

Laboratorio TAD

Estructura de datos

Juan Manuel López Vargas

Sebastián Almanza Galvis

David León Velásquez

Juan Sebastián Méndez

Profesor: John Corredor

Fecha: 16 de agosto 2024

Diseño:

****Paso 1: Implementar el TAD Vehículo****

- ****Crea la clase Vehículo**** con los atributos `placa` y `tipo`.
- ***Implementa los métodos*** `ObtenerPlaca` y `ObtenerTipo` para acceder a los valores de estos atributos.

****Paso 2: Implementar el TAD Caseta****

- ****Crea la clase Caseta**** con los atributos `cola` (una cola de vehículos), `totalRecaudado` (para el monto total recaudado) y `recaudacionPorTipo` (un mapa para almacenar el total recaudado por cada tipo de vehículo).
- ***Implementa el método*** `AgregarVehiculo` para agregar un vehículo a la cola de la caseta.
- ***Implementa el método*** `CobrarPeaje` para cobrar el peaje al vehículo que está al frente de la cola, actualizar la recaudación y retirar el vehículo de la cola.
- ***Implementa el método*** `RealizarArqueo` para devolver el total recaudado y reiniciar las cuentas de la caseta.
- ***Implementa los métodos*** `ObtenerTamanoCola` y `ObtenerRecaudacionPorTipo` para acceder a la información de la cola y la recaudación.

****Paso 3: Implementar el TAD PuntoDeCobro****

- ****Crea la clase PuntoDeCobro**** con los atributos `casetas` (una lista de casetas) y `tarifas` (un mapa de tarifas por tipo de vehículo).

- *Implementa el método* `AgregarVehiculo` para agregar un vehículo a la caseta con la cola más corta.
- *Implementa el método* `CobrarPeaje` para cobrar el peaje en la caseta especificada.
- *Implementa el método* `RealizarArqueoTotal` para realizar el arqueo de todas las casetas y obtener el total recaudado.
- *Implementa el método* `ObtenerRecaudacionPorTipo` para calcular el total recaudado por un tipo específico de vehículo en todas las casetas.

****Paso 4: Crear un programa principal para usar los TADs****

- *Inicializa las tarifas* para los diferentes tipos de vehículos.
- **Crea una instancia de PuntoDeCobro** con un número específico de casetas y las tarifas establecidas.
- *Simula la llegada de vehículos* al punto de cobro y asigna cada vehículo a la caseta correspondiente.
- *Realiza cobros de peaje* y muestra la recaudación.
- *Realiza el arqueo de caja* para obtener el total recaudado y reiniciar las cuentas.

****Paso 5: Prueba y depura el código****

- *Ejecuta el programa* y verifica que todas las funciones se comporten como se espera.
- *Realiza pruebas con diferentes escenarios* (diferentes tipos de vehículos, diferentes cantidades de vehículos) para asegurarte de que el sistema funcione correctamente.
- *Depura* cualquier problema o inconsistencia que surja durante la ejecución.

****Implementación:****

****TAD Vehículo****

- **Estado:**

- placa: String
- tipo: Int

- **Interfaz:**

- `Vehiculo(placa, tipo)`
 - *Post:* Crea un vehículo con la placa y el tipo especificados.
- `ObtenerPlaca() : String`
 - *Post:* Retorna la placa del vehículo.
- `ObtenerTipo() : Int`
 - *Post:* Retorna el tipo del vehículo.

****TAD Caseta****

- **Estado:**

- cola: Queue[Vehiculo]
- totalRecaudado: Float
- recaudacionPorTipo: Map[Int, Float]

- ****Interfaz:****
 - `Caseta()`
 - ***Post:*** Crea una nueva caseta con una cola vacía y una recaudación inicial de 0.
 - `AgregarVehiculo(vehiculo: Vehiculo)`
 - ***Post:*** Agrega un vehículo a la cola de la caseta.
 - `CobrarPeaje(tarifa: Float, vehiculo: Vehiculo) : Float`
 - ***Post:*** Cobra la tarifa al vehículo, actualiza la recaudación de la caseta y retira el vehículo de la cola. Retorna la tarifa cobrada.
 - `RealizarArqueo() : Float`
 - ***Post:*** Realiza el arqueo de caja, retornando el total recaudado hasta el momento y reiniciando la cuenta de la caseta.
 - `ObtenerTamanoCola() : Int`
 - ***Post:*** Retorna el tamaño de la cola de vehículos en la caseta.
 - `ObtenerRecaudacionPorTipo(tipo: Int) : Float`
 - ***Post:*** Retorna el total recaudado por vehículos del tipo especificado.

****TAD PuntoDeCobro****

- ****Estado:****
 - casetas: List[Caseta]
 - tarifas: Map[Int, Float]
- ****Interfaz:****
 - `PuntoDeCobro(numCasetas: Int, tarifas: Map[Int, Float])`

- **Post:** Crea un punto de cobro con `numCasetas` casetas y las tarifas especificadas por tipo de vehículo.
- `AgregarVehiculo(vehiculo: Vehiculo)`
 - **Post:** Agrega el vehículo a la caseta con la cola más corta.
- `CobrarPeaje(indiceCaseta: Int)`
 - **Post:** Cobra el peaje al vehículo en la caseta especificada por `indiceCaseta`.
- `RealizarArqueoTotal() : Float`
 - **Post:** Realiza el arqueo de caja de todas las casetas, retornando el total recaudado hasta el momento.
- `ObtenerRecaudacionPorTipo(tipo: Int) : Float`
 - **Post:** Retorna el total recaudado por vehículos del tipo especificado en todas las casetas.