

Principios de SOLID en el código de la suma asegurada:

1. Principio de responsabilidad Unica (SRP):

- a) Diccionario: Aquí la responsabilidad única es que la clase Diccionario es responsable de dividir cadenas en palabras, eliminar caracteres innecesarios y convertir todo a minúsculas.
- b) OrdenableIterativoAbstractClass: Aquí la responsabilidad única es la de intercambiar elementos de una lista durante la ordenación.
- c) Bubblesort: Aquí la responsabilidad única es la de implementar el algoritmo de ordenamiento burbuja.

2. Principio abierto/Cerrado (OCP):

- a) Diccionario: Esta cerrada para toda modificación pero abierta para toda extensión. Por ello se puede agregar funcionalidades nuevas que estén relacionadas con la manipulación de cadenas sin modificar la clase principal.
- b) OrdenableIterativoAbstractClass y BubbleSort: Estos son diseñados con la finalidad de que sean extendidos, por ello pueden crear otros algoritmos de ordenamiento que siguen la estructura definida por el “ordenableIterativoAbstractClass” el cual funge como un algoritmo rápido, sin modificar las clases existentes.

3. Principio de Segregación de la interfaz (ISP): No aplica en este caso ya que no hay interfaces como suelen estarlo en los lenguajes que soportan el concepto de forma explícita.

4. Principio de Sustitución de Liskov (LSP): Aquí la clase de “BubbleSort” es una subclase de “ordenableIterativoAbstractClass” sin cambiar el comportamiento correcto del programa.

5. Principio de inversión de Dependencias (DIP):

- a) diccionario: Depende de la clase “bubbleSort” a través de la instancia “bubble” para realizar la ordenación.
- b) BubbleSort: Aquí la dependencia está en la abstracción de “ordenableIterativoAbstractClass” en lugar de depender de implementaciones concretas.