

Iniciado em	Tuesday, 18 Apr 2023, 09:43
Estado	Finalizada
Concluída em	Tuesday, 18 Apr 2023, 12:23
Tempo empregado	2 horas 40 minutos
Avaliar	3,50 de um máximo de 10,00(35%)

Questão 1

Completo

Atingiu 1,50 de 3,00

Explique o que é polimorfismo e a sua importância na Programação Orientada a Objetos. Explique como comportamento polimórfico pode ser obtido através da interação entre tipagem dinâmica e o mecanismo de herança.

O polimorfismo é uma ferramenta utilizada na programação orientada a objetos que permite adicionar um novo grau de abstração sob o programa, possibilitando utilizar diferentes tipos de objetos (classes mais específicas provindas de uma classe pai) sem me preocupar com o funcionamento específico de cada um. Por exemplo, se quero desenhar na tela diversas instancias de formas geométricas (todas como classe pai classe abstrata Formas) , posso chamar unicamente a função desenhar(). Ainda que cada uma das instâncias seja a instância de uma classe específica (Quadrado, Triangulo etc) e possua sua forma diferente de desenhar, como essas classes são derivadas da classe Formas, a chamada da função desenhar() irá abstrair essas diferenças, aplicando para cada instancia a sua função específica.

Comentário:

Polimorfismo, como o próprio nome sugere, significa que uma variável ou método pode ter múltiplas formas. Isso facilita o desenvolvimento de código reusável e extensível pois permite que objetos e métodos sejam usados sem que se saiba qual exatamente é a implementação ou objeto específico sendo usado. Isto é muito útil pois permite que módulos de alto nível possam depender de abstrações e não de implementações específicas/concretas (especializações). Comportamento polimórfico pode ser obtido através da sobreescrita (*override*) de métodos por meio da especialização de classes base/pai (herança), para isso é necessário o uso do mecanismo de amarração dinâmica (late binding).

Questão 2

Incorreto

Atingiu 0,00 de 1,00

Julgue a veracidade das afirmações abaixo:

- Na programação orientada a objetos sistemas são construídos através de coleções de objetos que encapsulam dados e operações que manipulam estes dados, buscando aumentar o acoplamento entre objetos e reduzir a sua coesão.
- Na programação procedural, funções e procedimentos alteram os dados globais de um programa. Quanto maior a dependência entre tais funções e procedimentos, mais simples será sua manutenção e também a adição de novas funcionalidades.
- Uma classe específica as propriedades e operações comuns de um determinado grupo de objetos de um programa.

Escolha uma opção:

- ☐ 1. Apenas a III está correta.
- ☐ 2. Apenas I e III estão corretas.
- ☐ 3. Apenas II e III estão corretas.
- ☒ 4. Nenhuma das afirmações está correta. ✖
- ☐ 5. Apenas a I está correta.

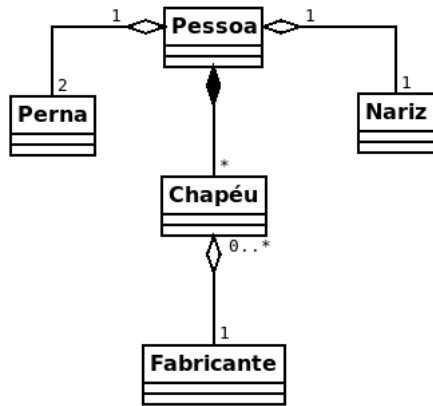
A resposta correta é: Apenas a III está correta.

Questão 3

Completo

Atingiu 0,00 de 3,00

Em relação ao diagrama de classes abaixo, identifique erros, explique-os e sugira correções.



A classe perna, nem a classe nariz nem a classe chapéu deverá receber em seu construtor um atributo pessoa. Logo, o número "1" ao lado da ligação está errado

Comentário:

Os três pontos principais são:

- Relação entre Chapéu e Pessoa deveria ser agregação.
- Relação entre Perna e Pessoa, e Nariz e Pessoa são de composição. Estes objetos são exclusivos de Pessoa neste Diagrama.
- A relação de agregação entre Chapéu e Fabricante está invertida. Um fabricante 'agrega' os diversos chapéus que fabrica, e não o contrário.

Questão 4

Correto

Atingiu 1,00 de 1,00

Em programação orientada a objetos, a herança nos permite definir atributos e operações em uma classe base, evitando a replicação de código, e posteriormente utilizar classes , para realizar a criação de novos atributos e operações nestas novas classes. Assim, este mecanismo nos permite criar uma hierarquia de classes onde o nível mais alto apresenta um mais nível de abstração e portanto possui um grau de reusabilidade.

Questão 5

Incorreto

Atingiu 0,00 de 1,00

Julgue a veracidade das afirmações abaixo:

- I - Listas encadeadas são áreas contíguas de memória.
- II - Dicionários em Python requerem que as chaves utilizadas sejam todas de um mesmo tipo.
- III - É possível utilizar uma lista como chave de um dicionário em Python.

Escolha uma opção:

- ☐ 1. Apenas a III está correta.
- ☒ 2. Apenas a I está correta. ✗
- ☐ 3. Nenhuma das afirmações está correta.
- ☐ 4. Apenas II e III estão corretas.
- ☐ 5. Apenas I e III estão corretas.

A resposta correta é: Nenhuma das afirmações está correta.

Questão 6

Correto

Atingiu 1,00 de 1,00

A orientação a objetos é um modelo de análise, projeto e programação de sistemas de software baseado na composição e interação entre diversas unidades de software chamadas de objetos.

Com relação a conceitos de orientação a objetos, julgue as afirmações abaixo:

I - As variáveis ou métodos declarados como privados só são acessíveis a métodos da classe em que são declarados.

II - Uma instância de uma classe abstrata herda atributos e métodos de sua superclasse direta.

III - O polimorfismo permite substituir a lógica condicional múltipla (lógica switch, ou if-elif-else em Python).

Escolha uma opção:

- ☐ a. Apenas II e III estão corretas.
- ☒ b. Apenas I e III estão corretas. ✓
- ☐ c. Apenas a II está correta.
- ☐ d. Apenas a I está correta.
- ☐ e. Apenas a III está correta.
- ☐ f. Nenhuma das afirmações está correta.
- ☐ g. Apenas I e II estão corretas.

Sua resposta está correta.

A resposta correta é: Apenas I e III estão corretas.

◀ Atividade do encontro síncrono: Integração e Evolução do Sistema de Chatbots

Seguir para...



PROVA 1 - parte prática - 18/04 - das 9h45 às 12h25 ►