Programação orientada a objetos

Nelson Seixas dos Santos

Faculdade de Ciências Econômicas
Universidade Federal do Rio Grande do Sul

May 9, 2019

Sumário

Seção - Introdução

Seção - Paradigmas de programação

Paradigmas de programação: definição

Um paradigma de programação é um conjunto de regras definidos por uma linguagem de programação para organização do código fonte de um programa, tendo em vista as funcionalidades oferecidas pela linguagem.

Seção - Programação orientada a objetos

Programação orientada a objetos: definição

Programação orientada a objetos é o paradigma de programação cujas regras objetivam fazer com que o código escrito seja uma representação (modelo) computacional dos objetos reais referentes ao problema a ser resolvido, ou seja, o programa é escrito de forma a simular do mundo real.

Programação orientada a objetos: histórico

O paradigma de programação orientada a objetos foi desenvolvido a partir do final da década de 1950 e início da década de 1960 em trabalhos inicialmente no MIT, mas a primeira linguagem orientada a objetos é chamada Simula 67, tendo sido desenvolvida no Centro Norueguês de Computação por Kristen Nygaard e Ole-Johan Dahl cujo intuito inicial é facilitar a realização de simulações de Monte Carlo.

Modelagem do mundo real e simulação

A modelagem é um processo de abstração dos objetos do mundo real para a memória do computador por meio da criação de variáveis (que representarão características ou atributos do objeto) e de funções (que representam as ação capazes de serem realizadas pelo objeto ou sobre o objeto).

Formulação matemática do modelo abstrato computacional

O domínio do problema no mundo real é modelado como sendo um conjunto universo (também chamado em matemática de classe) e cada um dos elementos do conjunto é chamado de objeto.
Por exemplo, podemos falar no conjunto (classe) de todos os pratos e cada prato em particular é um objeto da classe.

Classe e objetos

Como as classes são apenas conjuntos de objetos e cada objeto do mundo real se caracteriza por ter propriedades (ou atributos) e funções (aqui chamadas de métodos), então para definirmos o conjunto de todos os objetos de determinado tipo (isto é, a classe), precisamos definir os atributos e os métodos.

Definindo uma classe em Python

```
class Humano:
    def __init__(self):
        self.peso = 0
        self.altura = 0
        self.idade = 0
        self.sexo = "feminino"
        self.nome = ""
```

Definindo uma classe em Python II

Métodos construtores

A classe construída anteriormente tem apenas um método. Este método é chamado método construtor.

Métodos construtores são usados em orientação a objetos para instruir o compilador/interpretador a forma como se constróem os objetos.

Definindo uma classe em Python III

Em Python, embora estes métodos não sejam obrigatórios, eles servem para se definir os atributos que todos os objetos da classe terão. É boa prática de programação Python inicializar as classes com métodos construtores

Observe que os atributos da classe são variáveis e estas tem de ser inicializadas. No entanto, cada objeto poderá guardar um valor diferente para cada variável, isto é, cada objeto tem seus próprios atributos.

Exemplo 1: classe televisão

Confira (Menezes, 2014, p. 217–221)

Exemplo 2: modelo de apreçamento de ativos de capital

A equação fundamental do modelo de apreçamento de ativos de capital devido a Sharpe (1964):

$$E\left[R_i\right] = R_f + \beta_i . E_t\left[R_m - R_f\right] \quad (1)$$

Onde:

- R_i - retorno da ação i;
- R_m - retorno da carteira de mercado;
- R_f - retorno do ativo livre de risco;
- $R_m - R_f$ - prêmio de risco de mercado, e
- $R_i - R_f$ - prêmio de risco da ação i.

Exemplo 2: modelo de apreçamento de ativos de capital (cont.)

A equação fundamental do modelo pode ser reescrita como segue:

$$E[R_i] = R_f + \beta_i . E_t[R_m - R_f] \quad (2)$$

$$E[R_i] - R_f = \beta_i . E_t[R_m - R_f] \quad (3)$$

$$E[R_i - R_f] = \beta_i . E_t[R_m - R_f] \quad (4)$$

Ou seja, o modelo estabelece que há um relação linear entre prêmio de risco de uma ação e o prêmio de risco de mercado.

Exemplo 2: modelo de apreçamento de ativos de capital (cont.)

O modelo é equivalente a :

$$[R_i - R_f] = \alpha + \beta_i . [R_m - R_f] + \epsilon_i \quad (5)$$

Nestas condições, o parâmetro β - que é a sensibilidade do prêmio de risco sistemático do ativo a variações no prêmio de risco do mercado pode ser estimado por mínimos quadrados ordinários e seu teste consiste em realizar o teste t no parâmetro α e β. O modelo é aceito se aceita-se $\alpha = 0$ e $\beta \neq 0$.

Exemplo 2: modelo de apreçamento de ativos de capital (cont.)

Construa a classe que formaliza computacionalmente o CAPM

Referências

Nilo Ney Coutinho Menezes. *Introdução à programação com Python*. Novatec Editora, segunda edition, 2014. ISBN 978-85-7522-408-3.

William F Sharpe. Capital asset prices: A theory of market equilibrium under conditions of risk. *The journal of finance*, 19 (3):425–442, 1964.