DESAFÍO KLIMBER

Se propone como desafío facilitar la implementación de nueva funcionalidad en el proyecto DevelopmentChallenge, refactorizando el código provisto.

Dentro del proyecto se encuentra una clase que posee un método cuyo trabajo es generar reportes en base a una lista de Formas Geométricas como se observa en la Figura 1.

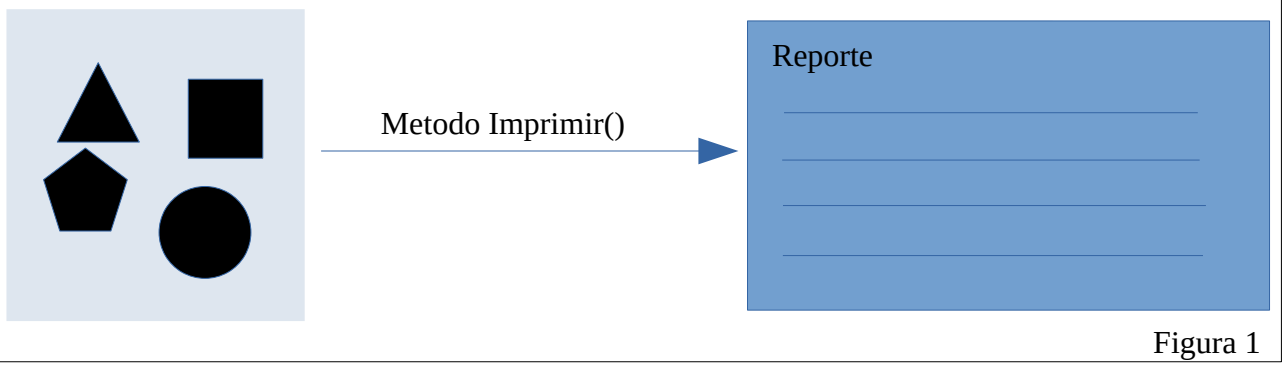

Figura 1

Los requisitos del desafío se detallan en un comentario dentro del código con las siguientes peticiones:

- Refactorizar la clase para **respetar principios de la programación orientada a objetos**.
- Implementar la forma Trapecio/Rectangulo.
- Agregar el idioma Italiano (o el deseado) al reporte.
- Se agradece la inclusión de nuevos tests unitarios para validar el comportamiento de la nueva funcionalidad agregada (los tests deben pasar correctamente al entregar la solución, incluso los actuales.)
- Una vez finalizado, hay que subir el código a un repo GIT y ofrecernos la URL para que podamos utilizar la nueva versión :).

Vista Previa

En una primera impresión, da la sensación de que implementar el patrón de diseño FactoryMethod sería la solución adecuada para conseguir el resultado deseado, pudiendo crear Formas Geométricas de distintas clases.

Esto nos permitiría que agregar nuevas Formas Geométricas sin tener que intervenir en el código previamente implementado.

Lo primero que se puede observar en el código preexistente es lo acoplada que está cada forma geométrica al código principal, teniendo que modificar el código en múltiples secciones en caso de querer adicionar alguna.

Segundo, se observa que el funcionamiento del método imprimir se encarga de contar y sumar las áreas y los perímetros de todas las formas similares, siendo esto el comportamiento sin equa non esperado.

En una segunda impresión nos damos cuenta que para implementar un nuevo idioma, habría que modificar todas las Figuras Geométricas previamente desarrolladas, para no hacer el código extremadamente complejo, pensamos en añadir a cada figura, un diccionario con los distintos idiomas a utilizar y una opción por defecto (Inglés). Otra manera sería usar el patrón Strategy y asignarle el comportamiento apropiado por cada idioma.

Desarrollo
Lo primero que hacemos es establecer una interfaz llamada IFormaGeométrica, que contiene la funcionalidad que van a implementar todas las Formas Geométricas que vayamos a crear, por lo que la misma contiene CalcularArea(), CalcularPerimetro() y ObtenerNombre() como métodos.

Segundo, separamos las distintas figuras preexistentes en distintas clases que van a incorporar la interfaz mencionada anteriormente, y asignamos a cada Figura Geométrica concreta el funcionamiento pertinente (implementamos la interfaz).

Al intentar implementar el comportamiento de TraducirForma() nos damos cuenta que una Forma Geométrica nunca va a tener la característica de ser plural, ya que es una única forma. Además decidimos que utilizar una clase aparte ayuda a no violar el Principio de Responsabilidad Simple del patrón de diseño SOLID.

Utilicé una interfaz, en vez de heredar la clase FormaGeometrica, ya que esto impediría la implementación de la forma Rectángulo, debido a que FormaGeometrica es entregada con la posibilidad de tener un único lado (Útil para Cuadrados, Circulos y Triángulos), mientras que un Rectángulo puede tener una base y una altura de distintas dimensiones.

Para el creador de reportes utilizamos el patrón de diseño Strategy, cosa de poder cambiar el comportamiento del mismo y que reproduzca los distintos idiomas deseados.

Finalmente, aprovechando la facilidad que nos ofrece haber implementado correctamente la Orientación a Objetos, podemos implementar el idioma Alemán y la figura Rectángulo, sin tener que modificar en absoluto la clase CreadorDeReportes.

Para crear una nueva figura simplemente creamos una nueva clase cuyo nombre será la figura a implementar, la cual va a poseer como interfaz IFormaGeometrica, y posteriormente, agregar las traducciones pertinentes a los distintos diccionarios dentro de cada Traductor.

Procedemos a correr los test unitarios y verificar que todo funciona como es esperado.

Una vez acontecido, procedemos a crear nuevos tests para la figura Rectángulo y el reporte en idioma Alemán. Confirmamos que todos los tests se ejecutan correctamente.

Procedemos a enviar el código al repositorio GIT.