an IDE window showing a file named `VerificarInstalacionYprepararNetBeans.java`. The code is as follows:

```
1  /*
2  * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
3  * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Main.java to edit this template
4  */
5  package verificarinstalacionyprepararnetbeans;
6
7  /**
8   *
9   * @author Santi
10  */
11  public class VerificarInstalacionYprepararNetBeans {
12
13      /**
14       * @param args the command line arguments
15       */
16      public static void main(String[] args) {
17          // TODO code application logic here
18      }
19  }
20
```

At the bottom of the IDE, the Output window shows the following message:

```
Output - holamundo (clean) X
ant -f C:\Users\Santi\Documents\NetBeansProjects\holamundo\clean
init:
deps-clean:
Updating property file: C:\Users\Santi\Documents\NetBeansProjects\holamundo\build\build-clean.properties
Deleting directory C:\Users\Santi\Documents\NetBeansProjects\holamundo\build
clean:
BUILD SUCCESSFUL (total time: 0 seconds)
```

Ejercicio 7

Código a analizar:

```
int x = 10;    // Línea 1
x = x + 5;    // Línea 2
System.out.println(x); // Línea 3
```

Instrucciones (sentencias): las tres líneas completas (1, 2 y 3) son instrucciones porque realizan una acción y terminan en ;.

Expresiones: 10 es una expresión literal; $x + 5$ es una **expresión** que se evalúa a un valor; el argumento de `println(x)` incluye la **expresión** x .

Diferencia clave: una **expresión** produce un **valor**; una **instrucción** realiza una **acción** (p. ej., asignar, imprimir).

Ejercicio 9

El **error** estaba en que el código original intentaba hacer:

```
String nombre = scanner.nextInt();
```

`nextInt()` devuelve un **valor numérico entero** (int), pero la variable donde se quería guardar (nombre) es de tipo **String**.

En Java no se puede asignar directamente un int a un String porque son tipos incompatibles → esto provoca un **error de compilación de tipos**.

Solución:

Usar **nextLine()** para leer texto y almacenarlo en una variable String.

Usar **nextInt()** solo para leer enteros y guardarlos en una variable int.

Código corregido:

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

```
String nombre = scanner.nextLine(); // ahora lee texto
```

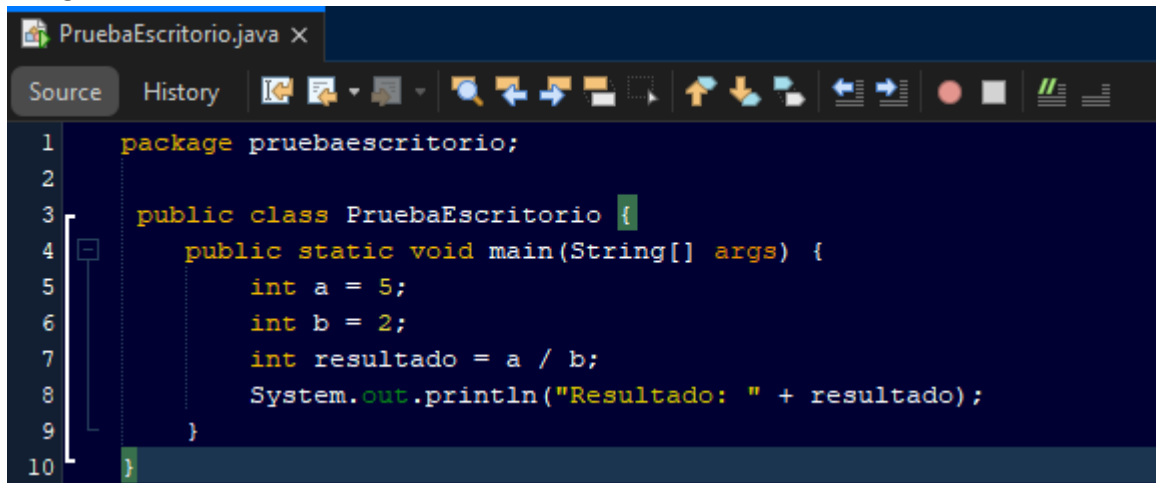
```
System.out.print("Ingresa tu edad: ");
```

```
int edad = scanner.nextInt();    // edad como número
```

De esta forma, cada variable recibe el tipo de dato que realmente puede contener, eliminando el error de compilación.

Ejercicio 10

Código



```
1 package pruebaescritorio;
2
3 public class PruebaEscritorio {
4     public static void main(String[] args) {
5         int a = 5;
6         int b = 2;
7         int resultado = a / b;
8         System.out.println("Resultado: " + resultado);
9     }
10 }
```

Prueba de escritorio (trazado paso a paso)

Paso	Instrucción	a	b	resultado	Salida en consola
1	int a = 5;	5	–	–	–
2	int b = 2;	5	2	–	–
3	int resultado = a / b;	5	2	2	–
4	System.out.println("Resultado: " + resultado);	5	2	2	Resultado: 2

Explicación

a / b es **división entera** porque **ambos son int**.

La división entera en Java **trunca** la parte decimal:

$5 / 2 = 2$ (no redondea, simplemente descarta el .5)

El valor 2 se almacena en resultado y luego se imprime.

Si quisieras que muestre el decimal (2.5) tendrías que hacer:

double resultado = (double) a / b; // o usar variables double

Esto forzaría una división en punto flotante.