
ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL
FACULTAD DE INGENIERÍA EN ELECTRICIDAD Y COMPUTACIÓN
DISEÑO DE SOFTWARE
PARADIGMA ORIENTADO A ASPECTOS (AOP)

Objetivos Específicos

1. Diseñar un producto de software aplicando principios de diseño orientado a objetos para que sea robusto, mantenible y escalable.
2. Emplear herramientas informáticas en el control de versiones para la generación de software de calidad en un entorno colaborativo.

Resultado de Aprendizaje

1. Funcionar efectivamente como miembro o líder de un equipo involucrado en actividades apropiadas para la disciplina del programa.
2. Habilidad para aplicar teoría de ciencias computacionales y fundamentos de desarrollo de software para producir soluciones basadas en computación.

Descripción

Un patrón de diseño describe un problema que ocurre de forma repetida en un cierto contexto y luego describe la parte medular de la solución al problema, de tal manera que esta solución pueda reutilizarse sin necesidad de estarla repensando [1]. El patrón de diseño *Adapter* funciona como un puente entre dos interfaces incompatibles [2]. El patrón de diseño *Singleton* asegura que exista solo una instancia de una clase y provee un punto de acceso global a esta única instancia [2]. Se requiere que trabaje en equipos de trabajo conformados por entre tres a cinco estudiantes y su código fuente en Java esté disponible en un repositorio público Git.

Especificaciones

Sección A

Presente el diseño e implementación del patrón de diseño Adapter utilizando el paradigma orientado a objetos. Además, ilustre el diseño e implementación del patrón de diseño Singleton utilizando el paradigma orientado a objetos. **[20%]**

Sección B

Muestre la implementación del patrón de diseño Adaptador utilizando el paradigma orientado a aspectos. Presente beneficios y/o limitaciones de esta implementación en comparación a la implementación puramente orientada a objetos. **[40%]**

Sección C

Muestre la implementación del patrón de diseño Singleton utilizando el paradigma orientado a aspectos. Presente beneficios y/o limitaciones de esta implementación en comparación a la implementación puramente orientada a objeto. **[40%]**

Entregables

1. Un **documento** en formato **pdf** que identifique los integrantes del equipo en la primera página y tenga un índice de contenido en la segunda. Se debe incluir imágenes claras del diseño y código fuente, según corresponda. Utilice diagramas UML para ilustrar su solución. Puede hacer uso de cualquier asunción razonable. Además, el documento debe incluir el URL de su repositorio Git.

Rúbrica de Calificación

Descripción	Valor
Sección A	
Diseño e implementación de Adapter	10
Diseño e implementación de Singleton	10
Sección B	
Implementación AOP de Adapter	30
Discusión de la implementación AOP	10
Sección C	
Implementación AOP de Singleton	30
Discusión de la implementación AOP	10
Total	100
No subir a Aula Virtual los entregables de acuerdo con lo especificado (máximo -50)	-10

Late Submission Policy

Delay (§)	Penalty (Ω)
1 hour or less	loss of 10%
1 to 12 hours	loss of 20%
12 to 48 hours	loss of 30%
Over 48 hours:	loss of 100%

(§) every clock hour counts including weekends or holidays

(Ω) automatic and non-negotiable penalty

Referencias

1. E. Gamma, Design Patterns
2. Refactoring Guru, <https://refactoring.guru/design-patterns>