Programación multihilo

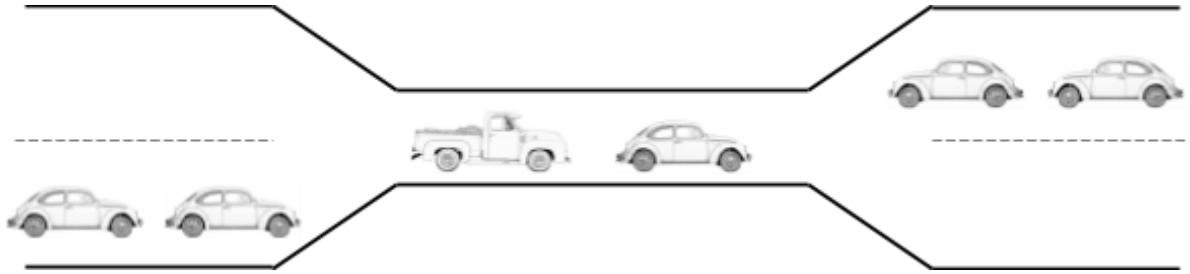
Control de tráfico

Índice:

Enunciado

Nos han encargado implementar un programa que controle el tráfico por una vía estrecha en la que sólo hay un carril de circulación.

Se trata de un punto problemático porque por ese carril deben pasar vehículos en ambos sentidos:

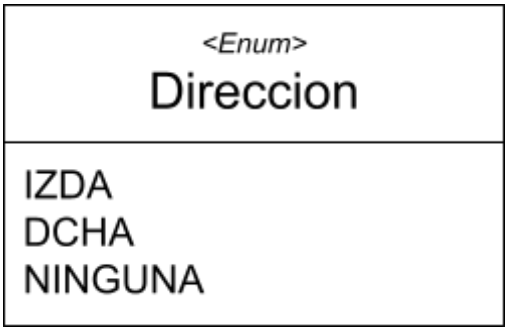


Nos han pedido que hagamos un programa que permita el paso de un sólo vehículo.

Usa `synchronized`, `wait()` y `notifyAll()` para la resolución de este ejercicio.

Implementación general

Implementa un enumerado que indique las direcciones de los coches:



También la interfaz **`CarrilUnico`**:

Donde:

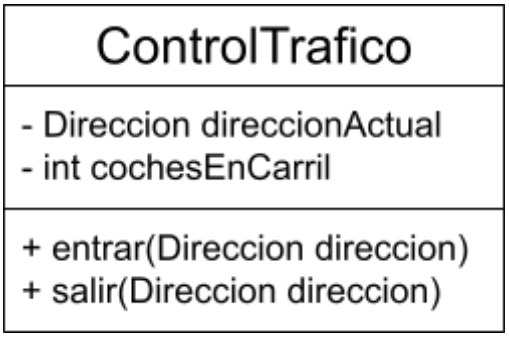
- El método **entrar** se usa para indicar que un coche entra al carril desde la dirección especificada. En todos los casos **debe esperar si hay coches en la dirección opuesta** (escalar `InterruptedException`).

- El método **salir** indica que un coche sale del carril. Debe notificarlo a los coches que están esperando para que puedan continuar.

⚠️ Esta interfaz se implementará en cada caso teniendo en cuenta que, al ser usados por múltiples vehículos, deben sincronizarse.

ControlTrafico

Vamos a crear una clase monitor **`ControlTrafico`** que sirva para controlar el tráfico que pasa por el paso estrecho:



Implementará la interfaz **`CarrilUnico`**

El atributo `cochesEnCarril` indica el número de vehículos que están circulando por el estrecho. Inicialmente debe estar a cero.

El atributo `viaOcupada` será **`true`** si hay un vehículo en el paso estrecho.

- **`entrar(Direccion direccion)`**
  Escalará la excepción `InterruptedException`.

  El método hará esperar al vehículo mientras haya algún vehículo pasando o la dirección del vehículo que quiere entrar no sea la misma que la actual. Mientras espera, se debe mostrar un **mensaje claro** del vehículo que está esperando, en qué dirección circula y cuál es la dirección actual que se está usando.

  Después, cuando el vehículo entra, si no hay coches circulando por el paso, establecerá `direccionActual` con el valor de la dirección del coche que entra. Obviamente el valor de `cochesEnCarril` debe incrementarse, ya que el vehículo está entrando.

  Se debe mostrar un **mensaje claro** del vehículo que está entrando, en qué dirección circula, cuántos coches hay en el carril y cuál es la dirección actual.

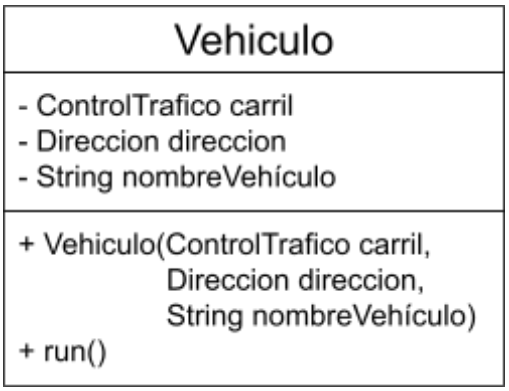
- **`salir(Direccion direccion)`**
  El nº de coches en el carril se decrementará. Si no hay coches circulando (hay cero coches en el carril):

  - Se debe actualizar la dirección actual como NINGUNA.

- Se muestra un mensaje indicando que el carril está libre y que se va a avisar a los vehículos en espera.
- Notificar a todos los vehículos en espera (de un lado y de otro) que el carril se ha liberado.

Vehículo

Los vehículos serán hilos con la siguiente estructura:



El constructor establecerá los valores del estado del vehículo.

El método run define lo que hará cualquier vehículo:

Mostrará por consola un mensaje **identificándose** y diciendo que ha llegado, la **dirección** en la que va y que quiere entrar.

Ejemplo:

```
Hyundai i30 (circulando hacia la DCHA) ha llegado y quiere entrar.
```

Entrará en el carril. Cuando lo haga debe mostrar un mensaje similar a este:

```
Hyundai i30 (circulando hacia la DCHA) está CRUZANDO el carril.
```

Esperará entre 1 y 3 segundos, para simular las distintas velocidades de los vehículos.

Después saldrá del carril. Cuando lo haga debe mostrar un mensaje similar a este:

```
Hyundai i30 (circulando hacia la DCHA) SALE el carril.
```

⚠️ Si al entrar, el vehículo es interrumpido, debe mostrar el siguiente mensaje:

```
Hyundai i30 (circulando hacia la DCHA) ha sido INTERRUMPIDO.
```

Main

Crea un `Main.java` donde:

- Se mostrará un mensaje indicando que la prueba ha comenzado.
- Se crea el control de tráfico.
- Se crean varios vehículos en un bucle que les asigne una dirección a cada uno a tu elección.

  Deja que pase un segundo antes de continuar y lánzalos.

- El hilo principal esperará a que todos los hilos terminen.
- Se mostrará un mensaje indicando que la prueba ha terminado.

Lanzar la ejecución y ver el resultado por consola.

Entrega

Adjunta en la entrega tanto el **código** como un **documento** (en formato Google Documentos) que explique la solución.