

Instantly share code, notes, and snippets.

dfleta / **bicipalmaADT.md**

Created 18 minutes ago

☆ Star

<> **Code** - Revisions 1

 **bicipalmaADT.md**

Bicipalma ADTs

Además de reflexionar sobre cómo trasladar a tu código los conceptos tratados en el documento [Working classes](#):

<https://docs.google.com/document/d/1DE9xVc7D0cL7SwTN6HNLOk8PAklZculjR3EriXgQ-vQ/>

este ejercicio sirve como introducción a las sesiones sobre qué es programar y paradigmas de programación.

Consulta el código del proyecto

[Bicipalma](#)

<https://github.com/dfleta/bicipalma-streams-jacoco>

Consulta las clases `Anclajes` y `Anclaje`.

Elabora un fichero markdown, google docs o usando el formato que estimes oportuno (odt, word, PDF, etc.) para **contestar por escrito a las siguientes cuestiones**.

1. Razones para crear una clase

Vamos a reflexionar por qué en el diseño se ha optado por crear una clase `Anclaje` si ya existía la clase `Anclajes`.

Responde a las siguientes preguntas sobre la clase `Anclaje`:

1. ¿Modela un objeto del mundo real? ¿Cuál?
2. ¿Modela un ADT? ¿Cuál?
3. ¿Cuáles son las operaciones sobre este tipo de dato abstracto?
4. Describe el tipo de dato correspondiente a este ADT. ¿Que tipo de dato es? ¿Incluye algún otro tipo de dato abstracto? ¿Cuál?
5. ¿De qué modo reduce la complejidad del problema. Fíjate, p.e., en el método `ocuparAnclaje(int posicion, Movil bici)` de `Anclajes`.
6. ¿Aisla la complejidad? ¿Qué operaciones aísla?
7. ¿Qué detalles de implementación esconde al resto de las clases que componen la aplicación?
8. ¿Limita los efectos de los cambios? Piensa un ejemplo de cómo la clase `Anclaje` aísla del resto del código los cambios producidos en su lógica.
9. Indica un ejemplo de cómo facilita la reutilización del código, si es que encuentras alguno.
10. ¿Empaqueta operaciones relacionadas entre sí? ¿Cuáles? ¿La clase presenta una alta o baja cohesión?
11. ¿Consideras que la clase `Anclaje` ha surgido como resultado de una refactorización? ¿En qué crees que consistió esta refactorización?
12. ¿De qué modo la clase respeta o viola el principio SRP? Señala un dónde y cómo lo consigue o no lo consigue.
13. ¿De qué modo la clase respeta o viola el principio OCP? Señala un dónde y cómo lo consigue o no lo consigue.

2. Calidad de una clase

Responde a las siguientes preguntas sobre la clase `Anclajes` :

Abstraction

- ¿Cuál es el propósito central de la clase?
- ¿Está bien nombrada la clase y su nombre describe su propósito central?
- ¿La interfaz de la clase deja en claro cómo se debe usar la clase?
- ¿Es la interfaz de la clase lo suficientemente abstracta como para no tener que pensar en cómo se implementan sus servicios? ¿Puedes tratar la clase como una caja negra?
- ¿Los servicios de la clase son lo suficientemente completos como para que otras clases no tengan que interferir con sus datos internos?
- ¿Ha pensado en subdividir la clase en clases de componentes y la ha subdividido tanto como pudo?

Encapsulación

- ¿La clase minimiza la accesibilidad a sus miembros?
- ¿La clase evita exponer los datos de los miembros?
- ¿La clase oculta sus detalles de implementación a otras clases?
- ¿Evita la clase hacer suposiciones sobre sus usuarios, incluidas sus clases derivadas?
- ¿Es la clase independiente de otras clases? ¿Exhibe alto o bajo acoplamiento?