

Principios de SOLID en el código de la suma asegurada:

1. Principio de responsabilidad (SRP):

- a) VerificarDatos (clase abstracta): Define los métodos abstractos para verificar los diversos datos relacionados con la persona.
- b) Persona (clase concreta): Implemente la lógica que verifica los datos y nos calcula la prima del seguro, así como genera el carnet final del programa.

2. Principio abierto/Cerrado (OCP): Permite la extensión (por medio de la herencia) sin modificar el código ya existente.

3. Principio de Segregación de la interfaz (ISP): El uso de una interfaz (VerificadorDatos) garantiza que las clases que la implementan solo implementen los métodos que necesitan para su propósito particular. Aunque en estos datos se puede optar por dividir las interfaces para poder cumplir de un mejor modo el principio.

4. Principio de Sustitución de Liskov (LSP): Aquí la clase de “persona” hereda la clase abstracta de “Verificardatos” y lo que hace es sobrescribir todo método abstracto, lo cual hace que cumpla el principio de sustitución de liskov.

5. Principio de inversión de Dependencias (DIP): Todo el código dependera de las abstracciones de “VerificadorDatos” y de sus metodos abstractos mas no de implemnetaciones concretas. Por ello hay una mayor flexibilidad y facilita la extension y por ende el remplazo de las implementaciones concretas.