

Práctica 1

Tecnologías para el Desarrollo de Software

Grado en Ingeniería Informática

Mención Ingeniería del Software

Mario Corrales

Curso 2022-2023

Introducción

Formáis parte de un equipo que tiene que desarrollar una aplicación para simular un juego de cartas. Vuestro grupo debe encargarse del almacenamiento y recuperación de la información de este juego. Se os dan unas clases con los métodos necesarios, y vuestro trabajo será implementar estas clases y realizar los test necesarios para comprobar la corrección de dichas clases.

Parte 1

La estructura de la información del juego debe cumplir los siguientes requisitos:

- El juego permite tener tres tipos de elementos: cartas, mazos y personajes.
- Datos de una carta: nombre, coste, número de carta y tipo. El nombre de la carta debe tener un tamaño entre 1 y 10 caracteres. El coste no puede ser menor que 0. El número de carta debe ser mayor que 0 y menor que 200. El tipo debe ser un valor entre: ATTACK, DEFENSE y UNIQUE. **El número de la carta debe ser un identificador único.**
- Datos de un mazo: nombre. El nombre del mazo debe ser un identificador único y no puede estar vacío
- Datos de un personaje: nombre, ataque y defensa. El nombre debe ser un identificador único y no puede estar vacío. El ataque debe ser un valor mayor o igual a 0. La defensa debe ser un valor mayor o igual a 0.
- Una carta puede pertenecer a varios mazos. Un mazo está compuesto por entre 1 y 5 cartas. Un mazo solo puede tener una carta de tipo UNIQUE.
- Un mazo puede tener varios personajes asociados. Un personaje puede tener varios mazos asociados.
- Un carta se considera igual a otra si coinciden sus nombres.
- Un mazo se considera igual a otro si coinciden sus nombres.
- Un personaje se considera igual a otro si coinciden sus nombres.

Parte 2

El equipo realizará un sistema que permita realizar las siguientes operaciones:

- Crear/eliminar/obtener una carta.
- Crear/eliminar/obtener un mazo.
- Crear/eliminar/obtener un personaje.
- Actualizar los datos de una carta (salvo el número de carta).
- Actualizar los datos de un personaje (salvo el nombre).
- Actualizar las cartas que pertenecen a un mazo.
- Actualizar los personajes que están asociados con un mazo.

Por supuesto estas operaciones solamente podrán realizarse si es posible de acuerdo a los requisitos de las entidades del juego (ver Parte 1).

La práctica deberá utilizar Hibernate 5.6.5 para mapear las clases Java con las tablas de una base de datos hsqldb. En el campus virtual de la asignatura se encuentran los esqueletos de las clases que hay que implementar (*Game*, *Card*, *Deck*, *Character*). No se pueden modificar los nombres ni eliminar ningún método de las clases proporcionadas, pero si se pueden añadir métodos nuevos o atributos a dichas clases. Los métodos que se encuentran en los esqueletos de las clases proporcionadas deben implementar la funcionalidad definida en apartados anteriores. Además de la implementación de las clases, dichas clases deberán ser documentadas con Javadoc según la especificación que se puede extraer de los requisitos detallados previamente. Se deberán lanzar las excepciones *IllegalArgumentException* o *IllegalStateException* según el caso. Si existe alguna ambigüedad en las especificaciones, la validez de la implementación realizada se medirá de acuerdo al javadoc descrito para cada método/clase. Parte de la evaluación de la práctica se realizará a través de test automáticos que probarán cada uno de los métodos definidos en las clases proporcionadas, **por lo que es muy importante seguir las normas que se definen en este documento y no modificar los nombres de los métodos.**

Características del proyecto Eclipse

- El proyecto Eclipse deberá nombrarse con 'tds-practical-grupox', siendo x el número del grupo asignado.
- Deberá ser un proyecto Eclipse válido para Eclipse IDE for Java developers release 2022-06 y jdk 11.
- La implementación desarrollada por el equipo se alojará en un paquete llamado tds.practical
- Las clases de test deberán alojarse en un paquete llamado tds.practical.test
- La carpeta src del proyecto Eclipse contendrá los fuentes .java necesarios para la realización del proyecto y los ficheros de test que implementan los casos de prueba con Junit.

Control de versiones

Se llevará un control de versiones utilizando git y se utilizará gitLab (gitlab.inf.uva.es) como repositorio centralizado. Al finalizar, para realizar la entrega, se añadirá al profesor al proyecto (cuando ya no se vayan a realizar más commits y pushes al repositorio centralizado en gitLab) tal y como se indica en las normas de entrega. En el repositorio no se tendrá ningún .class ni tampoco la documentación generada que puede obtenerse con javadoc. En esta Práctica 1 no se valorará la calidad del uso de git en cuanto a ramificación y merge. Aunque es recomendable emplear el método rama por tarea, teniendo una rama develop a partir de la que ramificar e integrar, dejando master siempre para versiones revisadas y funcionales.

Base de datos hsqldb

La base de datos utilizada deberá llamarse 'practical1' y el puerto deberá ser el 8000. El usuario y contraseña deben dejarse en los valores por defecto (username sa y password vacío). Esta información deberá incluirse en fichero server.properties de hsqldb.

Issues, estimación y tiempo empleado

El proyecto en gitlab.inf.uva.es tendrá un listado de issues que se irán creando bien al comenzar el proyecto, bien durante el desarrollo del mismo. Al crear cada issue, se asignará a un/una miembro del equipo, se estimará (comando corto /estimate) el tiempo que se cree que será necesario para completar la issue y al finalizar, se indicará el tiempo que realmente ha sido necesario (comando corto /spend). La creación de ramas se hará siempre desde el issue tracker, vinculando cada issue a la rama donde se desarrolla. Las acciones de corrección de errores tendrán su correspondiente issue y por tanto rama. No se valorará cómo se ha dividido el proyecto en issues, ni tampoco tendrá ningún efecto en la nota la diferencia entre el tiempo estimado y el empleado. Lo que se valora es que se haya hecho.

Tests

Se realizará una clase de Test por cada clase a probar (*Game*, *Card*, *Deck*, *Character*). Para realizar los casos de prueba, se deberán seguir las técnicas de partición en clases de equivalencia, análisis de valores límite y particiones basadas en el estado. Estas clases deben contener métodos de prueba. Para ello se utilizará JUnit 5.

Normas de entrega

- El repositorio git y el centralizado en gitlab deberán llamarse con el mismo nombre que el proyecto java.
- La entrega se realizará añadiendo al profesor (usuario *marcorr* en gitlab) con rol Reporter al repositorio que contiene el proyecto en gitLab cuando no se vaya a realizar ningún otro commit & push.
- Cualquier push al repositorio una vez realizada la entrega será penalizado con 0 en la Práctica.
- Al entregar la práctica todo deberá quedar integrado en la rama master.

- En el tracking de versiones, y por tanto en el repositorio remoto, solamente residirán los archivos de la configuración del proyecto Eclipse, los fuentes de las clases de la solución, los fuentes de los tests y las librerías necesarias.
- El proyecto Eclipse será un proyecto válido para Eclipse IDE for Java developers release 2022-06. En ese caso los archivos necesarios para el proyecto Eclipse (.project, .settings/*, .classpath) estarán incorporados al tracking de versiones y alojados en el repositorio.
- El proyecto Eclipse también deberá subirse a la tarea habilitada en el campus en formato comprimido.
- El proyecto deberá compilar y funcionar con jdk 11. Debe poder ejecutarse directamente tras clonarlo del repositorio (salvo la inicialización del servidor de base de datos hsqldb).
- En los fuentes entregados se hará referencia a los autores de la práctica en cada archivo fuente con el tag @author de javadoc y solamente en ese punto.
- El proyecto tendrá un archivo README.md que indicará toda la información que quieran aportar los/las autores/as sobre su proyecto además de la siguiente información:
 - Tiempo total en horas-persona empleado en la realización de la práctica.
 - Clases que forman parte de la solución.
- No se admitirán entregas fuera de plazo o por otra vía distinta a la indicada en estas normas.
- Fecha límite de entrega: **4 de noviembre de 2022 a las 15:00 (hora española)**.

Criterios de evaluación

Para realizar la evaluación de la práctica, se tendrán en cuenta los siguientes criterios:

- Corrección de la implementación realizada con respecto a las especificaciones dadas.
- Documentación en javadoc.
- Realización de test.
- Creación de issues.
- Reflejo de la participación de los/as integrantes del grupo en los commits del repositorio.
- El incumplimiento de las normas de entrega se verá reflejado en la nota.