Universidad de Alcalá - Escuela Politécnica
Departamento de Ciencias de la Computación

PROGRAMACIÓN ORIENTADA A OBJETOS (Laboratorio de Prácticas)

Titulaciones de Grado en Ingeniería de Computadores, Ingeniería Informática y Sistemas de Información

EJERCICIO 2

1. Ejercicio Aparcamiento

Se pretende realizar una aplicación en Java que simule un aparcamiento donde se pueden estacionar vehículos por un tiempo limitado. Existen dos tipos de vehículos que pueden estacionar en el aparcamiento: automóviles y camiones. De todos los vehículos hay que guardar su matrícula, la fecha/hora de entrada en el aparcamiento y si se trata de un vehículo con abono, de los automóviles el tipo: turismo, todoterreno y furgoneta, y de los camiones el número de ejes.

La clase Vehículo tendrá un método abstracto llamado calcularImporte() que tendrán que implementar las clases hijas (polimorfismo) y que será el encargado de determinar los minutos que han transcurrido y mediante la siguiente fórmula calculará el importe del tiempo de aparcamiento:

- Automóvil:
  - turismo → minutos * 1.5 € / 60
  - todoterreno → minutos * 2.5 € / 60
  - furgoneta → minutos * 3.5 € / 60
- Camión:
  - Ejes ≤ 3 → minutos * 4.5 € / 60
  - Ejes > 3 → minutos * 6.5 € / 60

Si se trata de un vehículo con abono se le aplicará un descuento del 40% sobre el importe final.

Hay que desarrollar una clase Aparcamiento que se encargará de almacenar los vehículos que tiene estacionados, para hacerlo usará una colección de tipo HashMap donde la clave será la matrícula del vehículo. El aparcamiento tiene un número de plazas determinada que se establece a través de un atributo denominado capacidad (int). La clase contará con los siguientes métodos:

- intoducirVehiculo(Vehiculo v): Se encargará de añadir un nuevo vehículo al aparcamiento comprobando que no está dentro y que hay capacidad suficiente. También decrementará la capacidad del aparcamiento.
- sacarVehiculo(String matricula): Se encargará de sacar un vehículo del aparcamiento comprobando que está dentro y calculará cuanto tiene que cobrar haciendo uso del método calcularImporte() del vehículo correspondiente. También incrementará la capacidad del aparcamiento.

La aplicación tiene que hacer uso de la clase AparcamientoException para la utilización de excepciones. *Ejemplo de utilización: Si detectamos que queremos sacar un vehículo del aparcamiento y éste no está, ejecutaremos una sentencia como la siguiente:*
throw new AparcamientoException(AparcamientoException.VEHICULO_FUERA);

```
public class AparcamientoException extends Exception {

    public static String VEHICULO_DENTRO = "El vehículo está en el
aparcamiento.";

    public static String VEHICULO_FUERA = "El vehículo no está en el
aparcamiento.";

    public static String APARCAMIENTO_LLENO = "El aparcamiento está
lleno.";

    public AparcamientoException() {
      super("Se ha producido una excepción en la aplicación.");
    }

    public AparcamientoException(String txt) {
        super(txt);
    }
}
```

Una vez codificadas las clases de lógica de negocio hay que realizar una aplicación Swing de forma que permita la gestión del aparcamiento. Se tienen que distinguir dos funcionalidades distintas en la aplicación, por un lado se podrá introducir un nuevo vehículo en el aparcamiento indicando todos sus datos, y por otro se podrá sacar un vehículo del aparcamiento con tan solo su matrícula.

Cuando se saque un vehículo del aparcamiento se deberá guardar un fichero de texto (cuyo nombre será la matrícula del vehículo y la fecha de salida) que simule una factura donde se indique la matrícula del vehículo, la fecha de entrada, los minutos transcurridos y el importe final.

Se deberá implementar la persistencia de los datos de la aplicación de forma que si se cierra se deberán guardar sus datos y recuperarlos cuando se vuelva a abrir.

Ejercicio obligatorio II y SI, voluntario IC: Además de la funcionalidad descrita anteriormente se desea conocer la cantidad de automóviles de distinto tipo que hay en el aparcamiento en un momento dado. Para realizarlo, la aplicación presentará un botón etiquetado con "Estadísticas" que mostrará una nueva ventana (puede ser un diálogo) donde se indicará el número de turismos, todoterrenos y furgonetas que hay en el aparcamiento en ese momento.

Entregar el proyecto Netbeans comprimido en la sección "Trabajos" de Blackboard en el grupo correspondiente.