

Técnicas de Programación Avanzada

TECNICATURA UNIVERSITARIA EN DESARROLLO DE SOFTWARE

Tarea 5: Hilos en Java

Cafetería Concurrente

En esta ocasión, deseamos simular el proceso de preparación de café. Nuestra cafetería tendrá varias máquinas de café y cada una puede preparar diferentes tipos, por ejemplo:

- Espresso
- Capuchino
- Latte
- Americano
- Cortado

Cada vez que el cliente pida un café, se asignará una máquina disponible para prepararlo. Luego se regresa al menú principal y otro cliente puede pedir otro café siempre y cuando hayan máquinas disponibles para preparar.

El objetivo es crear un programa que simule el funcionamiento de las máquinas y la preparación de los pedidos. Cada máquina de café será un hilo independiente y puede preparar cualquiera de los tipos mencionados.

Cada máquina tendrá un tiempo de preparación aleatorio para el café. Cada tipo de café tendrá un rango de tiempo determinado, a continuación un ejemplo (se pueden usar otros valores):

- Espresso: 10-15s
- Capuchino: 12-20s
- Latte: 5-30s
- Americano: 25-30s
- Cortado: 10-20s

Luego de terminar de prepararlo, el empleado realizará una operación adicional de libre implementación, se brindan algunas ideas a continuación, pero se sugiere implementar ideas propias:

- Mensaje que escriba una palabra aleatoria en el vaso
- Mostrar una lista de datos interesantes sobre el café
- Mostrar la lista de ingredientes según el tipo de café que se preparó
- Según el tiempo de preparación, mostrar cuánta energía se ahorró
- Sugerir otra variante de café que tarde en promedio similar al elegido (diferencia menor a 5 segundos respecto el promedio de otro café)

El programa permitirá realizar dos acciones, pedir un café o salir, al pedir el usuario escribirá la variante que quiere. El programa primero mostrará el tiempo estimado (los rangos de tiempo), y cuando termine mostrará el tiempo que ocupó realmente, seguido de la operación adicional elegida a implementar.

El sistema contará con la siguiente arquitectura:

Clases

Cafe

Una clase normal que tendrá los atributos:

- Tipo: Cadena
- TiempoDePreparacion: Entero

Incluir los getters y setters necesarios.

MaquinaCafe (extendiendo desde la clase Thread)

Atributos necesarios de la clase:

- Nombre: cadena. Puede usarse para identificar numéricamente a las máquinas, o identificarlas según modelo con número de serie.
- Café: Instancia de Cafe. Será el objeto con el que se trabajará un café puntual
- Random (opcional, recomendado): De forma opcional, se puede incluir una instancia de Random para utilizarla en run(), no es necesario, pero más óptimo que recrear instancias en cada ejecución.

Métodos de la clase:

- **preparar(Cafe)**: Aquí la máquina recibirá el café y lo asignará a su atributo, y procederá a iniciar su hilo
- **run()**: Método de ejecución del hilo. Realizará las siguientes acciones en orden:
 - Indicar la máquina que trabajar y el café que se va a preparar
 - Determinar y asignar el tiempo de preparación según el tipo de café pedido
 - Esperar el tiempo asignado pausando el hilo
 - Ejecutar el método prepararAdicional(café) luego de terminar el café
- **prepararAdicional(Cafe)**: Como se indicó al principio del enunciado, este método será de implementación libre con algunas de las opciones, o alguna opción alternativa del mismo nivel de complejidad de las sugerencias.

App (clase orquestadora del proyecto)

Atributos necesarios:

- Lista de máquinas de café disponibles
- Lista de pedidos de café

Métodos necesarios para el funcionamiento del programa:

- **InicializarMaquinas()**: Se encarga de crear las instancias disponibles. Se tendrán 3 máquinas disponibles para preparar café.
- **AsignarMaquina(café)**: Se encarga de preparar el café con la siguiente metodología:
 - Comprobar si la lista de pedidos no está vacía, si lo está, no se hará nada
 - Si no está vacía sucederá lo siguiente:
 - Recorremos todas nuestras máquinas, para cada una, preguntamos si no está activa (lo hacemos con !maquina.isAlive()). Si esto se cumple crearemos una nueva instancia de MaquinaCafe que reemplace a esta que ya finalizó.
 - Con esta máquina creada, prepararemos el café solicitado
 - Eliminamos el pedido de la lista de pedidos
 - Salimos del ciclo con return.

- **Main():** El método principal será el que se encarga del flujo e instanciación como se describe a continuación:
 - Inicialización de las máquinas
 - Inicialización de la lista de pedidos
 - Uso de Scanner para poder recibir entrada del usuario
 - Uso de booleano para repetir el menú las veces necesarias
 - Ciclo while con esta variable con el siguiente modo de trabajo:
 - Mostrar el menú con las dos opciones (pedir café, o salir)
 - Recibir la entrada del usuario
 - Con la estructura switch, en la opción de pedir café:
 - Mensaje indicando las variantes
 - Se recibe el tipo que quiere el cliente
 - Se crea una instancia de café y agrega a la lista de pedidos
 - Se procede a asignar la máquina para este café

Cuando el cliente pida un café, este pasará a ser preparado (si hay una máquina disponible) y el programa regresa al menú principal donde otro cliente puede pedir otro pedido.

Cuando un pedido finalice, mostrará su salida en la consola. Esto ocurrirá en cualquier momento sin afectar el uso del programa principal.