Universidad de las Americas

UNIVERIDAD DE LAS AMÉRICAS

CAMPUS: UDLAPARK

CARRERA: INGENIERÍA EN SOFTWARE

ASIGNATURA: PROGRAMACION III

MODALIDAD: PRESENCIAL

AUTOR/ES:
Jean Carlos Gómez Mafla

MARZO 2025 – AGOSTO 2025

QUITO – ECUADOR

Universidad de las Americas

Temas de Programación en Java – Semana 1

1. Uso de la Clase String en Java

La clase **String** en Java se utiliza para representar **cadenas de texto** en lo cual es una clase inmutable (lo que significa que una vez creado un objeto **String** su contenido no puede ser cambiado).

Ejemplo de declaración:

```
String saludo = "Hola mundo";
```

Métodos más importantes:

Método	Descripción	Ejemplo
length()	Devuelve la longitud del string	"Hola".length() → 4
charAt(int index)	Devuelve el carácter en la posición dada	"Hola".charAt(1) → 'o'
substring(int start, int end)	Devuelve un substring entre índices	"Hola".substring(1, 3) → "ol"
indexOf(String str)	Devuelve el índice de la primera ocurrencia	"Hola".indexOf("l") → 2
equals(String another)	Compara si dos strings son iguales	"Hola".equals("hola") → false
equalsIgnoreCase(String)	Compara ignorando mayúsculas/minúsculas	"Hola".equalsIgnoreCase("hola") → true
toUpperCase()	Convierte el string a mayúsculas	"hola".toUpperCase() → "HOLA"
toLowerCase()	Convierte a minúsculas	"HOLA".toLowerCase() → "hola"
replace(char old, char new)	Reemplaza caracteres	"casa".replace('a', 'e') → "cese"
trim()	Elimina espacios en blanco al inicio y final	" hola ".trim() → "hola"

2. Excepciones en Java: estructura y manejo

Las **excepciones** son eventos que interrumpen el flujo normal de un programa.

Java maneja las excepciones las siguientes palabras reservadas (mediante bloques)

try, **catch**, **finally**.

Estructura General:

```
try {
} catch (TipoDeExcepcion e) {
} finally {
}
```

Ejemplo:

```
try {
   int division = 10 / 0;
} catch (ArithmeticException e) {
   System.out.println("Error: División por cero.");
} finally {
   System.out.println("Este bloque siempre se ejecuta.");
}
```

Jerarquía de excepciones (simplificada):

1. Throwable
2. Error
3. Exception
4. RuntimeException
5. Otros (IOException, SQLException)

3. Tipos de Datos Abstractos (TDA)

Los tipos de dato abstracto (**TDA**) es una estructura que define operaciones de un conjunto de datos, sin especificar cómo se implementan.

Ejemplos comunes de **TDAs**:

TDA	Descripción	Operaciones comunes
Lista	Colección ordenada de elementos	insertar, eliminar, buscar
Pila (Stack)	Último en entrar, primero en salir (LIFO)	push, pop, peek
Cola (Queue)	Primero en entrar, primero en salir (FIFO)	enqueue, dequeue
Árbol	Estructura jerárquica de nodos	insertar, recorrer, buscar
Grafo	Conjunto de nodos conectados por aristas	agregarNodo, agregarArista, buscarRuta

Ventajas de usar **TDAs**:

- Permiten enfocarse en qué hace la estructura, no en cómo lo hace.
- Fomentan la reutilización de código.
- Ayudan a resolver problemas complejos de forma modular y organizada.