

S.O.L.I.D: Os 5 princípios da POO

1. S – Single Responsibility Principle (Princípio da responsabilidade única)

Uma classe deve ter um, e somente um, motivo para mudar.

- Esse princípio declara que uma classe deve ser especializada em um único assunto e possuir apenas uma responsabilidade dentro do software, ou seja, a classe deve ter uma única tarefa ou ação para executar.

2. O – Open-Closed Principle (Princípio Aberto-Fechado)

Objetos ou entidades devem estar abertos para extensão, mas fechados para modificação.

- Esse princípio declara que quando novos comportamentos e recursos precisam ser adicionados no software, devemos estender e não alterar o código fonte original.

3. L – Liskov Substitution Principle (Princípio da substituição de Liskov)

Uma classe derivada deve ser substituível por sua classe base.

- Se para cada objeto o1 do tipo S há um objeto o2 do tipo T de forma que, para todos os programas P definidos em termos de T, o comportamento de P é inalterado quando o1 é substituído por o2 então S é um subtipo de T

- De forma mais simples se S é um subtipo de T, então os objetos do tipo T, em um programa, podem ser substituídos pelos objetos de tipo S sem que seja necessário alterar as propriedades deste programa.

4. I – Interface Segregation Principle (Princípio da Segregação da Interface)

Uma classe não deve ser forçada a implementar interfaces e métodos que não irão utilizar.

- Esse princípio basicamente diz que é melhor criar interfaces mais específicas ao invés de termos uma única interface genérica.

5. D – Dependency Inversion Principle (Princípio da inversão da dependência)

Dependa de abstrações e não de implementações.

- Módulos de alto nível não devem depender de módulos de baixo nível. Ambos devem depender da abstração.

- Abstrações não devem depender de detalhes. Detalhes devem depender de abstrações