

PARADIGMAS DE LINGUAGENS DE PROGRAMAÇÃO EM PYTHON - ARA0066

Semana Aula: 13

PARADIGMA ORIENTADO A OBJETO

Tema

6. PARADIGMAS: ESTRUTURADO, ORIENTADO A OBJETOS, FUNCIONAL E LÓGICO (CRÉDITO DIGITAL)

Objetivos

Apresentar o paradigma orientado a objetos e sua importância para a modelagem de sistemas

Tópicos

6.1 PARADIGMA ORIENTADO A OBJETO

Procedimentos de Ensino-Aprendizagem

Nesta aula, estaremos conectados com o conteúdo digital. O aluno explora e estuda, previamente, o conteúdo digital disponível em seu ambiente virtual. Durante a aula, este conteúdo será discutido em sala em atividade mediada pelo professor, detalhada abaixo.

Situação-problema:

Os paradigmas de programação são um modelo, uma metodologia de programação. Você já conhece o paradigma conhecido como programação estruturada que utiliza subprogramas e três estruturas básicas: sequência, seleção e repetição. Seus códigos ficam em um mesmo bloco, sendo mais difícil e demorado realizar alterações pois é necessário verificar se outro trecho de código depende dele. Vamos supor que um determinado software possui uma função de conexão com um banco de dados, a qual é utilizada em diversos pontos dele e que é utilizada de forma corriqueira em outros softwares. Como você poderia reutilizar esse código? De que forma poderia ser disponibilizado?

Metodologia:

Aula inicia com um rápido debate sobre o entendimento do paradigma orientado a objetos em comparação ao paradigma estruturado. Em seguida o professor deve propor uma atividade prática de implementação de uma classe utilizando a linguagem de programação Python.

Classe Pessoa: Crie uma classe que modele uma pessoa:

- a. Atributos: nome, idade, peso e altura
- b. Métodos: Envelhecer, engordar, emagrecer, crescer.

Obs: Por padrão, a cada ano que nossa pessoa envelhece, sendo a idade dela menor que 21 anos, ela deve crescer 0,5 cm.

Após a implementação da classe o professor deve indagar aos alunos como eles iriam implementar em um paradigma estruturado essa mesma classe. Nesse ponto o professor deve fazer um paralelo entre funções e métodos, variáveis e objetos, chamadas a funções e mensagens.

Atividade verificadora de aprendizagem:

Classe Bichinho Virtual: Crie uma classe que modele um Tamagoshi (Bichinho Eletrônico): (Exercício 7 da lista de exercícios sobre Classes do site Python Brasil [3]).

- a. Atributos: Nome, Fome, Saúde e Idade
- b. Métodos: Alterar Nome, Fome, Saúde e Idade; Retornar Nome, Fome, Saúde e Idade e imprimir (deve apresentar os valores de todos os atributos)

Após a criação da classe, instancie dois Tamagoshis e altere seus atributos e depois os imprima.

Como atividades opcionais o aluno pode implementar os demais exercícios disponíveis no site Python Brasil [3]

Esta atividade computará 2,0 pontos para a AV2. O docente deve acompanhar o desenvolvimento dos alunos, dirimindo dúvidas, e zelando pela curretude das soluções. Recomenda-se o uso do SAVA ou Teams para a realização das entregas.

Recursos Didáticos

Laboratório de Informática com Internet com navegador Web instalado, equipado com quadro branco, projetor multimídia, acervo bibliográfico no ambiente virtual.

Leitura Específica

[1] SEBESTA, Robert W. Conceitos de Linguagens de Programação. 11. edição. Porto Alegre: Bookman, 2018., Capítulo 12 (Suporte para programação orientada a objetos), páginas 488 a 512, Disponível em:
<https://integrada.minhabiblioteca.com.br/#!/books/9788582604694/>

[2] Documentação do Python "Python e Programação Orientada a Objeto". Disponível em <https://wiki.python.org.br/ProgramacaoOrientadaObjetoPython>

[3] Exercício 7 da lista de exercícios sobre Classes do site Python Brasil, disponível em <https://wiki.python.org.br/ExerciciosClasses>

Aprenda +

[4] Vídeo "Programação orientada a objetos". Disponível em: <https://www.youtube.com/watch?v=IbXsrHGhBAU> (Ative a legenda e a tradução automática para o português)

Atividade Autônoma Aura:

Questão 1: (FUNCERN - IF-RN, 2017)(Adaptada)

Analise o seguinte código escrito em Python, que define a estrutura da classe ContaBancaria:

```
class ContaBancaria:

    num_contas = 0

    def __init__(self, agencia, numero, saldo):

        self.agencia=agencia

        self.numero=numero

        self.saldo=saldo

    ContaBancaria.num_contas +=1

    def __del__(self):

        ContaBancaria.num_contas -=1

    def depositar (self,valor):

        self.saldo = self.saldo + valor

    def sacar (self,valor):

        self.saldo = self.saldo - valor

    def consultarSaldo(self):
```

```
return self.saldo
```

Sobre a classe acima e as regras de programação orientada a objetos em Python,

- a) criação de objetos chama primeiro o método `__init__()` e, em seguida, o `__new__()`.
- b) palavra `self` deve ser fornecida como argumento em todos os métodos de instâncias.
- c) variável `num_contas` é encapsulada e individual para cada instância da classe.
- d) palavra `@static` escrita antes da definição do método `sacar()` torna o método estático.
- e) estamos diante de situação onde há herança

Questão 2: (SUGEP-UFRPE, 2016) Considere as afirmações abaixo, sobre os paradigmas de linguagens de programação.

- 1) As linguagens de programação Python, Ruby, C#, Cython e Lua são multiparadigmáticas e podem ser classificadas, pelo menos, nos paradigmas Orientado a Objetos, Funcional e Imperativo.
- 2) As linguagens de programação Object-Pascal (Delphi), Python, C++ e Java, embora deem suporte à Orientação a Objetos (OO), não são completamente orientadas a objetos.
- 3) As linguagens de programação Smalltalk e Ruby são completamente orientadas a objetos, uma vez que todo valor de dados é um objeto e todas as operações são vias chamadas de métodos.

Está(ão)correta(s):

- a) 1, 2 e 3.
- b) 2 e 3, apenas.
- c) 1 e 2, apenas.
- d) 1 e 3, apenas.
- e) 3, apenas.