

Resumo P. Prova SIBD Java

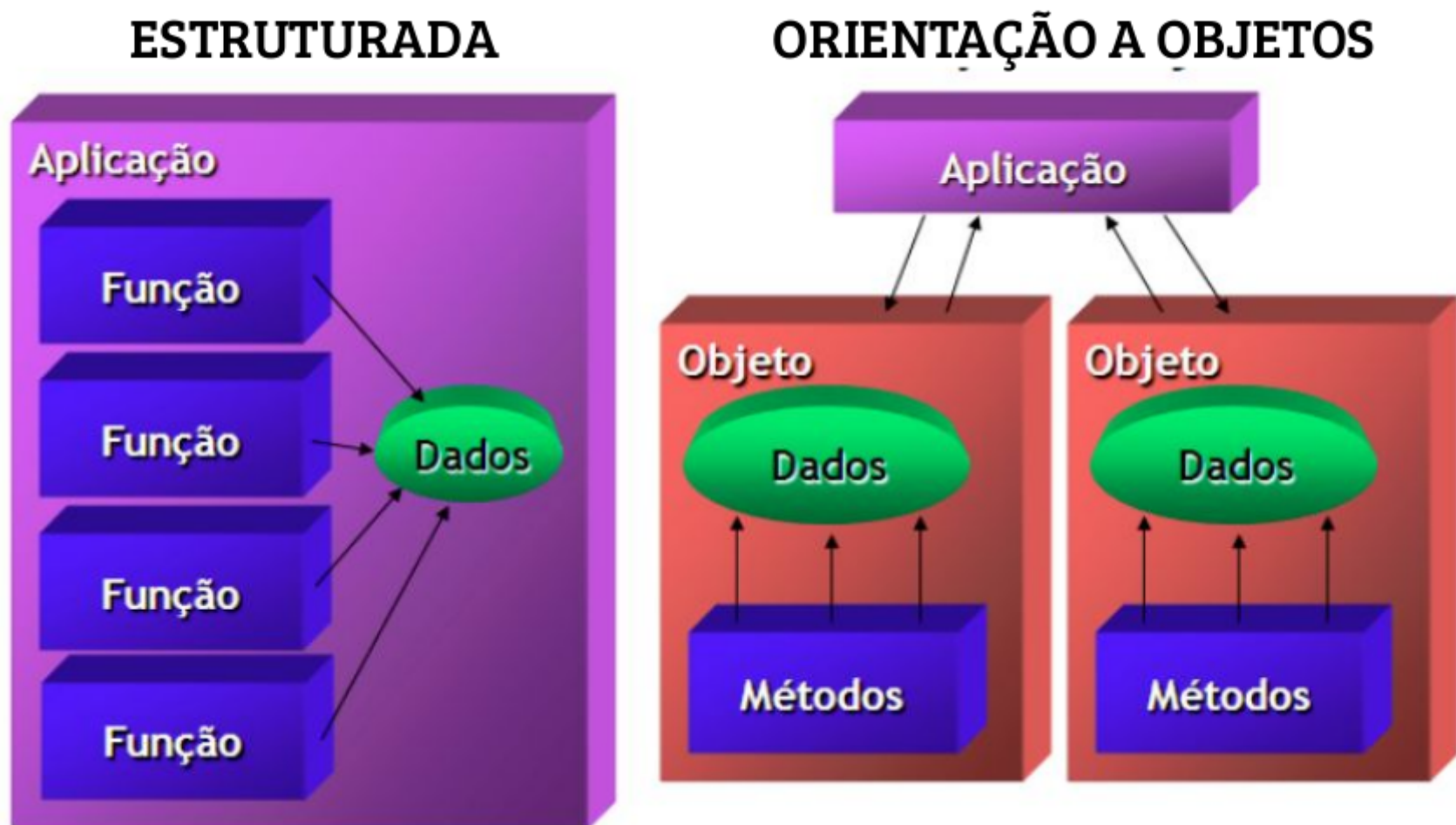
O que é Programação Estruturada?

- Essa programação preconiza que todos os programas possíveis podem ser reduzidos a apenas três estruturas:
- Sequência:** Uma tarefa é executada após a outra, linearmente.
 - Decisão:** A partir de um teste lógico, determinado trecho de código é executado, ou não.
 - Iteração (repetição):** A partir de um teste lógico, determinado trecho de código é repetido por um número finito de vezes.

O que é Programação Orientada a Objetos?

- A programação orientada a objetos (POO) é um modelo de programação onde diversas classes possuem características que definem um objeto.

Programação Estruturada vs POO



O que é uma POO?

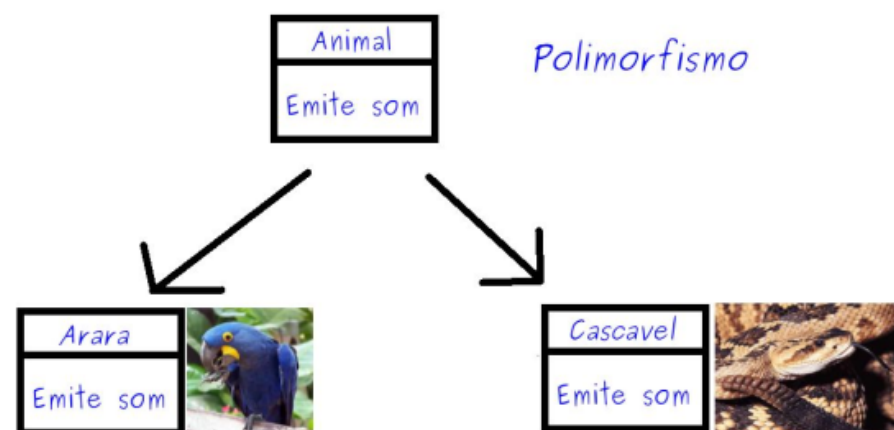
- Paradigma de programação baseado no conceito de classes e objetos.
- As classes são elementos onde dados e procedimentos são agrupados, segundo seu objetivo, para um determinado sistema.
- Quando uma classe é usada como um tipo de dado para a criação de uma variável, esta é chamada de objeto.

O que a POO Engloba?

- Encapsulamento:** É a capacidade de restringir o acesso a elementos de uma classe utilizando modificador de visibilidade (Public; Private; Protected)
- Pilares de POO:
 - Herança:** A herança permite o uso de propriedades e métodos de uma classe já existente ao invés de implementar um programa do zero. Ou ainda, uma maneira de reutilizar código à medida que podemos aproveitar os atributos e métodos de classes já existentes para gerar novas classes mais específicas que aproveitarão os recursos da classe "hierarquicamente superior".
 - Extends:** Essa palavra reservada é usada para indicar que a classe herda (estende) de outra classe

```
class Subclasse extends Superclasse {
    /* ... */
}
```

- **Encapsulamento:** Separar o programa em partes, o mais isolado possível. A idéia é tornar o software mais flexível, fácil de modificar e de criar novas implementações. O encapsulamento facilita a manutenção de classes, bem como, garante a integridade dos atributos de um objeto em um determinado instante
- **Polimorfismo:** É a característica única de linguagens orientadas a objetos que permite que diferentes objetos respondam à mesma mensagem, cada um ao seu modo.



O que é um Objeto?

- Um objeto é uma entidade que exibe algum comportamento bem definido.
- Os objetos tem:
 1. **Atributos:** Características
 2. **Métodos:** São ações que uma classe possui

O que é uma Classe?

- Uma classe é a forma para criação de objetos.
- Uma classe serve de modelo para vários objetos semelhantes que possuem os mesmos tipos de informação em seu estado e tem os mesmos comportamentos.
- Uma classe é uma estrutura que abstrai um conjunto de objetos com características similares.

Java

- Java é tanto compilada como interpretada
- **JVM:** A Máquina Virtual Java (JVM) é um interpretador, que transforma as instruções em linguagem de máquina (permitindo a execução do programa).

Assinatura do Método

- A combinação do nome de um método com os tipos de sua lista de parâmetros é conhecida como assinatura do método.

Redefinição de Métodos Herdados

- Métodos herdados de uma superclasse podem ser redefinidos na subclasse.
- Deste que a assinatura do método na subclasse seja a mesma assinatura do método da superclasse.
- O tipo do valor de retorno (embora não sendo parte da assinatura) também deve ser o mesmo.
- Se os métodos herdados da superclasse forem definidos na subclasse com assinaturas diferentes haverá a sobrecarga e não a redefinição dos métodos.

Sobrecarga

- Permite que um nome de função seja utilizado mais do que uma vez com diferentes tipos de parâmetros.

O que é Sobrecarga de Métodos?

- Permite que um nome de função seja utilizado mais do que uma vez com diferentes tipos de parâmetros.
- **Ex:** Uma função soma pode ser sobrecarregada para operar com dois parâmetros inteiros e dois reais.

Interfaces

- A interface é um contrato que quando assumido por uma classe deve ser implementado.
- Uma interface nada mais é do que um bloco de código definindo um tipo e os métodos e atributos que esse tipo deve possuir.
- A interface não contém nenhum código de implementação, apenas assinaturas de métodos e/ou atributos que devem ter seu código implementado nas classes que “chamar” essa interface.
- Uma interface é definida através da palavra reservada `interface`.
- Para uma classe implementar uma interface é usada a palavra `implements`

```
public class nomeClasse implements nomeInterface{  
    /* ... */  
}
```

- Interfaces admitem apenas os níveis de acesso `public` e `default`.

Classes Abstratas

- É um tipo de classe especial que não pode ser instanciada, apenas herdada.
- Uma classe abstrata não pode ter um objeto criado a partir de sua instanciação.
- Possibilita herança de código preservando comportamento (semântica).
- Uma classe abstrata nada mais é do que uma especificação conceitual para outras classes.
- Uma classe abstrata pode conter métodos abstratos que as classes que irão estendê-la devem implementar.
- A classe abstrata só diz o que deve ter a subclasse.

Diferença entre Interfaces e Classes Abstratas

- **Classes abstratas** podem conter métodos não-abstratos, isto é, que contêm implementação e que podem ser herdados e utilizados por instâncias das subclasses.
- **As interfaces** não podem conter nenhum método com implementação, todos os seus métodos são implicitamente abstract e public e não possuem corpo.
- A diferença essencial entre classes abstratas e interfaces em Java é que uma subclasse somente pode herdar de uma única classe (abstrata ou não), enquanto qualquer classe pode implementar várias interfaces simultaneamente.

Exceções

- Exceções são os erros imprevistos durante a execução de programas de computador. Ou ainda, são comportamentos inesperados que o programa assume.

Qual a Diferença entre Erro e Exceção?

- **Erro** é uma condição crítica que não pode ser tratada pelo código do programa.
- **Exceção** é a situação excepcional que pode ser tratada pelo código do programa.

RuntimeException

- É a superclasse dessas exceções que podem ser lançadas durante a operação normal da Java Virtual Machine.
- Não é obrigatório o tratamento dessa exceção, você pode tratar se sentir que é necessário.

Checked Exceptions

- Checked Exceptions (exceções verificadas) são exceções que você é obrigado a tratá-la, seja com um bloco `try-catch` ou mesmo com um `throws`.

Unchecked exceptions

- são utilizadas para erros irre recuperáveis.

