

Nome: Marcos Lucas da Silva (RA: 140782)

Prof. Otavio Lemos

UC: Programação Orientada a Objetos - 2022/2

Aula 7 - Princípio da Inversão de Dependência (DIP)

No paradigma da programação orientada a objetos existem alguns padrões que servem para guiar programadores a desenvolverem códigos que sejam mais compreensíveis a outros programadores. Com base nessa ideia, Robert C. Martin introduziu 5 princípios que facilitam a refatoração dos códigos. Tais princípios foram denominados S.O.L.I.D:

- Single Responsibility Principle (SRP)
- Open/Closed Principle (OCP)
- Liskov Substitution Principle (LSP)
- Interface Segregation Principle (ISP)
- Dependency Inversion Principle (DIP)

O último deles é o Princípio da Inversão de Dependência (DIP). Um código de um programa pode apresentar módulos ou rotinas de baixo e alto níveis. O DIP define que um nível não precisa depender de outro nível. A ideia é estabelecer a comunicação entre os níveis através de interfaces (abstrações). Fazendo uma analogia, por exemplo, pode-se pensar que a interface é uma tomada e os aparelhos são coisas de alto e baixo níveis. A tomada é uma interface comum, onde diferentes aparelhos podem se conectar.

HLM(alto nível) → I(Interface) ← LLM(baixo nível)

Abaixo tem-se um exemplo de uma aplicação do DIP em um programa de “cozinhar”, feito com a linguagem python:

```
1  from abc import ABC, abstractmethod
2  #A biblioteca ABC é usada para definir uma classe abstrata
3  #Classe abstrata (interface genérica)
4  class Bakable(ABC):
5      @abstractmethod
6      def bake(self):
7          pass
8
9  class Bread(Bakable):
10     def bake(self):
11         print("Baking bread")
12
13     class Cookie(Bakable):
14         def bake(self):
15             print("Baking cookie")
16
17     #Bread e Cookie dependem da interface abstrata Bakable!
18     def cook(bakable: Bakable):
19         bakable.bake()
20
21     cookie = Cookie()
22     bread = Bread()
23     cook(cookie)
24     cook(bread)
```

Saída:

```
Baking cookie
Baking bread
```