



1. Supuesto práctico

Se va a proceder a modificar el actual sistema de préstamos de bicis en la ciudad. Por una parte, se pondrán a disposición de los usuarios diferentes tipos de bicis y, por otra parte, se definirán unos *packs* que permitirán rebajar la fianza que se cobra a los usuarios en algunos casos en los que se prestan varias bicis con una única tarjeta.

Se definirá una interfaz *Resource* (recurso) que será implementada por una clase *Bike*, por una parte y, por otra parte, por una clase *Pack*. Todos los recursos responden a la consulta `getDepositToPay(double deposit)` que aplica un factor de corrección a la fianza que se cobra al realizar un préstamo.

Un *pack* es simplemente una agrupación de bicis que se prestan conjuntamente. En principio habrá dos tipos de *packs*, uno para grupos y otro para familias.

El sistema podrá prestar bicis individualmente o *packs* (de grupos o familiares).

1.1. Detalles sobre las bicis

En principio se definen 3 tipos distintos de bicis en el sistema: bicis para adultos (*AdultBike*), bicis infantiles (*ChildBike*) y bicis eléctricas (*ElectricBike*). En el caso de las bicis infantiles, `getDepositToPay(double deposit)` garantizará que se aplica un descuento de un 15 % sobre la fianza normal que establece el Ayuntamiento, mientras que para las bicis de adulto se aplica la fianza, sin descuento, establecida en el sistema.

Toda bici tiene un identificador, una marca y un modelo. También para todos los tipos de bici se conoce su talla, peso, número de platos (al menos 1) y número de piñones (al menos 1). Estos datos se establecen al crear la bici y no son modificables. Las tallas de las bicis infantiles son números pares del 12 al 26, que representan el tamaño en pulgadas de la rueda. En el caso de las bicis de adulto las tallas de las bicis son S, M, L y XL.

Las bicis eléctricas además tienen información sobre su batería y su motor. Cuando se crea una bici eléctrica se indica el par del motor en Newtons por metro (Nm) e información sobre la batería: el voltaje en Voltios y su carga eléctrica en AmperioHora (Ah). A partir de esta información almacenada, se dispondrá de una consulta para obtener el valor calculado que permite conocer la energía almacenada en la batería en kilovatiosHora teniendo en cuenta que $Wh = Voltios \times AmperiosHora$.

En el caso de las bicis eléctricas, la fianza a cobrar aumenta en un porcentaje a partir de la establecida en el sistema. Este porcentaje de incremento estará dado por el valor del voltaje de la batería (actualmente los valores habituales en el mercado son de 24V o 36V).

1.2. Detalles sobre los packs

El pack de grupos requiere que haya un mínimo de 10 bicis para formar un grupo y su versión de `getDepositToPay(double deposit)` garantizará que se aplica un descuento de un 20 % sobre la suma de las fianzas individuales. El pack familiar requiere que haya un mínimo de 4 bicis para formar el pack pero requiere que al menos dos de ellas sean bicis infantiles. Para este tipo de pack se garantizará que su versión de `getDepositToPay(double deposit)` aplica un descuento de un 50 % sobre la suma de las fianzas individuales.

- La inicialización de los packs estará basada en un array de bicis (`Bike[]`).

- Debe garantizarse que un pack tenga siempre el mínimo de bicis establecido (10 para los grupos, 4 para las familias).
- No podrá haber bicis repetidas en un pack.
- En el caso de los packs familiares deberá garantizarse que al menos dos bicis en el pack son infantiles.
- Se deberá poder gestionar los packs añadiendo y eliminando elementos siempre que se respeten los detalles anteriores.
- De un pack podrá conocerse la cantidad de bicis que lo forman, si una bici dada forma parte de él, así como saber qué bicis forman dicho pack (array de bicis).
- En ningún caso se permitirá tener packs formados por otros packs.
- El valor de la fianza es una aplicación de un descuento a la suma de las fianzas de las bicis contenidas en el pack que será del 20 %, en el caso de pack de grupo, y del 50 %, en el caso de las familias.
- La solución debe garantizar que si cambia tanto la fianza general, como las fianzas modificadas con el factor de cada bici en el pack, el precio del pack también cambia.

1.3. Clases

Se espera que las clases que forman el proyecto Eclipse de la entrega incluyan al menos interfaz `Resource`, una clase abstracta `Bike`, una clase abstracta `Pack`.

Todas las clases aportarán su propia definición de `toString()` y de `equals()`.

Se pedirán las clases de prueba implementadas mediante JUnit 4 para probar el pack de grupo (no el familiar) así como la bici de adulto normal solamente.

También se aportará una clase de pruebas JUnit 4 para probar el uso de los objetos polimórficamente desde el punto de vista más abstracto, esto es, cuando un pack o una bici se usa mediante una entidad del tipo dado por la interfaz `Resource`. Por ejemplo:

```
Resource r = new AdultBike(...);
fianzaModificada = r.getDepositToPay(FIANZA);
assertEquals(20, fianzaModificada, ERROR_ADMISIBLE);
```

2. Condiciones de entrega

- La entrega consistirá en un único archivo `.tar`, `.tgz`, `.tar.gz`, `zip`, `rar`.
- El archivo contendrá el proyecto ECLIPSE.
- El proyecto ECLIPSE deberá nombrarse con `entrega2-idAlumno1-idAlumno2`.
- El `idAlumno` se refiere al identificador de la cuenta de laboratorio de cada alumno.
- Ejemplo: `entrega2-javper-margar`
- El proyecto deberá compilar.
- La entrega se realizará mediante la subida del archivo así obtenido en una tarea habilitada al respecto en el aula virtual. Se entrega una sola vez por cada equipo. El equipo decide uno de sus integrantes encargado de subir la práctica.



- La fecha y hora límites para la entrega se establecen en la tarea del aula virtual y son: **22 de Diciembre de 2017 a las 23:55**. **Plazo no ampliable**
- **No se admitirán entregas que incumplan estas condiciones.**
- **En caso de incumplimiento** de las condiciones anteriores, se considerará la práctica como **no presentada**.
- Es recomendable que el código cumpla las convenciones de código JAVA¹.
- Será necesario documentar las clases mediante comentarios JavaDoc². Utilice los tags especiales @see cuando considere necesario.
- Utilice apropiadamente las anotaciones @Override.
- Es deseable que cada archivo .java contenido en la entrega tenga en la cabecera (comentarios JavaDoc) el nombre de los autores (mediante la etiqueta @author). Para indicar el nombre, se preferirá el identificador de la cuenta de laboratorio de cada alumno en lugar de su nombre completo.

2.1. Aclaración relativa a las defensas

Por razones de calendarización, para esta segunda práctica no se realizarán defensas. En cambio el examen de convocatoria ordinaria contendrá una pregunta relativa a la práctica 2.

¹<http://www.oracle.com/technetwork/java/codeconvtoc-136057.html>

²<http://www.oracle.com/technetwork/java/javase/documentation/index-137868.html>