

Uma classe representa um molde para criar objetos, definindo seus atributos (propriedades) e métodos (comportamentos). Em **UML**, uma classe é modelada como um retângulo com três divisões: o nome da classe, atributos e métodos.

Programação Orientada a Objetos com a Linguagem JAVA

Fábio Pereira Botelho

M.Sc. Ciência da Computação UFPE/2004
botelho@cnps.embrapa.br

Anápolis – GO, Agosto de 2024

Exercício de Revisão

Questões para a Avaliação I

- 1. Confronte o Paradigma da Orientação a Objetos (OO) com o clássico Paradigma Estruturado (Procedural).
-
- R.
- No paradigma estruturado, conceitos complexos do mundo real são abordados como estruturas de dados e as funções representam as operações que podem ser efetuadas nos dados. No modelo de objetos, a estrutura de dados (propriedades) e as operações (métodos) que podem alterar os dados são reunidas em uma única construção denominada classe.
-
- O modelo procedural guarda funções e atributos (propriedades) em separado, enquanto que o OO os guarda juntos.

Exercício de Revisão

Questões para a Avaliação I

- 1. Confronte o Paradigma da Orientação a Objetos (OO) com o clássico Paradigma Estruturado (Procedural).
-
- R.
- A classe possui a vantagem da proximidade física dos dados e das funções que pertencem ao mesmo conceito do mundo real, o que torna mais fácil modificar a implementação.
-
- No modelo de objetos, a classe é um tipo que pode ser usada para criar objetos (instâncias). É possível nas instâncias, usar os métodos definidos para a classe.

Exercício de Revisão

Questões para a Avaliação I

2. Como a Unified Modeling Language (UML) pode auxiliar no desenvolvimento de software orientado a objetos ?

R. Através de um conjunto de diagramas que auxiliam o projeto do software segundo a abordagem OO, através de uma representação gráfica tanto das características estáticas quanto dinâmicas do sistema em desenvolvimento.

Exercício de Revisão

Questões para a Avaliação I

3. O que representa uma classe ? Como é modelada em UML e como é implementada em JAVA ?

R. .

Uma classe representa um molde para criar objetos, definindo seus atributos (propriedades) e métodos (comportamentos). Em **UML**, uma classe é modelada como um retângulo com três divisões: o nome da classe, atributos e métodos.

Exercício de Revisão

Questões para a Avaliação I

Acerca das Definições que envolvem o paradigma OO, responda as questões seguintes

4. O que representa um objeto ? Como é definida a criação de um objeto em JAVA ?

R. .

Um objeto é uma instância de uma classe, representando uma entidade concreta com atributos e comportamentos definidos pela classe.

Exercício de Revisão

Questões para a Avaliação I

5. O que é instância ? Como um objeto é instanciado em JAVA ?

R. .

Instância é a criação de um objeto a partir de uma classe, ou seja, quando um objeto é criado e passa a existir na memória.

Exercício de Revisão

Questões para a Avaliação I

Acerca das Definições que envolvem o paradigma OO, responda as questões seguintes

6. Defina atributo e diga qual a diferença em relação a uma propriedade. Como os atributos são definidos para classes, métodos e parâmetros de métodos em UML e em JAVA.

R. .

Um atributo é uma característica ou variável que pertence a uma classe ou objeto. A diferença entre atributo e propriedade é que **atributo** refere-se às variáveis internas de uma classe, enquanto **propriedade** é um conceito mais amplo, que pode incluir atributos com métodos de acesso (getters e setters).

- **Em UML**, os atributos são representados na segunda divisão do retângulo da classe, listando o nome, tipo e visibilidade (+, -, #).
- **Em Java**, os atributos são definidos dentro da classe, com modificadores de acesso (como private ou public):

Exercício de Revisão

Questões para a Avaliação I

7. O que é um método ? Qual a diferença em relação ao conceito de operação ? Como um método é definido para um classe em UML e em JAVA ?

R. .

Um método é uma função definida dentro de uma classe que descreve o comportamento de seus objetos. A diferença entre **método** e **operação** é que operação é o conceito geral de ação realizada por uma classe, enquanto o método é a implementação específica dessa operação.

- **Em UML**, os métodos são representados na terceira divisão do retângulo da classe, com nome, tipo de retorno, parâmetros e visibilidade (+, -, #).

Exercício de Revisão

Questões para a Avaliação I

Acerca das Definições que envolvem o paradigma OO, responda as questões seguintes

8. O que representa uma mensagem ?

R. . Uma mensagem representa a ****comunicação**** entre objetos em programação orientada a objetos (OO). Quando um objeto deseja que outro execute um de seus métodos, ele envia uma mensagem, que consiste na chamada do método com os parâmetros necessários.

Exercício de Revisão

Questões para a Avaliação I

9. O que é herança e quais as suas características ? Como a herança é representada em UML e em JAVA ?

R. . Herança é um mecanismo que permite a uma classe (subclasse) herdar atributos e métodos de outra classe (superclasse), promovendo reuso de código e hierarquia.

- . **Características:** reuso, especialização, e polimorfismo.
- . **Em UML**, herança é representada por uma seta com linha contínua e um triângulo apontando para a superclasse.
- . **Em Java**, é implementada usando extends

Exercício de Revisão

Questões para a Avaliação I

Acerca das Definições que envolvem o paradigma OO, responda as questões seguintes

10. O que você entende por hierarquia de herança ? Como é representada em UML e em JAVA ?

R. .Hierarquia de herança é a estrutura em que várias classes estão organizadas de forma hierárquica, com classes mais especializadas herdando características de classes mais gerais (superclasses). Isso forma uma árvore de herança, onde subclasses herdam atributos e métodos de suas superclasses.

- **Em UML**, é representada por várias setas de herança conectando subclasses a superclasses.
- **Em Java**, a hierarquia é criada com várias classes usando extends para estabelecer as relações

Exercício de Revisão

Questões para a Avaliação I

Acerca das Definições que envolvem o paradigma OO, responda as questões seguintes

11. O que é polimorfismo ? Defina cada tipo. Dê exemplos usando UML e JAVA.

R. .

****Polimorfismo**** é a capacidade de objetos diferentes responderem de maneiras variadas a uma mesma mensagem (chamada de método).

1. ****Polimorfismo de Subtipo (ou em Tempo de Execução)****: Permite que objetos de subclasses sejam tratados como objetos da superclasse, mas com comportamento específico da subclasse.

- ****UML****: Representado por herança entre classes, com métodos sobrescritos.

2. ****Polimorfismo de Sobrecarga (ou em Tempo de Compilação)****: Vários métodos com o mesmo nome, mas com assinaturas diferentes (número ou tipo de parâmetros), na mesma classe.

- ****UML****: Representado por múltiplos métodos com o mesmo nome, mas assinaturas diferentes.

Exercício de Revisão

Questões para a Avaliação I

Acerca das Definições que envolvem o paradigma OO, responda as questões seguintes

12. Qual a funcionalidade da interface ? Como é representada em UML e em JAVA ?

R. **.Interface** define um contrato de métodos que uma classe deve implementar, sem fornecer a implementação desses métodos. Ela permite a criação de código flexível e reutilizável, pois classes podem implementar múltiplas interfaces, promovendo polimorfismo.

- **Em UML**, uma interface é representada por um retângulo com o nome precedido por <<interface>>. As interfaces são conectadas às classes que as implementam por setas com linha contínua e um triângulo vazio.
- **Em Java**, uma interface é definida usando a palavra-chave interface.

Exercício de Revisão

Questões para a Avaliação I

Acerca das Definições que envolvem o paradigma OO, responda as questões seguintes

13. O que é encapsulamento ?

R. . ****Encapsulamento**** é o princípio de ocultar os detalhes internos de uma classe e expor apenas o que é necessário para o uso externo. Isso é feito através do uso de modificadores de acesso (como ``private``, ``protected``, ``public``) e métodos de acesso (getters e setters), protegendo os dados e controlando como eles são manipulados.

- ****Benefícios****: Melhora a segurança dos dados e facilita a manutenção e evolução do código, permitindo alterações internas sem afetar o código externo que utiliza a classe.

Exercício de Revisão

Questões para a Avaliação I

Acerca das Definições que envolvem o paradigma OO, responda as questões seguintes

14. O que você entende por moderadores de acesso associados ao conceito de polimorfismo ? Quais os três principais estudados ? Como são representados em UML e em JAVA ?

R. .

Modificadores de acesso controlam a visibilidade e o acesso aos membros (atributos e métodos) de uma classe, influenciando o polimorfismo ao definir como e onde esses membros podem ser acessados ou sobrescritos.

Os três principais modificadores de acesso são:

1. **public**: Membros acessíveis de qualquer lugar.
 - **UML**: Representado por +.
 - **Java**: Usado como public.
2. **protected**: Membros acessíveis apenas dentro da mesma classe, subclasses e classes no mesmo pacote.
 - **UML**: Representado por #.
 - **Java**: Usado como protected.
3. **private**: Membros acessíveis apenas dentro da própria classe.

Exercício de Revisão

Questões para a Avaliação I

Acerca das Definições que envolvem o paradigma OO, responda as questões seguintes

- **UML:** Representado por -.
- **Java:** Usado como private.

Esses modificadores afetam como os métodos podem ser sobrescritos e acessados em contextos de herança e interfaces.

Exercício de Revisão

Questões para a Avaliação I

Acerca das Definições que envolvem o paradigma OO, responda as questões seguintes

15. Quais as vantagens do modelo OO para o desenvolvimento de software ?

R. .

As vantagens do modelo orientado a objetos (OO) para o desenvolvimento de software incluem:

1. ****Reuso de Código****: Permite criar classes reutilizáveis e estender funcionalidades através da herança.
2. ****Manutenção Facilitada****: A modularidade e encapsulamento tornam mais fácil modificar e manter o código.
3. ****Abstração****: Facilita o entendimento e gerenciamento de sistemas complexos ao ocultar detalhes internos.
4. ****Polimorfismo****: Permite que diferentes classes possam ser tratadas de forma uniforme, promovendo flexibilidade e escalabilidade.
5. ****Modularidade****: O código é dividido em classes e objetos, o que ajuda a organizar e estruturar o software de forma mais clara.

Exercício de Revisão

Questões para a Avaliação I

Acerca das Definições que envolvem o paradigma OO, responda as questões seguintes

16. Dado o Projeto Geometria, com o método `getArea()` implementado para as classes `Quadrado`, `Triangulo`, `Circunferencia` e `Retangulo`, implemente os métodos `getPerimetro()` e `getCor()`.

R. .