



ESCUELA DE
INGENIERÍA EN CIENCIAS Y SISTEMAS
FACULTAD DE INGENIERÍA
UNIVERSIDAD DE SAN CARLOS DE GUATEMALA



Día, Fecha:

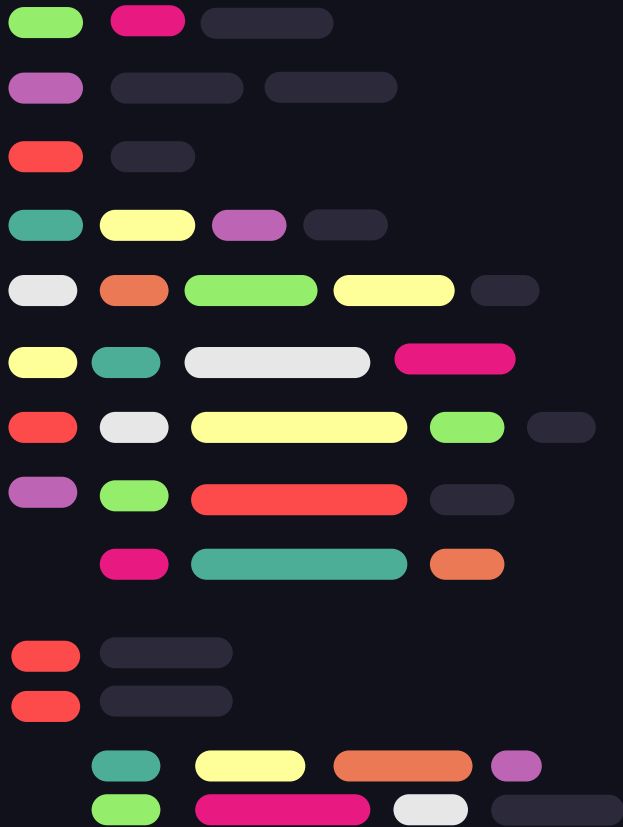
Viernes, 23/08/2024

Hora de inicio:

09:00

Introducción a la Programación y Computación 1 [B]

Josué Rodolfo Morales Castillo



{ ..



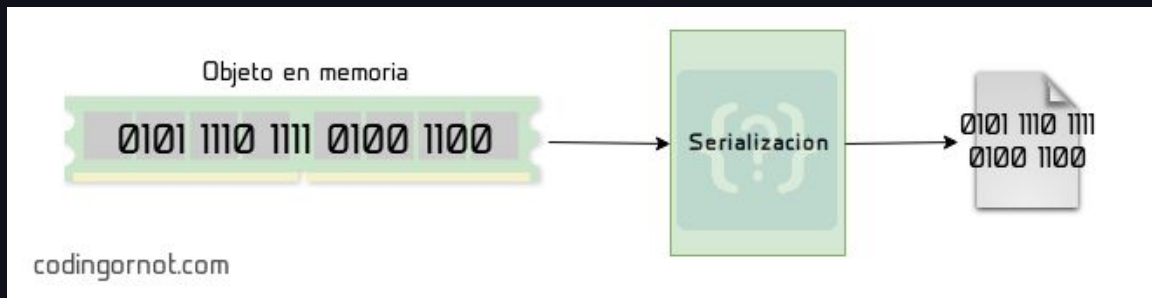
Clase 5- Agenda

- Foro No. 5
- Preguntas Proyecto 1
- Recordatorio Modalidad Calificación
- Manejo de Archivos, Manejo de Errores y Serialización

} ..

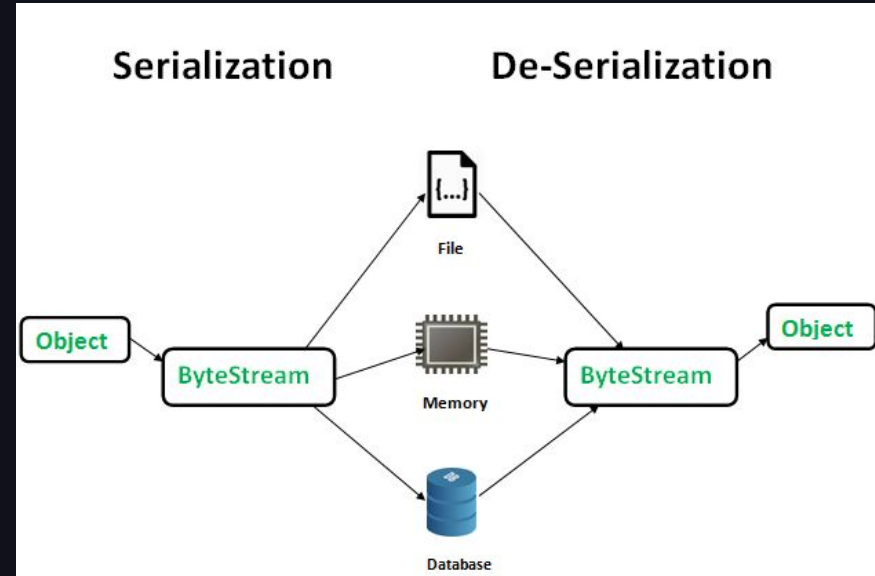
Serialización de objetos

La serialización es el proceso de convertir un objeto en una secuencia de bytes que puede ser almacenada, transmitida o guardada en un archivo. Para que un objeto sea serializable, su clase debe implementar la interfaz **Serializable**. Una vez que un objeto es serializado, puede ser guardado o transmitido utilizando clases como **ObjectOutputStream**.



Deserialización de objetos

Es el proceso inverso de la serialización. Convierte una secuencia de bytes (que representa un objeto serializado) de vuelta en un objeto. Esto se hace principalmente para reconstruir objetos previamente serializados, por ejemplo, cuando se lee un objeto de un archivo. Para deserializar un objeto, se utiliza la clase `ObjectInputStream`.





Manejo de errores

El manejo de errores se realiza mediante el uso de excepciones. Una excepción es un evento que ocurre durante la ejecución de un programa y que interrumpe el flujo normal de las instrucciones. El manejo de errores en Java se implementa mediante la captura y manejo de excepciones.

```
try
{

}
catch (ExceptionType name)
{

}
catch (ExceptionType name)
{

}
finally
{

}
```

Arithmetic Exception

Esta excepción se produce cuando ocurre un error aritmético, como la división por cero (/0). También puede ocurrir cuando se realiza una operación aritmética inválida en un tipo numérico, como la operación de módulo (%) con divisor cero.





```
try {  
    double division = 21 / 0;  
} catch (ArithmeticException ae) {  
    System.out.println("Error: " + ae.getMessage());  
}
```

```
run:  
Error: / by zero  
BUILD SUCCESSFUL (total time: 1 second)  
|
```

Null Pointer Exception

`NullPointerException` es una excepción que indica que la aplicación intentó usar una referencia a un objeto que tenía un valor nulo.

```
try {  
    String str = null;  
    int length = str.length();  
} catch (NullPointerException ne) {  
    System.out.println("Error: " + ne.getMessage());  
}
```






```
run:  
Error: Cannot invoke "String.length()" because "str" is null  
BUILD SUCCESSFUL (total time: 0 seconds)  
|
```

Array Index Out Of Bounds Exception

Esta excepción ocurre cuando se intenta acceder a un índice fuera del rango válido de un array. Puede ocurrir al intentar acceder a un índice negativo o a un índice mayor que la longitud del array.

```
try {  
    int[] myNumbers = {1, 2, 3};  
    System.out.println(myNumbers[10]);  
} catch (ArrayIndexOutOfBoundsException aioobe) {  
    System.out.println("Something went wrong: " + aioobe.getMessage());  
}
```



```
run:  
Something went wrong: Index 10 out of bounds for length 3  
BUILD SUCCESSFUL (total time: 0 seconds)
```


Class Cast Exception

Esta excepción se lanza cuando se intenta realizar una conversión de tipo (casting) entre dos tipos de objetos que no están relacionados por herencia. Ocurre cuando se trata de convertir un objeto a un tipo que no es compatible con su tipo real.

```
try {  
    Object obj = "Hola";  
    Integer num = (Integer) obj;  
} catch (ClassCastException ce) {  
    System.out.println("Error: " + ce.getMessage());  
}
```



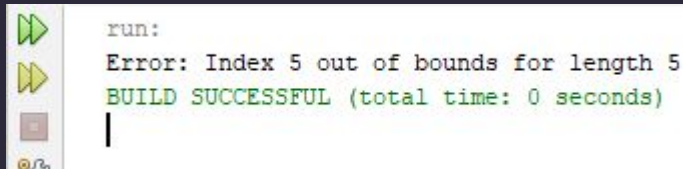
run:

Error: class java.lang.String cannot be cast to class java.lang.Integer (java.lang.String and java.lang.Integer are in module java.base of loader 'bootstrap')
BUILD SUCCESSFUL (total time: 0 seconds)

Exception

Es la clase base para todas las excepciones en Java. Cuando se captura Exception, se está manejando cualquier excepción que sea subclase directa o indirecta de Exception. Sin embargo, capturar Exception de manera general no es siempre la mejor práctica, ya que podría ocultar problemas específicos y hacer que el código sea menos mantenible.

```
try {  
    int[] arr = new int[5];  
    System.out.println(arr[5]);  
} catch (Exception e) {  
    System.out.println("Error: " + e.getMessage());  
}
```



The screenshot shows an IDE interface with a left sidebar containing icons for Run, Debug, and Test. The main area displays the output of a program execution. It starts with 'run:' followed by an error message 'Error: Index 5 out of bounds for length 5' in red text. Below the error, it says 'BUILD SUCCESSFUL (total time: 0 seconds)' in green text. A vertical cursor is visible at the end of the output line.

```
run:  
Error: Index 5 out of bounds for length 5  
BUILD SUCCESSFUL (total time: 0 seconds)  
|
```

¿Bug?



(Insecto en inglés) se refiere a un error, fallo o defecto en el código de un programa que provoca un comportamiento no deseado o incorrecto.

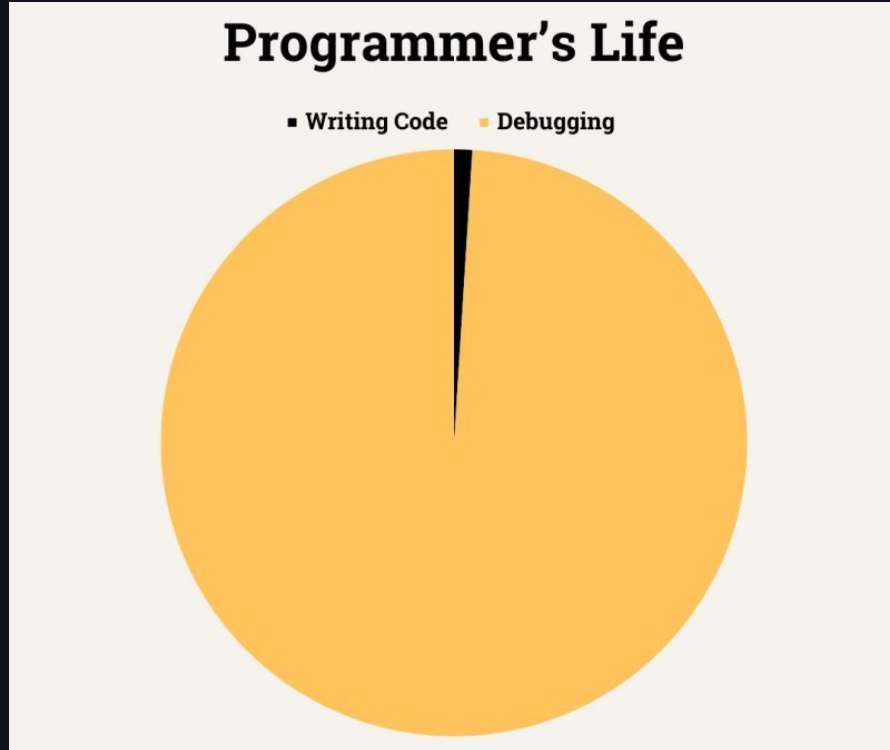


Proceso de identificar, analizar y corregir errores o bugs en el código de un programa.



Debugging

Vida de un Programador



¿Dudas?

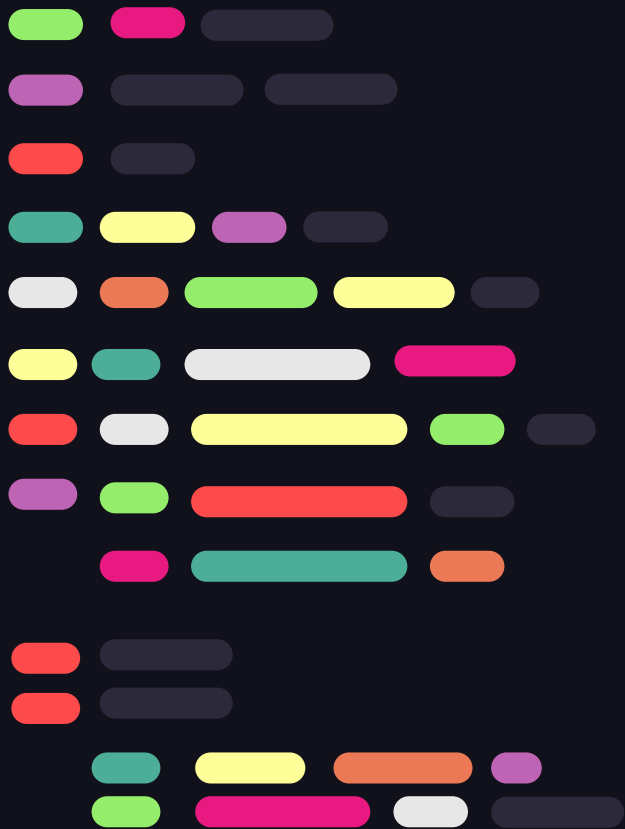


{



}





{ ..



Ejemplo

} ..