

# Programación orientada a objetos

## Ejercicios Tema 1.7: Diseño de clases

Partiendo del proyecto *chapter07/zuul-bad*:

1. Haz las modificaciones oportunas en la clase *CommandWords* y en las que la utilicen para poder imprimir la lista de comandos válidos reduciendo el acoplamiento existente. Esto es:
  - En la clase *CommandWords* crea un método *validCommands()* que devuelva un *String* con todos los comandos.
  - En la clase *Parser* un método similar que también devuelva un *String* con todos los comandos. Este método básicamente delega en el de *CommandWords*.
  - En la clase *Game*, modifica el método *printHelp()* para que utilice el del *Parser*.
2. Cambia la clase *Room* para que utilice un *HashMap* para almacenar las salidas de la habitación. Haz los cambios adecuados en la clase *Game* para aprovechar la nueva especificación de la clase *Room*.
  - Define un atributo *exits* en *Room* que sea un *HashMap* de *String* (clave) y *Room* (valor). Habrá que inicializarlo en el constructor.
  - Define un método para introducir una nueva salida a la habitación:  

```
public void setExit(String direction, Room neighbor)
```

Este método puede utilizar el método *put(clave, valor)* de la clase *HashMap* para introducir este nuevo par en el Map.
  - Define un método para obtener la habitación de salida por una dirección (*String*) de la habitación:  

```
public Room getExit(String direction)
```

Este método puede utilizar el método *get(clave)* de la clase *HashMap* que devuelve el objeto asociado (en este caso, de tipo *Room*).
  - Define un método que devuelva la lista de direcciones de salida posibles para una habitación (para poder utilizarlo en el método *printLocation()*):  

```
private String getExitString()
```

Este método devuelve una salida de la forma:  
*Exits: north arriba sur.*
  - En la clase *Game* habrá que modificar los métodos *createRooms()*, *printWelcome()*, *processCommand(Command command)*, y *goRoom(Command command)*.

3. Plantéate nuevas extensiones, por ejemplo:

- Hacer un programa main() que permita ejecutar el juego fuera de BlueJ.
- Cambiar el idioma de los comandos (por ejemplo, a español).
- Añade nuevos comandos al juego, por ejemplo: Agrega el comando “eat” para que cuando se ejecute aparezca el texto “I have eaten and I am not more hungry”
- Definir una configuración de habitaciones en tres dimensiones para el juego y probarla.
- Añadir llaves en el juego que puedan estar en habitaciones y que sirvan para abrir puertas especiales.
- Añadir puntuación al juego.
- Invéntate una extensión...