 **SEP SES TecNM**

**INSTITUTO TECNOLÓGICO DE TOLUCA**

**“GUÍA DE ESTUDIO U1-U3”**

**MATERIA:**

**FUNDAMENTOS DE PROGRAMACIÓN**

**CARRERA:**

**INGENIERÍA EN TECNOLOGÍAS DE LA INFORMACIÓN Y COMUNICACIONES**

**PRESENTA:**

**DIEGO ALEJANDRO REYES MEDINA**

**No. CONTROL:**

**23280703**

**DOCENTE:**

**ITZHAMARY VILCHIS GARCÍA**

**METEPEC, ESTADO DE MÉXICO, NOVIEMBRE DE 2023**

1.\_ Escribe 3 características de Java:

* Portabilidad
* Orientación a objetos
* Seguridad

2.\_ Escribe de 10 palabras reservadas:

1. boolean: Es un tipo de dato que representa valores lógicos, true (verdadero) o false (falso).
2. break: Se utiliza en bucles (for, while, switch) para salir de ellos antes de que se complete la iteración.
3. case: Se utiliza en las declaraciones switch para definir casos específicos que se ejecutan cuando el valor de la expresión coincide.
4. char: Es un tipo de dato que representa un solo carácter Unicode. Se declara utilizando comillas simples, como 'A'.
5. class: Se utiliza para declarar una clase en Java, que es una plantilla para crear objetos.
6. default: Se utiliza en declaraciones switch como un caso predeterminado que se ejecuta cuando ninguno de los casos coincide con el valor de la expresión.
7. double: Es un tipo de dato de punto flotante de doble precisión que se utiliza para representar números con decimales.
8. else: Se utiliza en una declaración if para especificar un bloque de código que se ejecuta si la condición no se cumple.
9. final: Se utiliza para indicar que un elemento (clase, método o variable) no puede ser modificado o extendido. Por ejemplo, una variable final es una constante.
10. public: Se utiliza para especificar que un miembro de una clase es accesible desde cualquier parte del código.

3.\_ Escribe los tipos de datos primitivos:

* byte: Representa números enteros de 8 bits con signo.
* short: Representa números enteros de 16 bits con signo.
* int: Representa números enteros de 32 bits con signo (el tipo de dato entero más común).
* long: Representa números enteros de 64 bits con signo.
* float: Representa números de punto flotante de precisión simple.
* double: Representa números de punto flotante de precisión doble (el tipo de dato flotante más común).
* char: Representa un carácter Unicode de 16 bits.
* boolean: Representa un valor verdadero (true) o falso (false).

4.\_ Escribe cómo funciona el método main:

public static void main(String[] args){

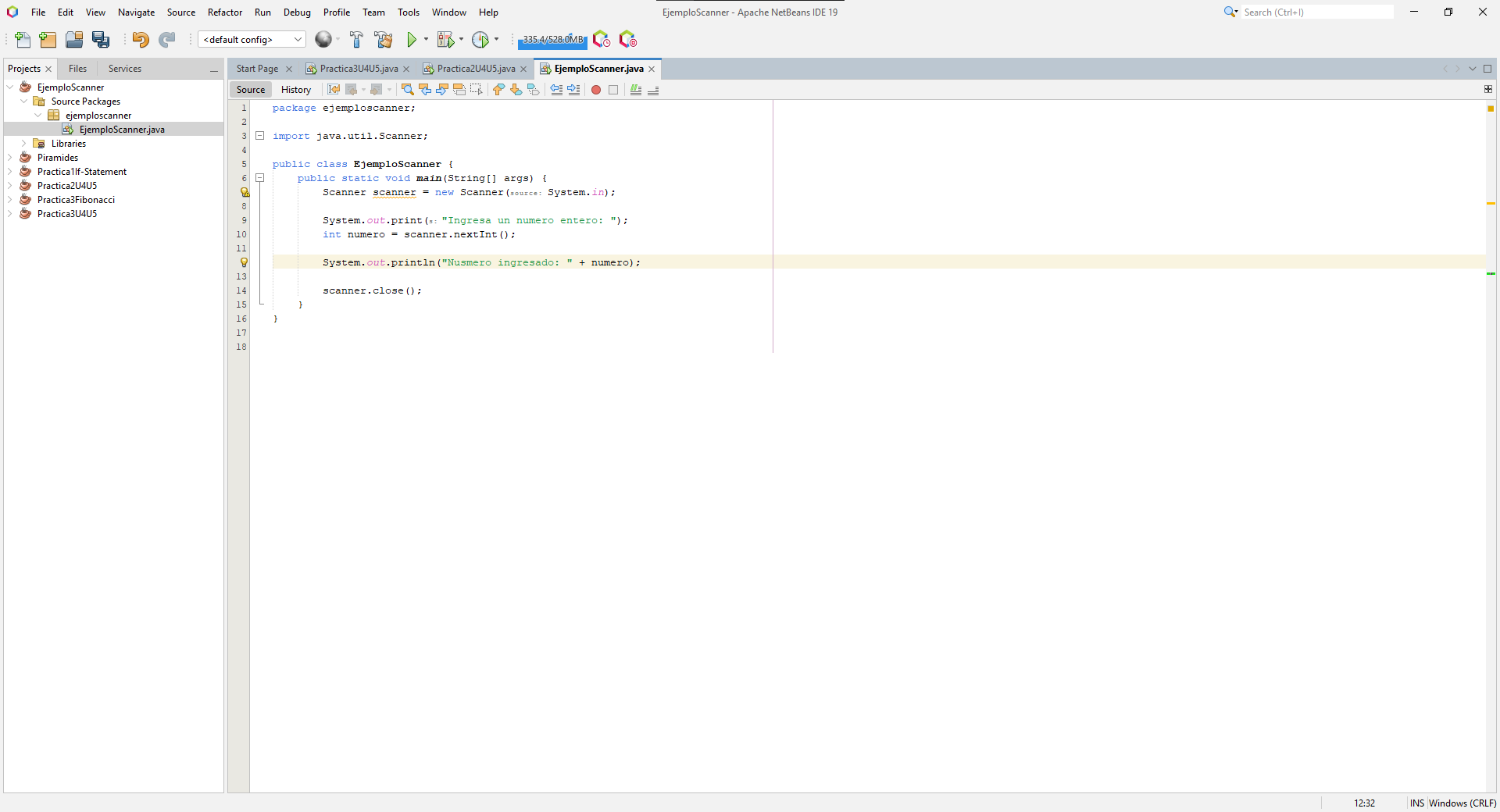
El método principal en Java es un método especial llamado main. Este método es el punto de entrada de cualquier programa Java y se utiliza para iniciar la ejecución del programa

Partes del método main:

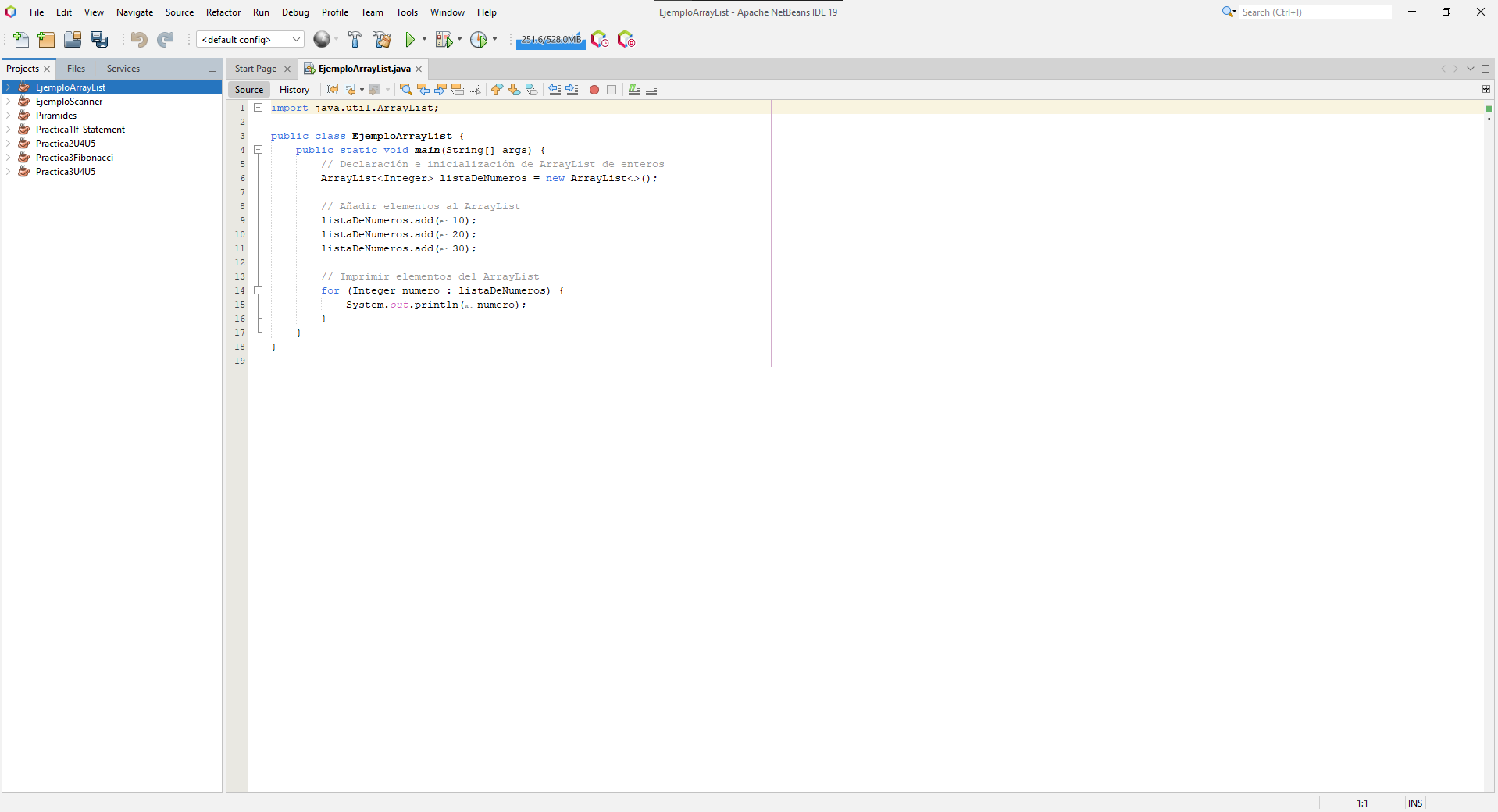
“public”: Indica que el método es accesible desde cualquier parte del programa.  
“static”: Indica que el método pertenece a la clase en lugar de a una instancia específica de la clase.  
“void”: No devuelve un valor.  
“main”: Nombre del método.  
“String[] args”: El método puede recibir argumentos como array de cadenas.

5.\_ Escribe 2 ejemplos de tipos de datos clase:

1.\_ Tipo de clase Scanner:  
La clase Scanner se encuentra en el paquete java.util y se utiliza para obtener la entrada del usuario desde la consola. Permite leer diferentes tipos de datos, como enteros, decimales y cadenas.



2.\_ Tipo de clase ArrayList:  
La clase ArrayList es parte del paquete java.util y se utiliza para almacenar elementos dinámicamente. A diferencia de los arrays tradicionales en Java, los ArrayList pueden cambiar de tamaño durante la ejecución del programa.



6.\_ Escribe dos operadores lógicos:

* Operador OR (||): El operador OR lógico devuelve true si al menos una de las expresiones booleanas es verdadera; devuelve false si ambas son falsas.
* Operador AND (&&): El operador AND lógico devuelve true si ambas expresiones booleanas son verdaderas; de lo contrario, devuelve false.

7.\_ Escribe dos operadores aritméticos:

* Operador de Suma (+): El operador de suma se utiliza para sumar dos valores.
* Operador de Multiplicación: El operador de multiplicación se utiliza para multiplicar dos valores.

8.\_ Escribe dos operadores de comparación:

* Operador de Mayor que (>): El operador de mayor que compara dos valores y devuelve true si el valor de la izquierda es mayor que el de la derecha; de lo contrario, devuelve false
* Operador de igualdad (==): El operador de igualdad compara dos valores y devuelve true si son iguales, y false si son diferentes.

9.\_ Escribe dos operadores de incremento/decremento:

* Operador de Decremento (--): El operador de decremento disminuye el valor de la variable en una unidad. Al igual que el operador de incremento, puede aplicarse como prefijo (--variable) o sufijo (variable--).
* Operador de Incremento (++): El operador de incremento aumenta el valor de la variable en una unidad. Puede aplicarse tanto como prefijo (++variable) o sufijo (variable++).

10.\_ Escribe para qué sirve el operador +=:

El operador += en Java se utiliza para realizar una operación de adición y asignación al mismo tiempo. Es una forma abreviada de escribir una expresión que suma el valor de la variable actual con otro valor y luego asigna el resultado de esa suma de nuevo a la variable original.

11.\_ Escribe una variable por cada tipo de dato primitivo (considera String):

* String nombre;
* int numeroEntero;
* long numLargo;
* double numeroDecimal;
* short numShort;
* float numFlotante;
* byte numeroByte;
* char caracter;
* boolean esVerdadero;

12.\_ Escribe una variable inicializada por cada tipo de dato primitivo (considera String):

* String nombre = Diego;
* int numeroEntero = 48;
* long numLargo = 987654321M;
* double numeroDecimal = 4.8671;
* short numShort = 23456;
* float numFlotante = 3.34567j;
* byte numeroByte = 118;
* char caracter = “B”;
* Boolean esVerdadero = true;

13.\_ Escribe una constante por cada tipo de dato primitivo (considera String):

* final String ConstanteNombre = “Diego”;
* final int ConstanteEntera = 48;
* final long ConstanteLarga = 987654321M;
* final double ConstanteDecimal = 4.8671;
* final short ConstanteShort = 23456;
* final float ConstanteFlotante = 3.34567j;
* final byte ConstanteByte = 118;
* final char ConstanteChar = “B”;
* final Boolean ConstanteBooleana = true;

14.\_ Escribe el código para pedirle al usuario su ficha de contacto (Nombre completo, edad, correo electrónico, teléfono) e imprímelo:

15. Escribe el código para obtener las tablas de multiplicar pidiendo al usuario la tabla del número que desea ver:

16. Escribe el código para obtener el área del círculo. Pide al usuario el radio:

17.\_ Escribe para que sirve if–else:

Es una declaración de control de flujo que se utiliza para tomar decisiones en función de una condición. Permite que un programa ejecute cierto bloque de código si una condición es verdadera y otro bloque de código si la condición es falsa.

Partes del if–else:

if: Es la palabra clave que inicia la estructura de control de flujo. Indica que se evaluará una condición.

else: Se utiliza después del bloque de código asociado al if. Indica que se ejecutará un bloque de código si la condición en el if es falsa.

18.\_ Escribe para qué sirve switch:

Es una estructura de control de flujo que se utiliza para tomar decisiones basadas en el valor de una expresión. Es una alternativa a la declaración if-else cuando se desea realizar múltiples comparaciones de igualdad sobre el mismo valor.

19.\_ Escribe cuál es la diferencia entre while y do-while:

El bucle while verifica la condición antes de ejecutar el bloque de código, mientras que el bucle do-while ejecuta el bloque de código al menos una vez antes de verificar la condición.

while: En un bucle while, la condición se evalúa antes de que se ejecute el bloque de código asociado al bucle.

do-while: En un bucle do-while, la condición se evalúa después de que se ha ejecutado al menos una vez el bloque de código asociado al bucle.

20.\_ Escribe la estructura de un for y explica sus partes:

for(inicialización;condición;incremento) {  
}

Inicialización: Se ejecuta una sola vez, aquí se inicializa el contador, por lo regular se utilizan las letras i, j y k.

Condición: Es la segunda parte del ciclo “for” y se evalúa antes de cada iteración. Si la condición es verdadera, el ciclo continuará ejecutándose; de lo contrario, se sale del ciclo.

Incremento: Utilizado para modificar las variables que se declararon en la inicialización y, generalmente, se utiliza para incrementar o decrementar el contador.

21.\_ Escribe el código de un menú de una taquería que se repita hasta que el usuario teclee la opción de salir:

22.\_ Escribe el código para imprimir el factorial de un número:

23.\_ Escribe el código donde le pidas al usuario un número e imprimas si es positivo o negativo:

24.\_ Escribe el código para imprimir los números primos entre el 2 y el 500:

25.\_ Escribe el código para obtener los días de la semana de la siguiente manera: Cada día es una opción diferente, le vas a preguntar al usuario un número del 1 al 7 y de acuerdo al número ingresado vas a imprimir “Hola soy el día” + día:

Los ejercicios 14, 15, 16, 21, 22, 23, 24 y 25 fueron realizados y subidos a la plataforma de GitHub, los códigos de dichos ejercicios fueron plasmados en un repositorio especial para la guía.

Link del repositorio:  
<https://github.com/ITzDielex/GuiaU1-U3>

Link del documento de las impresiones de los trabajos del repositorio de GitHub:

<https://github.com/ITzDielex/GuiaU1-U3/blob/main/ImpresionesEjerciciosGuía.docx>