

Patrones de diseño con Objetos

Ejercicios

Ejercicio 1:

En una sesión de code review se detecta el siguiente código. Se pide que proponga un refactor.

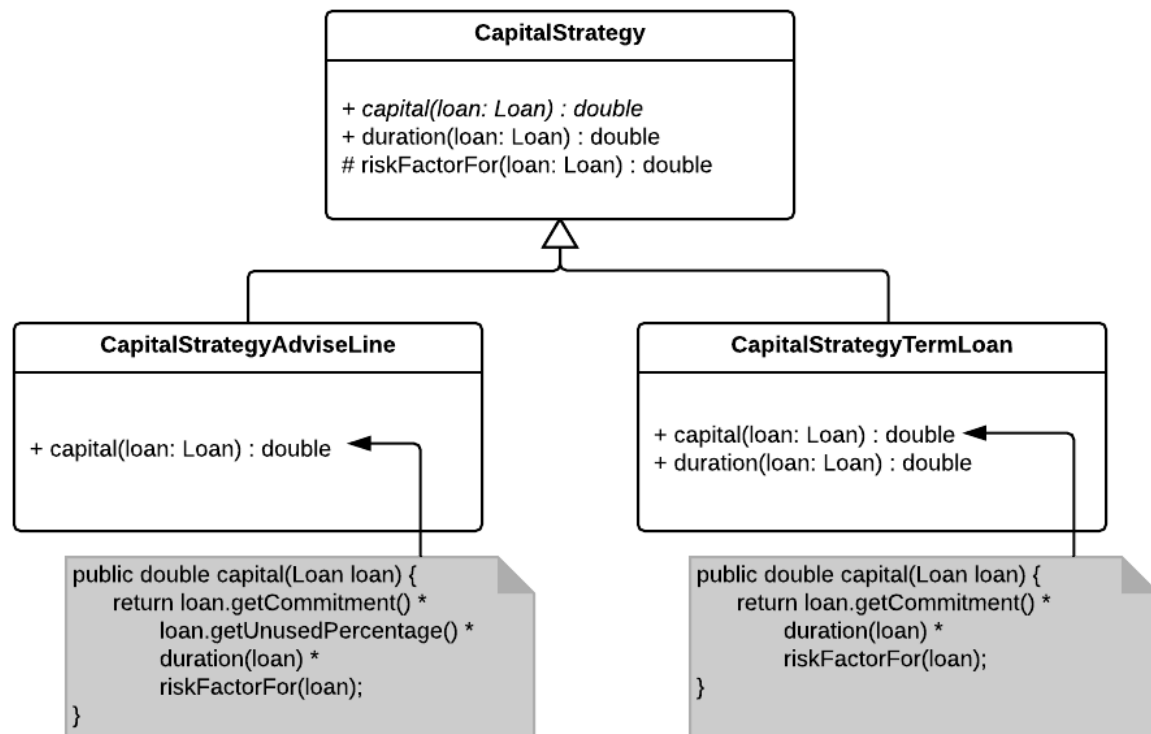
```
public class Loan {
    [...]
    public Loan(float notional, float outstanding,
                int rating, Date expiry) {
        this.strategy = new TermROC();
        this.notional = notional;
        this.outstanding = outstanding;
        this.rating = rating;
        this.expiry = expiry;
    }

    public Loan(float notional, float outstanding,
                int rating, Date expiry, Date maturity) {
        this.strategy = new TermROC();
        this.notional = notional;
        this.outstanding = outstanding;
        this.rating = rating;
        this.expiry = expiry;
        this.maturity = maturity;
    }

    public Loan(CapitalStrategy strategy, float notional, float outstanding,
                int rating, Date expiry, Date maturity) {
        this.strategy = strategy;
        this.notional = notional;
        this.outstanding = outstanding;
        this.rating = rating;
        this.expiry = expiry;
        this.maturity = maturity;
    }
}
```

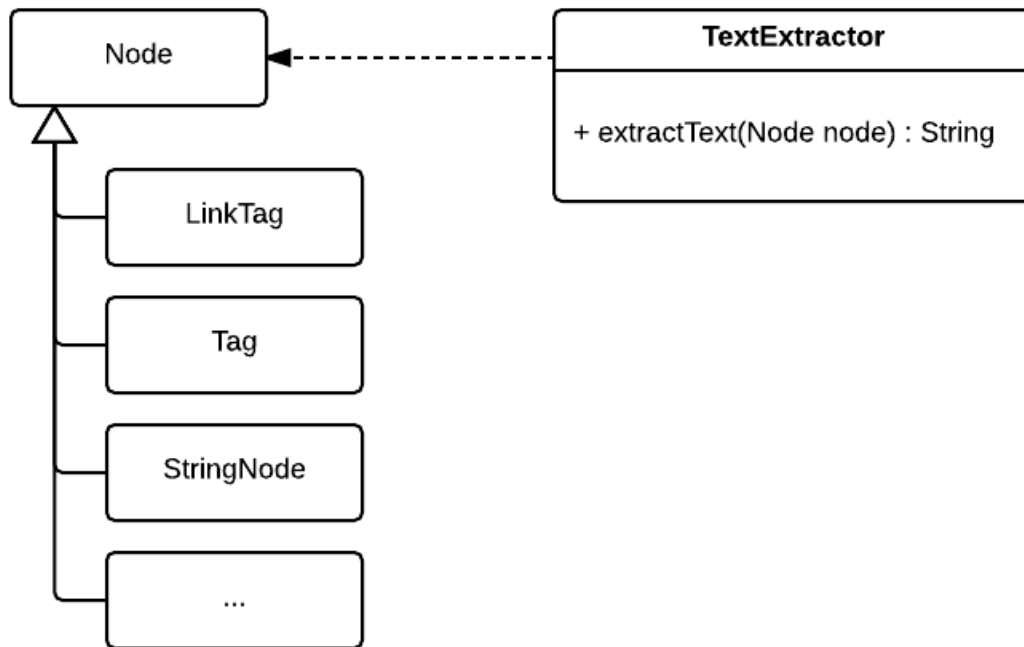
Ejercicio 2:

Se cuenta con el siguiente diseño para el cálculo del monto máximo a entregar en un préstamo según el método de amortización. Proponga una mejora para evitar la repetición de código.



Ejercicio 3:

Para un módulo de extracción de texto de markup se construyó la siguiente solución y ha pasado a revisión para mejorar el diseño. ¿Qué alternativa de diseño podría proponer para mejorar esta solución?



Donde el método extractText quedó como se indica:

```
public String extractText(Parser parser) {
    StringBuffer text = new StringBuffer();
    for (Node node : parser.nodes()) {
        if (node instanceof LinkTag) {
            extractLinkTag(node, text);
        } else if (node instanceof Tag) {
            extractTag(node, text);
        } else if (node instanceof StringNode) {
            extractStringNode(node, text);
        } else if (node instanceof ...) {
            [...]
        }
    }
    return text.toString();
}
```

Ejercicio 4:

Se propone la siguiente problemática para la que debe proponer una solución usando correctamente los patrones de diseño que corresponda. Incluya en su respuesta un diagrama de clases que ilustre su solución.

*Su empresa quiere lanzar un nuevo portón para cocheras propietarias de edificios, que se manejan con un control remoto de un único botón. Como es de esperarse, cuando el portón está cerrado, el botón activa el mecanismo de apertura, y cuando está abierto, activa el mecanismo de cierre. En caso que se oprima el botón mientras se esté abriendo, deberá interrumpir la apertura para iniciar el cierre desde el punto donde se encuentra, y viceversa en el caso de que se esté cerrando, que deberá pasar a abrirse. Por otra parte, los motores tienen indicadores de límites, que son configurables en posiciones del portón, y se usarán para demarcar la posición donde se considera **cerrado** y donde se considera completamente **abierto**.*

Además, los sistemas, contarán con un timer de cerrado, para evitar que los portones queden abiertos si un usuario por olvido, o descuido, dejara de oprimir el botón para cerrar el portón. Si bien esta sería el único propósito del timer, es deseable que luego se puedan incorporar nuevas acciones disparadas por tiempo.

Para trabajar, considere que se estableció una interfaz que cuenta con una clase 'EngineControl' con al menos dos métodos, uno para el mensaje de botón presionado, y otro para el mensaje de posición marcada alcanzada. El control deberá actuar y arrancar o detener el motor en uno de sus dos sentidos, OPEN_SIDE y CLOSE_SIDE.