

1) Selecione a alternativa CORRETA a respeito do encapsulamento:

- A) Tem por objetivo o reaproveitamento de código, reutilizando a implementação dos métodos de uma classe em suas subclasses;
- B) Deve ser implementado em Python com as palavras reservadas public, protected e private, que alteram a visibilidade dos atributos;
- C) Limita-se a alteração de visibilidade dos atributos de um objeto, sem permitir a execução de qualquer trecho de código antes ou depois da alteração do valor do atributo;
- D) Deve ser implementado em Python com os decoradores @make_getter e @make_setter;
- E) É usado para ocultar atributos e métodos que não devem estar acessíveis externamente, possibilitando a inclusão de regras de negócio e validações antes e/ou depois da alteração de um atributo ou execução de um outro método interno;

2) Com os seus conhecimentos sobre o paradigma de programação orientada a objetos, marque a alternativa FALSA:

- A) O processo de abstração significa escolher as qualidades em comum de uma entidade do mundo real que sejam relevantes para a situação que desejamos modelar
- B) A programação orientada a objetos permite proteger os dados da aplicação (através dos atributos privados), mas não permite o reaproveitamento de código, pois este acaba ficando maior quando codificamos vários métodos para solucionar um problema
- C) O encapsulamento busca esconder partes da classe do restante da aplicação. Em resumo, seu objetivo é proteger os dados do objeto e esconder o funcionamento interno do objeto para o restante da aplicação
- D) O polimorfismo refere-se a funções ou métodos com o mesmo nome, mas com comportamentos diferentes
- E) Herança é uma das várias formas que podemos reutilizar código em POO

3) Considere as 3 estruturas a seguir: uma lista **x=["alface", "coentro", "cebolinha"]**, uma tupla **y=("alface", "coentro", "cebolinha")**, e um dicionário **z={"alface":5.00, "coentro":2.50, "cebolinha":4.00}**. Qual dos comandos a seguir NÃO é permitido em Python?

- A) y.append("batata")
- B) x.append("batata")
- C) x.pop(1)
- D) w = y + ("batata",)
- E) z["batata"] = 7.00

4) Selecione a alternativa CORRETA a respeito da criação de novas exceções em Python:

- A) É feita criando-se uma nova classe que deve herdar da classe Exception (ou de outra classe que tenha Exception como classe base). É possível sobrescrever métodos da classe base caso seja necessário, mas não é obrigatório. Pode-se deixar a nova classe vazia (apenas com o comando pass);
- B) É feita criando-se uma nova classe que deve herdar da classe Exception, sendo obrigatório definir nesta nova classe o método __init__;
- C) Não é possível criar novas exceções personalizadas em Python;
- D) É feita criando-se uma nova classe que deve herdar da classe PythonError, (ou de outra classe que tenha PythonError como classe base). É possível sobrescrever métodos da classe base caso seja necessário, mas não é obrigatório. Pode-se deixar a nova classe vazia (apenas com o comando pass);
- E) É feita criando-se uma nova função que deve ser decorada com o decorador @new_exception;

5) Leia as seguintes afirmações a respeito das listas em Python:

- São imutáveis
- São heterogêneas;
- São uma estrutura de dados linear e ordenada;
- Possuem um limite máximo de 65536 itens;
- É possível verificar se um item pertence à lista usando o operador in;

São VERDADEIRAS:

- A) I, II e III;
- B) II, III e IV;
- C) II, III e V;
- D) I e V;
- E) III, IV e V;

6) Selecione a alternativa FALSA a respeito do conceito de polimorfismo e sua aplicação em Python:

- A) Em Python, um exemplo de polimorfismo de sobrecarga é a função print(), que dependendo dos parâmetros que recebe (diferentes assinaturas) exibe um comportamento diferente;
- B) Para sobrescrever um método herdado de outra classe, devemos preceder sua definição com o decorador @overwrite;
- C) O polimorfismo ocorre quando há um mesmo método com comportamento diferente quando em objetos de tipos diferentes (sobrescrita) ou quando executado com assinaturas diferentes (sobrecarga);
- D) Ao sobrescrever um método em uma classe filha, é possível acessar e executar internamente o respectivo método da classe mãe através da função embutida super();
- E) Para sobrescrever um método herdado de outra classe, basta defini-lo novamente na classe filha, pois o Python apenas irá procurar e executar o método da classe mãe caso este não exista na classe filha;

7) Selecione a alternativa que descreve CORRETAMENTE o método __init__ do Python.

- A) É um método que faz parte do conjunto de métodos especiais do Python, e sua função é limpar o objeto da memória quando este não for mais necessário para a execução do código;
- B) É uma referência para o objeto criado pelo Python no momento em que este é instanciado, e seu valor pode ser conferido através da função embutida id();
- C) É um método que faz parte do conjunto de métodos especiais do Python, e sua função é inicializar o objeto no momento em que este é instanciado caso o programador escolha executá-lo manualmente. Caso não seja executado manualmente, é ignorado pelo Python;
- D) É um atributo opcional usado para guardar a documentação a respeito de um objeto ou classe em Python;
- E) É um método que faz parte do conjunto de métodos especiais do Python, e sua função é inicializar o objeto no momento em que este é instanciado, sendo chamado automaticamente pelo Python;

8) A respeito do parâmetro self nos métodos de uma classe em Python, qual afirmativa é correta?

- A) É usado somente para acessar e criar atributos dentro da classe, não podendo ser usado para acessar/chamar outros métodos dentro da própria classe
- B) Sua função é representar uma instância da classe, permitindo que sejam acessados atributos e métodos dela
- C) Só é utilizado para acessar atributos e métodos de superclasses (herança), permitindo inclusive acessar atributos privados das superclasses
- D) É um parâmetro opcional, ou seja, podemos construir métodos e acessar atributos sem a sua necessidade
- E) Pode aparecer em qualquer posição na lista de parâmetros de um método

9) Selecione a alternativa que MELHOR descreve o que é o espaço de nomes de um módulo:

- A) É o escopo global do módulo, no qual há uma tabela relacionando os identificadores (variáveis, classes, funções, etc.) aos objetos criados na memória;
- B) É o escopo local da função ou classe, criada dentro de um arquivo Python;
- C) É a lista dos caminhos de busca do Python quando executamos um comando de importação;
- D) É o escopo intermediário do módulo, criado temporariamente durante o processo de inicialização do interpretador do Python;
- E) É a lista de ordem de resolução de um método, quando chamado a partir de um objeto qualquer do Python;

10) Sobre a função super(), marque a alternativa correta:

- A) Se desejamos chamar um método hipotético chamado imprime_dados() que está implementado na superclasse, usamos a sintaxe: super.__imprime_dados()
- B) Permite chamar o construtor da superclasse, usando a sintaxe: super().init(...), onde ... é a lista de parâmetros do construtor
- C) Pode sempre ser substituída pelo parâmetro self
- D) Permite chamar qualquer método acima na hierarquia de classes, exceto o método construtor
- E) É usada para acessar métodos de classes que estão acima na hierarquia de classes, quando trabalhamos com herança de classes