

Auxiliar Revolucionaria 1

Patrones de diseño 2: Ejercicios resueltos y propuestos

Profesor: Alexandre Bergel
Auxiliares: Juan-Pablo Silva
Ignacio Slater
Semestre: Primavera 2019

1. Definiciones necesarias

Sea el juego de cartas *Yu-Gi-Oh!* definido como la tupla $Y = \dots$

Sería interesante (?) dar una definición matemática para el juego, pero no lo voy a hacer.

Yu-Gi-Oh! es un *TCG* con varios tipos de cartas, pero para los siguientes ejercicios nos centraremos en 2 tipos en particular, *cartas de monstruos* y *cartas de hechizos*.

2. Template method

Ejercicio 1 Zonas de juego

Si bien existen muchas diferencias entre estos dos tipos de cartas, por ahora consideraremos que solo se diferencian en la zona del campo en la que son jugadas.

Defina las cartas de monstruos y de hechizos para que puedan jugarse en su zona correspondiente siguiendo buenas metodologías de diseño.

Solución del ejercicio ??

Lo primero que necesitamos hacer es crear una interfaz y una clase abstracta que represente todos los tipos de cartas, así:

```
// com.github.islaterm.yugi.card.ICard
public interface ICard {
    void playtoMath(GameMat gameMat);
}

// com.github.islaterm.yugi.card.AbstractCard
public abstract class AbstractCard implements ICard {
    @Override
    public abstract void play();
}
```

Lo que estamos haciendo al definir el método como abstracto es relegar la labor de decidir cómo se jugarán las cartas a las implementaciones particulares de cada tipo de carta. Esto es lo que se conoce como *template method pattern*.

Ahora, las implementaciones particulares de cada método serán:

```
// com.github.islaterm.yugi.card.MonsterCard
public class MonsterCard extends AbstractCard {

    @Override
    public void playtoMath(@NotNull GameMat gameMat) {
        gameMat.addMonster(this);
    }
}

// com.github.islaterm.yugi.card.SpellCard
public class SpellCard extends AbstractCard {
    @Override
    public void playtoMath(@NotNull GameMat gameMat) {
        gameMat.addSpell(this);
    }
}
```