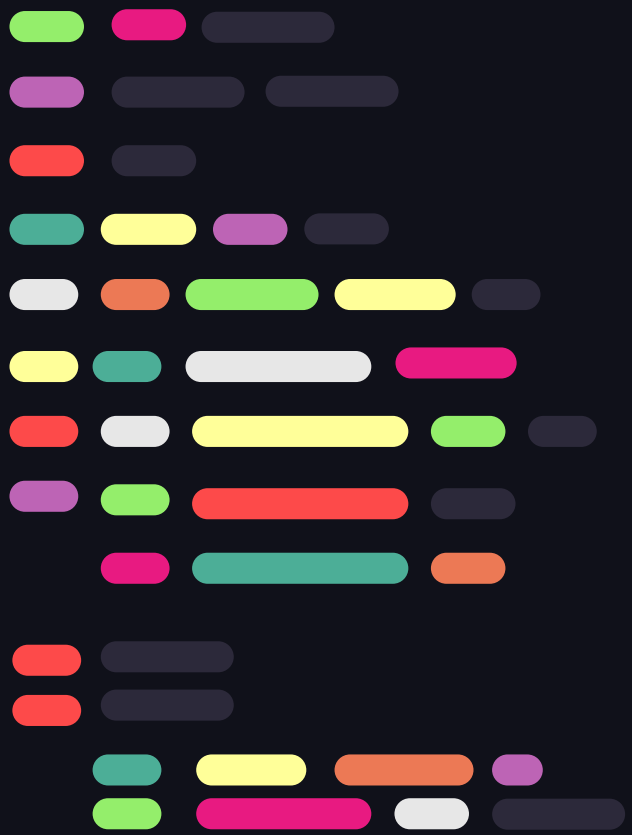
ESCUELA DE
INGENIERÍA EN CIENCIAS Y SISTEMAS
FACULTAD DE INGENIERÍA
UNIVERSIDAD DE SAN CARLOS DE GUATEMALA

ECYS | DIRT
DESARROLLO DE TRANSFERENCIA TECNOLÓGICA

Día, Fecha:	Jueves, 14/03/2024
Hora de inicio:	17:20

Introducción a la Programación y Computación 1 [B]

Josué Rodolfo Morales Castillo



Clase 8- Agenda

- Foro No. 8
- Preguntas Practica 2
- Modificación Practica 2 (Botón reanudar)
- Horarios de Calificación
- Manejo de Archivos y Serialización

Archivos de texto plano

Los archivos de texto plano son una forma fundamental de almacenar y manipular datos en el desarrollo de software. En Java, trabajar con archivos de texto es una tarea común y esencial para una amplia gama de aplicaciones.

Creación de Archivos (FileWriter)

```java
import java.io.FileWriter;
import java.io.IOException;

public class EscribirArchivo {
    public static void main(String[] args) {
        try {
            FileWriter archivo = new FileWriter("archivo.txt");
            archivo.write("¡Hola, mundo!");
            archivo.close();
            System.out.println("Texto escrito correctamente.");
        } catch (IOException e) {
            System.out.println("Error al escribir en el archivo.");
            e.printStackTrace();
        }
    }
}
```

Lectura de Archivos(FileReader)

```java
import java.io.FileReader;
import java.io.IOException;

public class LeerArchivo {
    public static void main(String[] args) {
        try {
            FileReader archivo = new FileReader("archivo.txt");
            int caracter;
            while ((caracter = archivo.read()) != -1) {
                System.out.print((char) caracter);
            }
            archivo.close();
        } catch (IOException e) {
            System.out.println("Error al leer el archivo.");
            e.printStackTrace();
        }
    }
}
```

Eliminación de Archivos(delete)

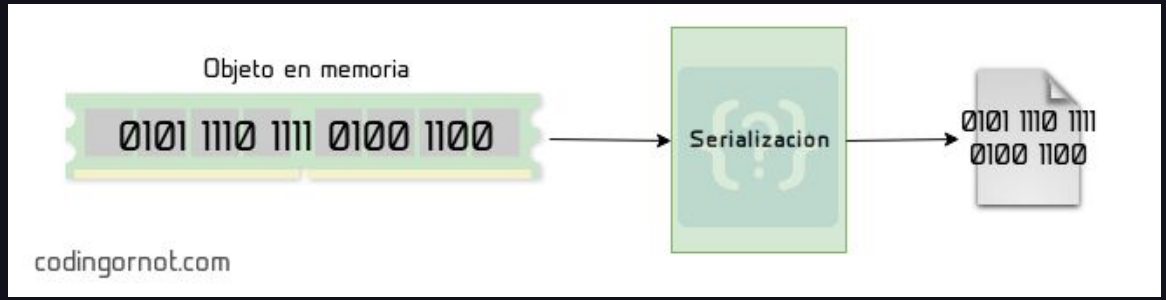
```java
import java.io.File;

public class EliminarArchivo {
    public static void main(String[] args) {
        File archivo = new File("archivo.txt");
        if (archivo.delete()) {
            System.out.println("Archivo eliminado correctamente.");
        } else {
            System.out.println("Error al eliminar el archivo.");
        }
    }
}
```

Modificar archivo

```java
String nombreArchivo = "archivo.txt";

try {
    // Abrir el archivo en modo de lectura y escritura
    FileWriter fileWriter = new FileWriter(nombreArchivo, true);
    BufferedWriter bufferedWriter = new BufferedWriter(fileWriter);
    PrintWriter printWriter = new PrintWriter(bufferedWriter);

    // Modificar el contenido del archivo
    printWriter.println("Nueva línea agregada al archivo.");

    // Cerrar el archivo
    printWriter.close();

    System.out.println("El archivo se ha modificado correctamente.");
} catch (IOException e) {
    System.out.println("Ocurrió un error al modificar el archivo: " + e.getMessage());
}
```

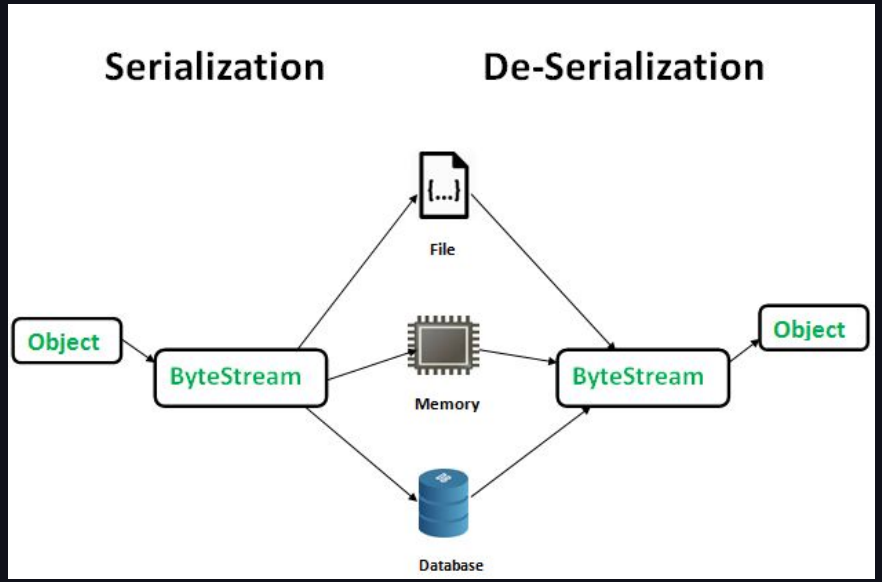
Serialización de objetos

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.



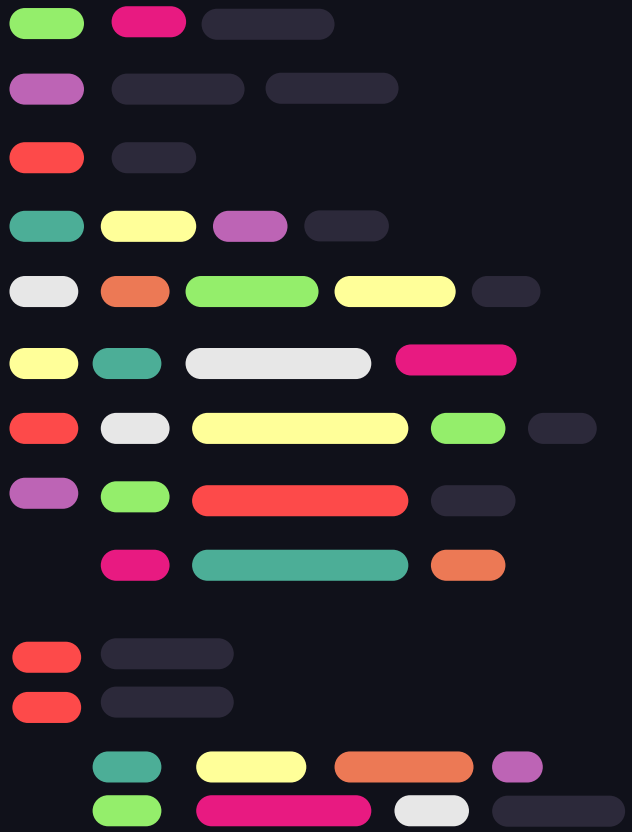
Deserializaciónde objetos

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.



¿Dudas?


GRACIAS POR SU ATENCION,SI TIENEN DUDAS...
CONSULTEN GOOGLE
memegenerator.es



{ ..

Ejemplo

} ..