



Orientación a Objetos 2 – Práctica 8

Ejercicio 1: Refactoring

Considerando la implementación de la clase Shape, provista por la cátedra. Realice las siguientes tareas:

- Revise el código y determine qué tipos de figura implementa la clase Shape.
- Defina e implemente los test cases para todos los tipos de figura.
- Tenga en cuenta que, dada la implementación, ud. debe extenderla para dar soporte a dos nuevos requerimientos:
 1. Un nuevo tipo de figura: el paralelogramo.
 2. Se requiere el cálculo del perímetro para todas las figuras.
- Discuta con un ayudante:
 1. Es posible realizar de manera sencilla las extensiones propuestas?
 2. Cómo se puede mejorar el código y el diseño?
 3. Considerando los nuevos requerimientos discuta en qué orden realizará las siguientes tareas (puede repetir algunas de ellas): refactoring, diseño final al que quiere llegar, testing, refactoring de los test cases.
- Implemente en Smalltalk los requerimientos planteados, y todas aquellas modificaciones que considere necesarias (incluyendo el mantenimiento y extensión de los test cases).

Ejercicio 2: Refactorice usando patrones

Sean las clases User, Tweet y TweetBase provistas por la cátedra que forman parte de una aplicación para realizar análisis estadísticos sobre una base de tweets.

En la clase TweetBase se encuentran implementados algunos métodos para calcular descriptores estadísticos:

- `deviationTweetPerUser`: calcula la desviación estándar de la cantidad de tweets enviados por usuario
- `varianceTweetsPerUser`: calcula la varianza de la cantidad de tweets enviados por usuario

En los dos casos, el algoritmo de cálculo tiene los mismos pasos:

1. agrupar la cantidad de tweets por usuario,
2. calcular la media de la cantidad de tweets por usuario,
3. calcular desviación o varianza usando los datos de los pasos 1 y 2

Se solicita que refactorice el código tratando de implementar alguno de los patrones de diseño que ha visto.

Tareas:

1. Realice los casos de prueba para validar los métodos provistos. En el archivo txt incluido en el material adicional encontrará expresiones que puede emplear como base para armar los casos de prueba.
2. Ejecute los casos de prueba
3. Realice todas las refactorizaciones que considere necesarias en el código de `deviationTweetsPerUser` y `varianceTweetsPerUser` de la clase `TweetBase`.
4. Ejecute los casos de prueba sobre el código refactorizado