



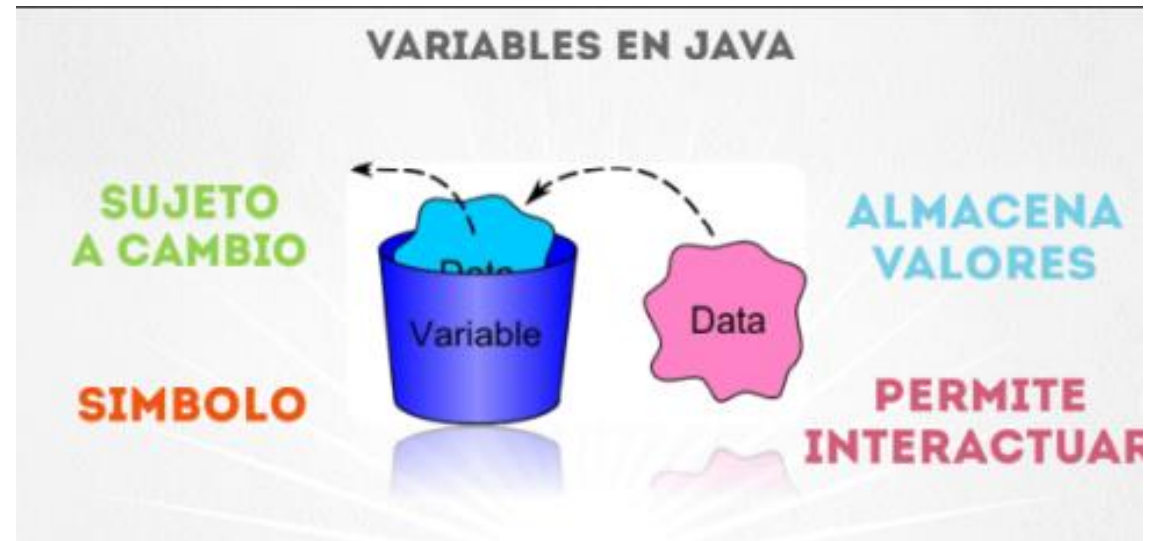
FUNDAMENTOS DE PROGRAMACIÓN

UNIDAD 2: CONCEPTOS BÁSICOS

VARIABLES



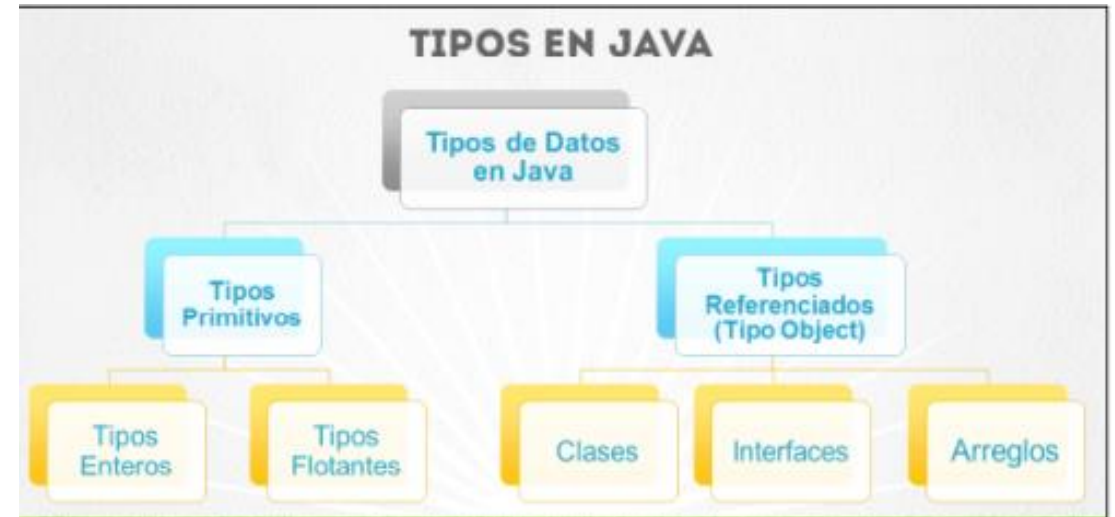
- En Java, como en cualquier lenguaje de programación, para almacenar información, es necesario la creación de variables, las cuales nos permitirán almacenar datos de nuestro programa de manera temporal.
- Cuando se declara una variable es reservar espacio en memoria dependiendo del tipo de variable
- Como se puede ver en la figura, tenemos datos que pueden ir cambiando a lo largo de la ejecución de nuestro programa.



VARIABLES - TIPOS DE DATOS



- En Java, existe una clasificación amplia respecto a los tipos que se manejan en java, sin embargo podemos resumirla en la figura mostrada.
- Tenemos tipos primitivos y los tipos referenciados (tipo Objeto) los cuales puede ser clases interfaces o arreglos.
- Empecemos por los primitivos..



VARIABLES – TIPOS PRIMITIVOS



- Los tipos primitivos tenemos los enteros, entre los cuales tenemos el tipo byte el cual ocupa 8 bits.
- Posteriormente tenemos el tipo short, el cual ocupa 16 bits.
- También tenemos el tipo char, el cual ocupa 16 bits pero maneja el código UNICODE para almacenar valores tipo char.
- A su vez tenemos el tipo int el cual ocupa 32 bits.
- finalmente el tipo long el cual ocupa 64 bits



Por otro lado tenemos los tipos flotantes, por un lado el tipo float el cual ocupa 32 bits, y el tipo double que ocupa 64 bits. El tipo boolean en Java también es un tipo primitivo y puede almacenar sólo el valor de true o false. Su valor por default es false

VARIABLES – TIPOS PRIMITIVOS



➤ En Java el manejo de cadenas es un tipo Object, pero que tiene varias características en particular.

➤ Ejemplo:

➤ `String saludo = "Hola Mundo";`



En Java, a diferencia de otros lenguajes, no se permite la sobrecarga de operadores, lo que significa que no podemos alterar la función de los operadores para hacer lo que nosotros deseemos, sin embargo existe una excepción a esta regla cuando manejamos tipos cadena. Cuando usamos el operador + y se detecta un tipo String en la operación, se dice que tenemos el contexto String, y por lo tanto en lugar de sumar valores, lo que hace Java es concatenar los valores que se encuentren en la operación

CARACTERES DE ESCAPE



➤ En la tabla mostrada, podemos observar los caracteres de escape que son utilizados al momento de desplegar mensajes en la salida estándar. Estos caracteres los pondremos en práctica en los ejercicios que vamos a realizar a continuación

CARACTERES DE ESCAPE AL UTILIZAR CADENAS

Secuencia de Escape	Descripción
\t	Inserta un tabulador
\b	Inserta un retroceso (backspace)
\n	Inserta una nueva línea
\r	Inserta un retorno de carro
\f	Se mueve a la siguiente página (Form feed). Se utiliza para impresoras, no en consolas.
\'	Inserta una comilla simple
\"	Inserta una comilla doble
\\	Inserta una barra invertida

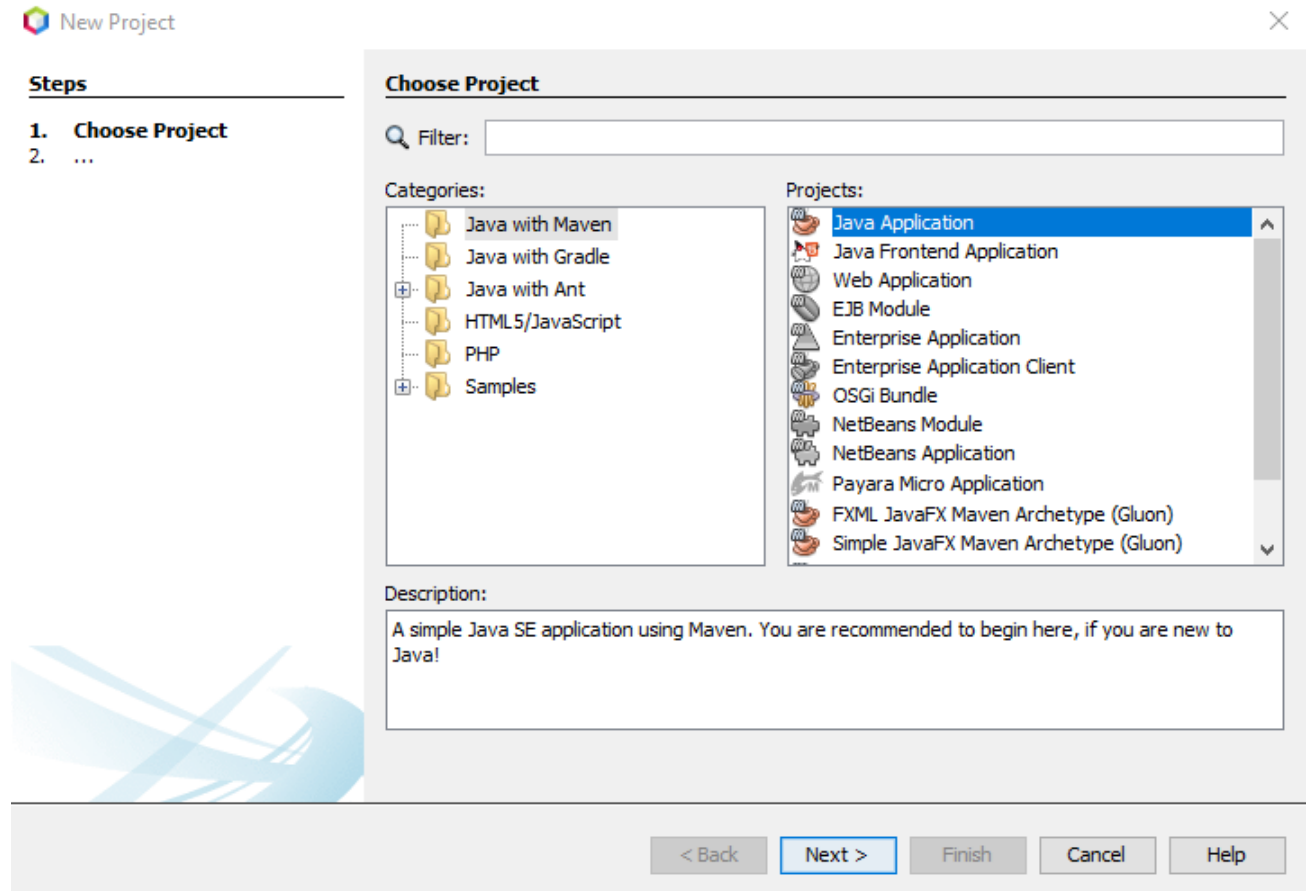
EJERCICIOS



➤ Crearemos un nuevo proyecto llamado variablesJava, desde nuestro ApacheNetBeans, recordemos se hace desde el menú File->new Project.

Elegimos “java with Maven -> java application”

Next



EJERCICIOS



- Nombre de proyecto “Variables”
- Group id: com.interlem
- Versión : 1
- Package: com.interlem.variables
- Finish

Name and Location

Project Name:

Project Location:

Project Folder:

Artifact Id:

Group Id:

Version:

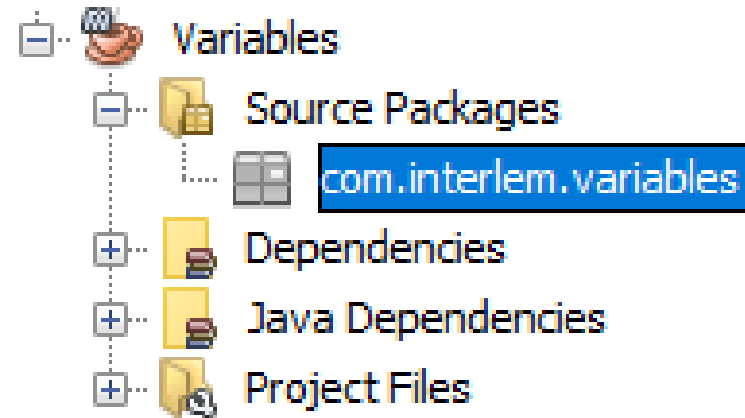
Package: (Optional)

< Back Next > **Finish** Cancel Help

EJERCICIOS



- Como se muestra en la figura, debe quedar creado nuestro proyecto.
- Con el nombre que le asignamos y el nombre del package que se le dio al momento de crear el proyecto.
- Se procede a dar clic derecho en el package “com.interlem.variables” y seleccionamos “nuevo-Java class”.



EJERCICIOS



- Limpiamos nuestro código, como se muestra en la figura.
- Para poder ejecutar nuestros proyectos en java, debemos crear el método “main”.
- Para eso escribimos:

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

```
}
```

- Todo lo que quede dentro de estas llaves se ejecutan en primera instancia.

```
Variable.java x
Source History
1 package com.interlem.variables;
2
3 public class Variable {
4
5 }
6
```

CREAR VARIABLES



➤ Para crear una variable en java debemos tener en cuenta las reglas; para ello.

1. se debe escribir el tipo de variable que vamos a usar. (ver tipos de variables).
2. declaramos el nombre que le queremos dar, utilizando el camelCase, donde la primera letra es minúscula.
3. Podemos asignar el dato que va a contener la variable con el operador de asignación “=”.
4. Luego escribimos el dato que va almacenar, debemos tener en cuenta las reglas explicadas en los tipos de variables con anterioridad.

```
package com.interlem.variables;

public class Variable {

    public static void main(String[] args) {
        int numero = 1;
        String nombre = "Edwin";
        double numeroGrande = 10;
    }
}
```

STRING



- El tipo de variable string aunque es de tipo object, almacena cadena de caracteres.
- Crearemos la el siguiente código.

```
public static void main(String[] args) {  
    String saludar = "saludos desde Java";  
}
```

- Para imprimir en la consola del IDE Apache Netbeans, utilizamos `system.out.println()`;
- El código completo quedaría como en la figura, seguido de eso ejecutamos nuestro programa.

```
package com.interlem.variables;  
  
public class Variable {  
  
    public static void main(String[] args) {  
        String saludar = "saludos desde Java";  
  
        System.out.println(saludar);  
    }  
}
```

```
--- exec-maven-plugin:1.5.0:exec (default-  
- saludos desde Java  
-----  
BUILD SUCCESS  
-----  
Total time: 1.612 s
```

SCANNER



- Con el tipo de variable String, podemos usar el operador “+” el cual va a concatenar los cadenas de caracteres, para ellos crear dos variables de tipo String.
- Colocan los datos que deseen, para mostrar en consola usamos el comando para imprimir por consola.
- `System.out.println();`
- Referenciamos “nombre” la cual es la primera variable, y adicionamos con comillas un espacio vacío, usando el operador “+”, nuevamente lo usamos para finalmente hacer uso de la variable “apellido”.

```
package com.interlem.variables;

public class Variable {

    public static void main(String[] args) {
        String nombre = "Pepito";
        String apellido = "Perez";

        System.out.println(nombre + " " + apellido);
    }
}
```

Pepito Perez

BUILD SUCCESS

Total time: 1.576 s

CARACTERES DE ESCAPE - EJERCICIO



- Tal cual se había explicado, java tiene caracteres de escape, que son utilizados al momento de desplegar mensajes en la salida estándar.
- Escribir el código que se muestra en la figura, e identifica que hace cada caracter de escape.

```
public class Variable {  
  
    public static void main(String[] args) {  
        String nombre = "Karla";  
        String apellido = "Esparza";  
        System.out.println(nombre + " " + apellido);  
        System.out.println("Nueva linea: \n" + nombre);  
        System.out.println("Tabulador: \t" + nombre);  
        System.out.println("Retroceso: \b\b" + nombre);  
        System.out.println("Retorno de carro: \r" + nombre);  
        System.out.println("Comilla simple: \' + nombre + '\");  
        System.out.println("Comilla doble: \" + nombre + \"\");  
        System.out.println("saludos");  
        System.out.print("adios");  
        System.out.println("nos vemos");  
    }  
}
```

SCANNER



- Para obtener información por teclado, desde la consola, podemos usar la función Scanner de java.

```
public static void main(String[] args) {  
    Scanner scanner = new Scanner(System.in);
```

- Declaramos una variable de nombre scanner del tipo de dato Scanner. (que se llamará un objeto scanner, lo veremos mas adelante).
- Creamos el objeto de tipo Scanner con el código “new Scanner(System.in)”

```
1 package com.interlem.variables;  
2  
3 public class Variable {  
4  
5     public static void main(String[] args) {  
6         Scanner scanner = new Scanner(System.in);  
7  
8  
9  
10  
11  
12  
13
```

- 💡 Add import for java.util.Scanner
- 💡 Create class "Scanner" in package com.interlem.variables (Source Packages)
- 💡 Create class "Scanner" with constructor "Scanner(java.io.InputStream)" in package com.interlem.variables (Source Packages)
- 💡 Create class "Scanner" in com.interlem.variables.Variable
- 💡 Create class "Scanner" in com.interlem.variables.Variable
- 💡 Search Dependency at Maven Repositories for Scanner

SCANNER



- Para que java identifique que es el tipo Scanner, se deben importar las librerías
- Java.util.Scanner, para que java reconozca esta nueva librería.
- Se debe crear la variable que va almacenar el dato ingresado por teclado. Todo aquel dato que ingrese por teclado siempre será de tipo String.
- Con la función “scanner.nextLine();”
- Estamos diciéndole a java que en esa variable usuario se almacene lo que se ingrese por teclado.
- Finalizamos imprimiendo y concatenando el resultado. Realice el ejercicio y cambie valores para seguir explorando.

```
package com.interlem.variables;  
  
import java.util.Scanner;  
  
public class Variable {  
  
    public static void main(String[] args) {  
        Scanner scanner = new Scanner(System.in);  
  
        System.out.println("proporciona el valor del usuario:");  
        String usuario = scanner.nextLine();  
        System.out.println("usuario = " + usuario);  
    }  
}
```


REFERENCIAS BIBLIOGRÁFICAS



GLOBAL MENTORING
www.globalmentoring.com.mx



GRACIAS

TODO LO PUEDO EN CRISTO QUE ME FORTALECE- Fil. 4.13