

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? Si ¿Cuál?

Un anclaje para fijar una bicicleta.

2. ¿Modela un ADT? ¿Cuál?

Si, con la información de si un anclaje está o no ocupado y con la información de la bicicleta que está estacionada en dicho anclaje.

3. ¿Cuáles son las operaciones sobre este tipo de dato abstracto?

Saber cual es el estado del anclaje, si está ocupado o no y cual es la bicicleta la cual ocupa dicho anclaje

4. Describe el tipo de dato correspondiente a este ADT. ¿Qué tipo de dato es? ¿Incluye algún otro tipo de dato abstracto? ¿Cuál?

Sería un booleano si está ocupado o está libre y en el caso de la bicicleta que lo ocupa sería de tipo integer o string (dependiendo de como se denominen a las bicicletas

5. ¿De qué modo reduce la complejidad del problema? Fíjate, p.e., en el método ocuparAnclaje(int posicion, Movil bici) de Anclajes.

Con la clase anclaje lo que se hace es que el resto del programa se despreocupe de las acciones propias de esta clase.

6. ¿Aísla la complejidad? ¿Qué operaciones aísla?

Si como ya dije en la respuesta anterior aísla al resto del programa de las acciones de anlar y retirar la bicicleta.

7. ¿Qué detalles de implementación esconde al resto de las clases que componen la aplicación?

Nos oculta los datos que están almacenados en la clase, si está o no ocupado y que bicicleta está anclada.

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.

Si puesto que al hacer cualquier cambio en la clase anclaje no va a afectar a ningún otro elemento del programa

9. Indica un ejemplo de cómo facilita la reutilización del código, si es que encuentras alguno.

Si se pretende hacer otra aplicación de las mismas características.

10. ¿Empaqueta operaciones relacionadas entre sí? ¿Cuáles? ¿La clase presenta una alta o baja cohesión?

Si, anclar y liberar es un booleano una depende de la otra, alta cohesión.

11. ¿Consideras que la clase Anclaje ha surgido como resultado de una refactorización? ¿En qué crees que consistió esta refactorización?

Si, en aislar y dar funcionalidad a la clase anclaje y liberar a la clase anclajes.

12. ¿De qué modo la clase respeta o viola el principio SRP? Señala un dónde y cómo lo consigue o no lo consigue.

Lo respeta puesto que esta clase ya está hecha para tener su propia responsabilidad de si está el anclaje ocupado o no.

13. ¿De qué modo la clase respeta o viola el principio OCP? Señala un dónde y cómo lo consigue o no lo consigue.

Lo respeta puesto que al ser una clase privada está cerrada a modificaciones externas a la clase.

2. Calidad de una clase

Responde a las siguientes preguntas sobre la clase Anclajes:

Abstracción

- ¿Cuál es el propósito central de la clase?

Gestionar los anclajes de la clase anclaje

- ¿Está bien nombrada la clase y su nombre describe su propósito central?

Si creo que está bien definida

- ¿La interfaz de la clase deja en claro cómo se debe usar la clase?

Opino que sí, te dice cual esta libre y cual ocupado y el número de anclajes que tiene la estación.

- ¿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?

Si puesto que tu como usuario puedes ver que anclaje está ocupado y cual está libre sin tener que saber cómo el programa realiza estas operaciones.

- ¿Los servicios de la clase son lo suficientemente completos como para que otras clases no tengan que interferir con sus datos internos?

Creo que sí.

- ¿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?

Creo que si

- ¿La clase evita exponer los datos de los miembros?

- Creo que si

- ¿La clase oculta sus detalles de implementación a otras clases?

Creo que si

- ¿Evita la clase hacer suposiciones sobre sus usuarios, incluidas sus clases derivadas?

Creo que si

- ¿Es la clase independiente de otras clases? ¿Exhibe alto o bajo acoplamiento?

Existe alto acoplamiento ya que **depende** de la clase anclaje que le indica cual está ocupado