**Prueba técnica Refactor**

Solición.

1. Parte conceptual.

Allasgoz:

1. Las varaible output de los metodos getContent() y getContentWithoutUnicode() se deberia crear con un StringBuilder o StringBuffer según sea el caso ya que por cada iteracion cuando se concatena se crear un nuevo espacio en memoria y por ende el aputador de la variable cambia a ese nuevo espacio y eso conlleva a un aumento de la memoria.

No obstante recomiendo utilizar StringBuffer ya que es mutable, lo que significa que se puede cambiar el valor del objeto, anexo a esto todos los métodos del StringBuffer son sicronizados lo que es recomendable cuando se usa con hilos ya que esta protegido, es decir es threas safe.

StringBuilder no es seguro a la hora de ser usado en en hilos no es thread safe.

1. Los metodos setFile() y getFile() no funcionan como deberian funcionar a parte de eso no hace que la lectura y escritura de archivo sea thread safe.
2. Para que la lectura y escritura sea thread safe se deberia sincroniza tanto la ruta del archivo como la propia funcion de lectura y escritura de archivo, esto evita que mientras se este escribiendo llegue otro hilo a escribir, esto previene tener comportamientos inesperados en la aplicación.
3. Refactor de código

Las Clases refactorizadas se adjunta en el correo y estan disponibles en el siguiente enlace git:

<https://github.com/romergi1/ProyectosGER/tree/master/Proceso%20Lider%20Te%CC%81cnico%20Masiv>

Se realizó refactor con dos clases.

1. Main.java
2. Printer.java

Solución prueba técnica Clan Code.

Esta prueba fue desarrolla sobre AWS, en arquitectura serverless y sobren un runtime NODE.js

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Se creó el API:

<https://e6gr5zk027.execute-api.us-east-1.amazonaws.com/dev/>

Con los respectivos recursos de acuerdo a las necesidades del ejercicio.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

1. El primer recurso , corresponde a generar la ruleta ,devolviendo el id correspondiente.

Curl

curl --location --request POST 'https://e6gr5zk027.execute-api.us-east-1.amazonaws.com/dev/setoperationnewrul' \

--header 'Content-Type: application/json' \

--data-raw '{

"viOperation": "0",

"viIdRo": "0",

"viTypeElec": "0",

"viIdenUs": "0",

"viValueTypeElec": "0",

"viValueQuantity": 0

}'

1. El segúndo recurdso, obtiene el estado de la ruleta 1= activo 0= inactiva.

curl --location --request POST 'https://e6gr5zk027.execute-api.us-east-1.amazonaws.com/dev/getoperationgetrulbyid' \

--header 'Content-Type: application/json' \

--data-raw '{

"viOperation": "3",

"viIdRo": "2",

"viTypeElec": "1",

"viIdenUs": "74770935",

"viValueTypeElec": "30",

"viValueQuantity": 7000

}'

1. El tercer recurso, genera la participación del usuario de acuerdo a las siguientes variables.

{

"viOperation": "3",

"viIdRo": "2",

"viTypeElec": "1",

"viIdenUs": "45454545",

"viValueTypeElec": "30",

"viValueQuantity": 3000

}

ViOperacion = 3, es la operación del método o del recurso.

viIdRo = 2 , es el código de la ruleta.

viTypeElec valores : (1= Numero, 2 = Color)

viIdenUs = identificacíon de usuario.

viValueTypeElec = aquí recibe las opciones númericas de 0 a 36 y dos tipos de colores RED o BLACK.

viValueQuantity = valor apostado.

CURL:

curl --location --request POST 'https://e6gr5zk027.execute-api.us-east-1.amazonaws.com/dev/managementrultypeop' \

--header 'Content-Type: application/json' \

--data-raw '{

"viOperation": "3",

"viIdRo": "2",

"viTypeElec": "1",

"viIdenUs": "45454545",

"viValueTypeElec": "30",

"viValueQuantity": 3000

}'

1. EL Cuarto recurso. Optiene los valores actualices de la ruleta.

CURL:

curl --location --request POST 'https://e6gr5zk027.execute-api.us-east-1.amazonaws.com/dev/setstateoperationgetrulbyid' \

--header 'Content-Type: application/json' \

--data-raw '{

"viOperation": "4",

"viIdRo": "2",

"viTypeElec": "0",

"viIdenUs": "0",

"viValueTypeElec": "0",

"viValueQuantity": 0

}'

El código fuente se encuentra en el siguiene enlace.

<https://github.com/romergi1/ProyectosGER/tree/master/sd-bbog-masiv-test-20201129%202>