Métodos getters() e setters()

● Definição

Esses métodos fazem exatamente o que seus nomes implicam. Um método getter é um método que “obtém” o valor de uma variável, enquanto um método setter é um método que “define” o valor de uma variável. (BUYYA; SELVI; CHU, 2009, pg. 180, tradução nossa).

Quadro 1 - Definição dos métodos getter() e setter()

| **Método** | **Definição** |
| --- | --- |
| getter() | Getters são utilizados para acessar valores de variáveis apenas para visualização. Você deve criar um método getter para cada variável privada. Dependendo do nível de acesso que você deseja dar à variável, você pode definir o modificador de acesso do seu método getter. |
| setter() | Setters são os métodos que você utiliza para modificar os valores das variáveis. Assim como os getters, você deve criar setters para cada variável. Setters podem substituir as declarações que você utilizou no método construtor. |

Fonte: BAESENS; BACKIEL; BROUCKE (2015)

● Em que contexto esses métodos devem existir? Por que devem ser criados?

Quando há muitos itens para atualizar ou acessar, geralmente é melhor definir métodos específicos para cada propósito. Esses métodos são informalmente chamados de accessors ou métodos getter e setter. (BUYYA; SELVI; CHU, 2009, pg. 180, tradução nossa).

Um dos objetivos básicos da programação orientada a objetos é esconder os detalhes da implementação de uma classe dentro da classe enquanto cuidadosamente controlando quais aspectos da classe são expostos para o mundo de fora. Isso é muitas vezes referido como encapsulamento. [...] Um jeito de fazer isso é evitar criar campos públicos. Em vez disso, deixe seus campos privados. Então, seletivamente conceda acesso para a informação que esses campos contém adicionando para a classe métodos especiais chamados accessors. Há dois tipos de accessors. O get accessor (também chamado de getter) é um método que recupera um valor do campo, enquanto que o set accessor (setter) é um método que define um valor do campo. (LOWE, 2020, pg. 261-262, tradução nossa).

Referências

BAESENS, Bart; BACKIEL, Aimée; BROUCKE, Seppe Vanden. **Beginning Java Programming**: the object-oriented approach. Indianapolis: John Wiley & Sons, 2015.

BUYYA, Rajkumar; SELVI, S Thamarai; CHU, Xingchen. **Object-Oriented Programming with Java**: essentials and applications. New Delhi: Tata McGraw Hill, 2009.

LOWE, Doug. **Java All-in-One For Dummies**. 6. ed. New Jersey: John Wiley & Sons, 2020.