

Andréia Tiveron - 27101112

## RESUMO - The SOLID Principles of OO design

- Inicialmente, o capítulo recaptula a ideia de que abordagem rígida de has-a e is-a talvez não seja a melhor, por isso, foca em reduzir a dependência entre as classes.
- Diante de problemas como: rigidez, fragilidade e imobilidade nascem os princípios SOLID, na tentativa de tornar códigos mais flexíveis, de fácil manutenção e evolução.
- Os cinco princípios SOLID são:

### 1) SPr - Single Responsibility Principle

Cada classe deve conter UM objetivo/responsabilidade bem definida.

"Uma classe deve ter um, e apenas um, motivo para mudar - uma única responsabilidade".

### 2) OCP - Open/Closed Principle

Deve-se poder extender o comportamento de uma classe sem modificá-la.

"Você deve extender seu código por meio de subclasses e a classe original não precisa ser alterada."

### 3) LSP - Liskov Substitution Principle

Deve haver a possibilidade de substituir qualquer instância da classe mãe com uma instância da classe filha.

"Se a classe mãe pode fazer algo, a classe filha deve poder também".

#### 4) IPS - Interface Segregation Principle

É melhor ter várias pequenas interfaces do que poucas grandes.

No exemplo citado no capítulo, foi preferível usar composição em detrimento de herança.

#### 5) DIP - Dependency Inversion Principle

O código deve depender de abstrações.

O objetivo é adiar o máximo possível a criação (por meio de "new") de um objeto concreto, como no main.