

Algorítmica y Programación por Objetos 2

Ejercicio de Nivel 8

Parqueadero

Descripción global

Se quiere construir una aplicación para administrar un parqueadero. Dicho parqueadero tiene 40 puestos enumerados, en cada puesto se puede parquear un carro. Se conoce la hora actual del parqueadero, la cual corresponde a un valor entre 6:00 y 20:00, dado que el parqueadero está abierto entre 6 de la mañana y cierra a las 8 de la noche. Los carros pueden ingresar únicamente durante las horas de apertura, al alcanzarse la hora de cierre se prohíbe el acceso. El parqueadero tiene una tarifa por hora, la cual se utiliza para estimar el valor que deben cancelar los carros según la cantidad de horas que permanecen dentro del parqueadero. Se pueden modificar la hora actual y la tarifa, al avanzar una hora el reloj o ingresar una nueva tarifa. En todo momento, se conoce la lista de carros que están parqueados en el parqueadero.

Objetivos del ejercicio

El objetivo de este ejercicio es que el estudiante comprenda y adquiera práctica en:

- El desarrollo de aplicaciones siguiendo un proceso incremental.
- La creación y captura de distintos tipos de excepciones para informar al usuario de manera conveniente cualquier problema detectado.
- La construcción de métodos para hacer persistir la información del modelo del mundo por medio del mecanismo de serialización.
- La construcción de métodos para importar y exportar información desde y hacia archivos secuenciales de texto, utilizando las clases que provee Java para manejo de archivos y manipulación de cadenas de caracteres.
- La utilización del elemento gráfico JComboBox de Java como mecanismo para manejar listas desplegables en la interfaz de usuario.

Los siguientes pasos conforman el plan sugerido para desarrollar el ejercicio. La idea es ir desarrollando y probando incrementalmente los métodos de las clases.

Este ejercicio debe ser realizado de manera **INDIVIDUAL**.

Preparación

Esta sección presenta una lista de chequeo de todas las tareas necesarias para la preparación del ejercicio. Por favor, revise que cada tarea haya sido completada **antes** de pasar a la siguiente sección de esta guía de trabajo.

Nota: En el siguiente enlace se encuentran las instrucciones de instalación de Java y Eclipse: <https://cupitaller.uniandes.edu.co/instaladores/>.

Universidad de los Andes | Vigilada MinEducación.

Reconocimiento como Universidad, Decreto 1297 del 30 de mayo de 1964 Personería Jurídica: Resolución 28 del 23 de febrero de 1949 MinJusticia.

1. Para conocer el funcionamiento esperado de la aplicación, descargue y/o ejecute el archivo demo del ejercicio que se encuentra en el siguiente enlace:
<https://cupi2.virtual.uniandes.edu.co/ejercicios-del-semester-apo2/ejercicio-n8>.
2. Descargue el esqueleto del ejercicio que se encuentra en el siguiente enlace:
<https://cupi2.virtual.uniandes.edu.co/ejercicios-del-semester-apo2/ejercicio-n8>.
3. Descomprima este archivo e importe el proyecto llamado **n8_parqueadero** a Eclipse. Limpie el ejercicio; para ello en Eclipse vaya a: **Project > Clean > Ok**. En el siguiente enlace hay un video con un ejemplo de cómo limpiar un proyecto en eclipse:
<https://youtu.be/mbcpY46wXS0>.
4. Lea el enunciado del problema disponible en:
n8_parqueadero/docs/specs/Descripcion.pdf.
5. Estudie el documento de requerimientos funcionales disponible en:
n8_parqueadero/docs/specs/RequerimientosFuncionales.pdf.
6. Estudie el documento de requerimientos no funcionales disponible en:
n8_parqueadero/docs/specs/RequerimientosNoFuncionales.pdf.
7. Estudie el documento de consideraciones adicionales de diseño disponible en:
n8_parqueadero/docs/specs/ConsideracionesAdicionalesDeDisenho.pdf.
8. Estudie el modelo del mundo diseñado para este ejercicio. Este modelo se encuentra en: **n8_parqueadero/docs/specs/ModeloConceptual.png**. Identifique las clases, relaciones entre clases, constantes, atributos y métodos.
9. Asegúrese de tener activado el uso de aserciones para la ejecución del programa. Puede ver el tutorial en: <http://cupitaller.virtual.uniandes.edu.co/videos-guia/>.

Desarrollo

Parte 1: Clase Puesto

1. Copie la clase Puesto desarrollada en el nivel 7.
2. Modifique la declaración de la clase para que sea serializable.

Parte 2: Clase Carro

1. Copie la clase Carro desarrollada en el nivel 7.

2. Modifique la declaración de la clase para que sea serializable.
3. Copie la documentación y la declaración del atributo **dias** que se encuentran en el archivo adjunto Carro.txt.
4. Copie la documentación y la implementación del método **avanzarDia** que se encuentran en el archivo adjunto Carro.txt.
5. Reemplace el método **darTiempoEnParqueadero** junto con su documentación con el método que se encuentra en el archivo adjunto Carro.txt.

Parte 3: Creación de nuevas excepciones

1. Revise el modelo del mundo e identifique las clases que permiten manejar las excepciones de la aplicación. Estudie su relación y dependencia con el resto del modelo.
2. Implemente y documente las clases propias de excepción de la aplicación. Estas son: **PersistenciaException**, **FormatoArchivoException**, **PlacaException** y **EstadoParqueaderoException**. Para su implementación, puede guiarse por su definición en el modelo conceptual, en el documento de descripción y en el documento de consideraciones adicionales de diseño. Las excepciones **DEBEN** estar de acuerdo al modelo conceptual.

Parte 4: Clase Parqueadero

En el archivo Parqueadero.txt encontrará los contratos y firmas de algunos de los métodos que debe completar de la clase Parqueadero. Recuerde reemplazar los contratos y las firmas de sus métodos con los proporcionados en el archivo.

1. Copie la clase Parqueadero desarrollada en el nivel 7.
2. Copie el método siguienteDia que se encuentra en el archivo Parqueadero.txt.
3. Modifique el método darPlacaCarro para que cumpla con la documentación que se encuentra en el archivo Parqueadero.txt.
4. Modifique el método ingresarCarro para que cumpla con la documentación que se encuentra en el archivo Parqueadero.txt.
5. Modifique el método sacarCarro para que cumpla con la documentación que se encuentra en el archivo Parqueadero.txt.
6. Modifique el método buscarPuestoLibre para que cumpla con la documentación que se encuentra en el archivo Parqueadero.txt.

7. Modifique el método `buscarCarroPorPlaca` para que cumpla con la documentación que se encuentra en el archivo `Parqueadero.txt`.
8. Copie la documentación y la signature del método `guardar` que se encuentra en el archivo `Parqueadero.txt`. Implemente este método para que cumpla con la documentación.
9. Copie la documentación y la signature del método `cargar` que se encuentra en el archivo `Parqueadero.txt`. Implemente este método para que cumpla con la documentación.
10. Copie la documentación y la signature del método `importarArchivoTexto` que se encuentra en el archivo `Parqueadero.txt`. Implemente este método para que cumpla con la documentación.
11. Copie la documentación y la signature del método `generarReporte` que se encuentra en el archivo `Parqueadero.txt`. Implemente este método para que cumpla con la documentación.
12. Modifique el método constructor para que cumpla con la documentación que se encuentra en el archivo `Parqueadero.txt`.

Parte 5: Modificación de la interfaz

En `PanelLista` completar los TODOs que comienzan por: **//TODO Parte 5 punto X**

Validación

Para comprobar el funcionamiento de su ejercicio usted puede:

1. Ejecutar el programa e interactuar con todas las opciones disponibles en la interfaz. Los resultados obtenidos deben ser iguales a aquellos mostrados en el video demo.
2. Ejecutar las pruebas automáticas disponibles en el ejercicio. En el siguiente video <https://youtu.be/rVd4AD8XMJk> se explica cómo efectuar esas pruebas. Estas pruebas deben presentar resultados en verde (0 errores y 0 fallas) cuando el ejercicio ha sido completado correctamente.

Tenga en cuenta que esas pruebas no son exhaustivas y que su correcto funcionamiento no garantiza que no haya ningún error en su programa.

Entrega

Este ejercicio debe ser realizado de manera **INDIVIDUAL**.

1. Indente el código fuente de todas las clases del mundo. En el siguiente enlace <https://youtu.be/cwQ9QiauaSc> encuentra un video que explica cómo indentar el código fuente de su ejercicio.
2. Limpie el proyecto para que la entrega no contenga archivos ejecutables ni temporales (<https://youtu.be/mbcpY46wXS0>).
3. Construya el archivo entregable con su ejercicio desarrollado y validado completamente. En el siguiente video <https://youtu.be/xuSDFfEZW78> se explica detalladamente el proceso para producir el comprimido del ejercicio y enviarlo a SicuaPlus. Renombre el archivo a entregar con su login de la siguiente forma:

n<nivel del ejercicio>_<login estudiante>.zip
Por ejemplo: **n8_tsuarez.zip**

La no indentación del código fuente o el nombramiento incorrecto del ejercicio en su entrega es una acción penalizada en la plantilla de calificación del mismo.

4. Entregue el archivo del ejercicio vía SicuaPlus, de acuerdo con las normas, fecha y hora de entrega.