

1 Estructura del Proyecto

1.1 UML

1.2 la clase 'Main'

La principal función de la clase 'Main', además de correr el programa, es unir al resto del programa para su funcionamiento óptimo, en 'Main.java' esta definida la estructura que las clases que vamos a manejar para solucionar el problema.

Una de las primeras cosas que van a saltar a la vista es el uso de constantes definidas como 'final' dentro de la lógica de java. con esto, pretendemos hacer que las variables acá definidas sean inmutables y no puedan ser cambiadas ni por el usuario ni por ninguno de los procesos que el programa corra una vez definidas. Las variables que caen dentro de este conjunto inmutable son

1. MESSAGE NUM: Define el número de mensajes a enviar.
2. PROCESS NUM: Define el número de procesos del programa.
3. BUFFER CAP: Define el limite del Buffer
4. BUFFER NUM: Define el número de Buffers
5. STAGE NUM: Define el número de etapas por las que pasarán los procesos

Estas conses son luego utilizadas para la creación de los objetos que serán usados en el programa, y conectará la lógica de esos nuevos objetos creados.

1.3 Estructura de los Procesos

1.4 Estructura de Input

1.5 Estructura de Prints

2 Pruebas del programa

2.1 Estructura y requerimientos de prueba.

Este programa fue probado en el IDE integrado IntelliJ IDEA 2022.3.2, con el JDK Java 11 y en la consola integrada del IDE. Fue probado en Windows 10 2H22 y Manjaro Linux 22.10, la estructura del proyecto esta planteada con el sistema de Build 'Maven' y es dependiente principalmente de las librerías estandar de Java. Sin embargo, para conseguir los prints con símbolos integrados, la fuente de la consola en la cuál se compile debe ser compatible con simbolos Unicode, de lo contrario, solo será mostrado el código Unicode referente a los símbolos.

2.2 Pruebas

2.2.1 Input

2.2.2 Output

2.2.3 Input

2.2.4 Output

2.2.5 Input

2.2.6 Output

3 Glosario

Función Lambda

Objeto anónimo