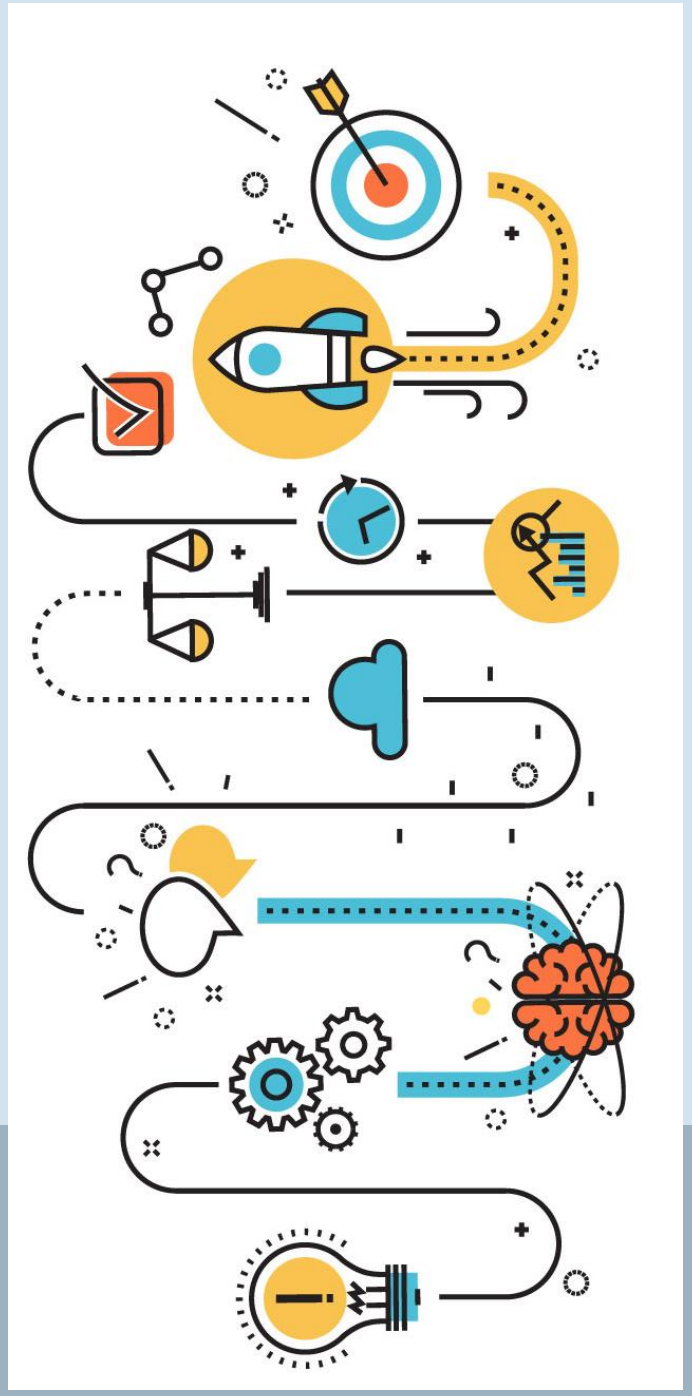


Introdução a Classes, Métodos e Pacotes

Prof. Rhafael Freitas da Costa, M. Sc.

Desenvolvimento de Software

O que é um paradigma de desenvolvimento?

- Os paradigmas de desenvolvimento são conjuntos de princípios e práticas que guiam a criação de software. Eles servem como lentes através das quais os desenvolvedores concebem, estruturam e implementam seus projetos. Cada paradigma oferece diferentes abordagens para resolver problemas de software, com vantagens e desvantagens distintas.

Paradigma Procedural

Definição

- Neste paradigma, o software é estruturado em torno de procedimentos ou rotinas que manipulam dados. As linguagens de programação procedurais mais comuns incluem C, Pascal e BASIC. A ênfase está na sequência de instruções para alcançar um resultado específico.

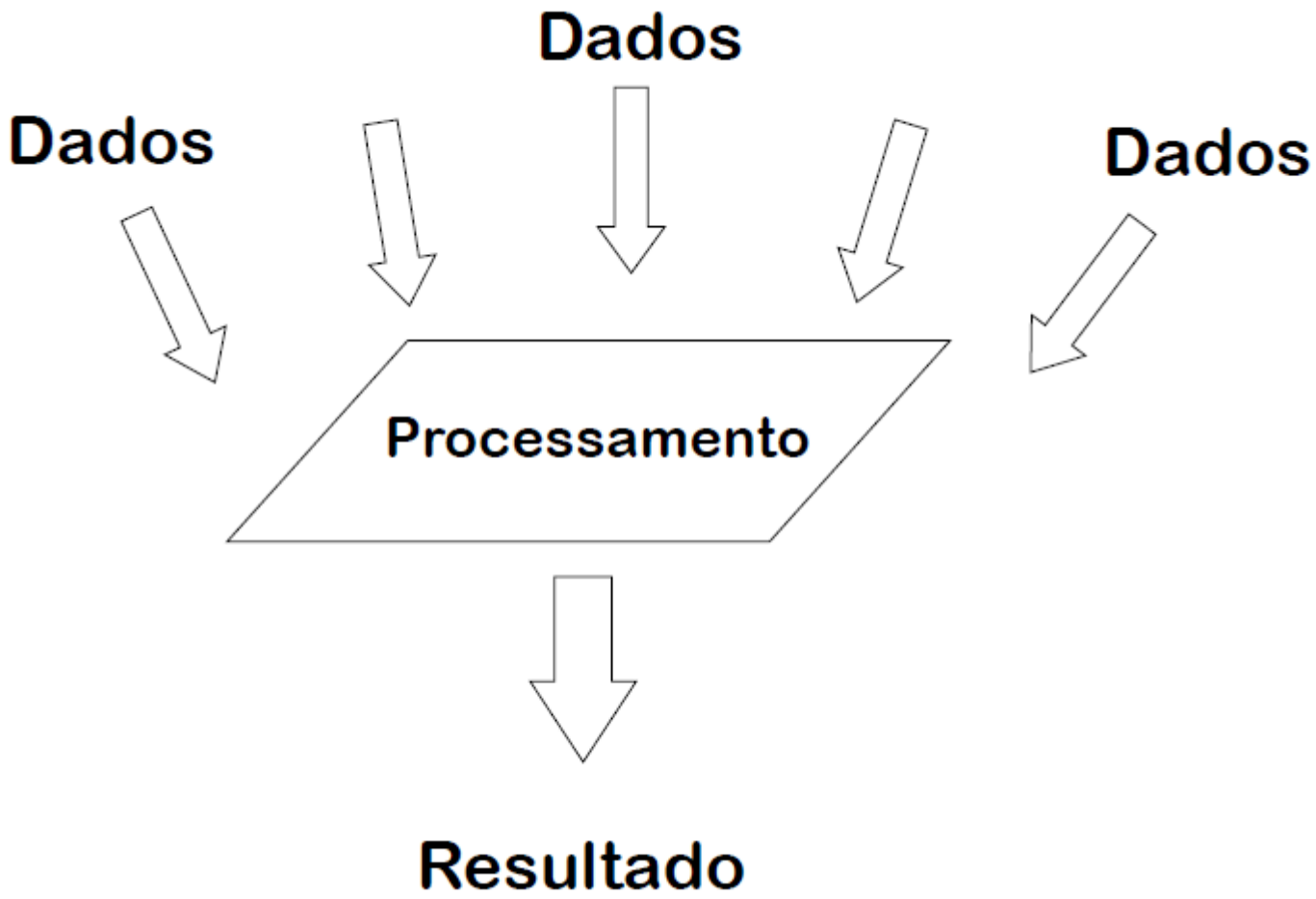
Vantagens

- Fácil entendimento para iniciantes.
- Controle direto sobre os recursos do sistema.
- Eficiente em problemas simples e algoritmos lineares.

Desvantagens

- Difícil de manter e escalar em projetos grandes.
- Menos modularidade e reutilização de código.
- Propenso a erros devido à manipulação direta de memória.

Paradigma Procedural



Paradigma Orientado a Objetos

Definição

- No paradigma orientado a objetos, o software é estruturado em torno de objetos, que podem conter dados (conhecidos como atributos) e métodos (ou procedimentos). O foco está na modelagem do mundo real, onde entidades do mundo real são representadas como objetos. Linguagens como Java, C++, e Python são exemplos de linguagens orientadas a objetos.

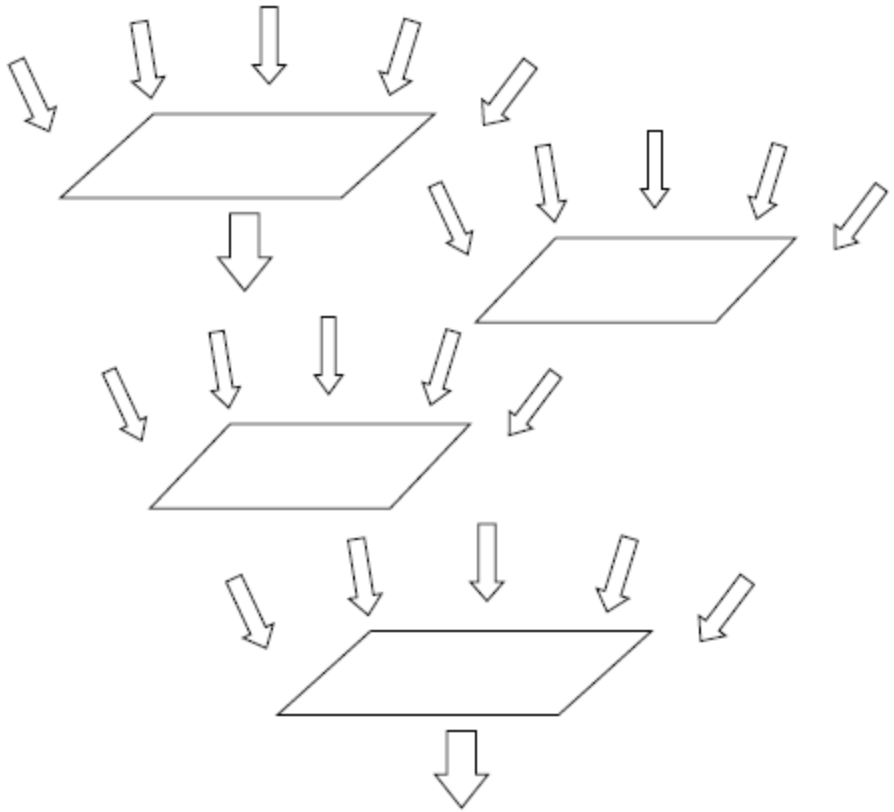
Vantagens

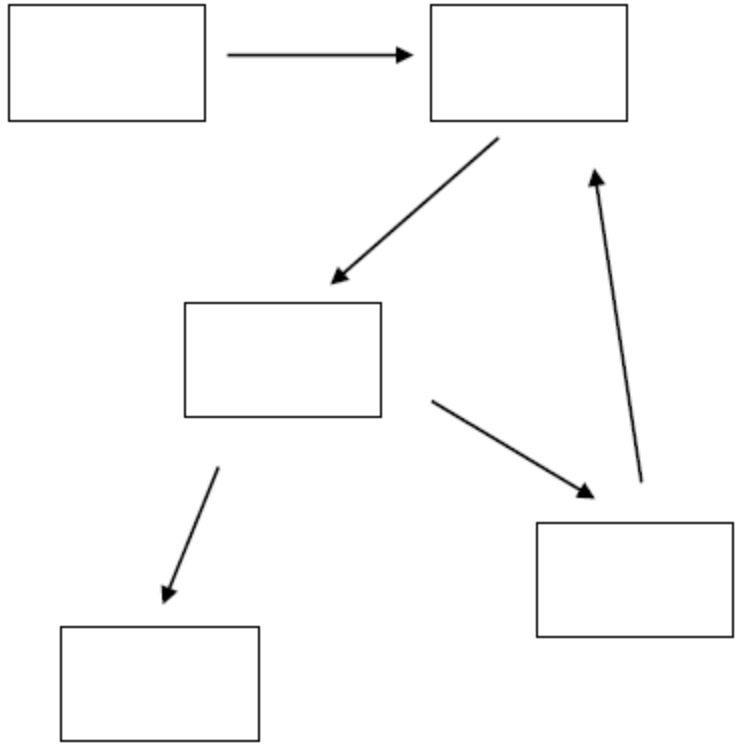
- Reutilização de código através de herança e polimorfismo.
- Estrutura organizada, facilitando a manutenção.
- Modelagem do mundo real, tornando o código mais intuitivo.

Desvantagens

- Pode levar a um excesso de complexidade em projetos pequenos.
- Overhead de desempenho devido à alocação de objetos.
- Curva de aprendizado mais íngreme para iniciantes.

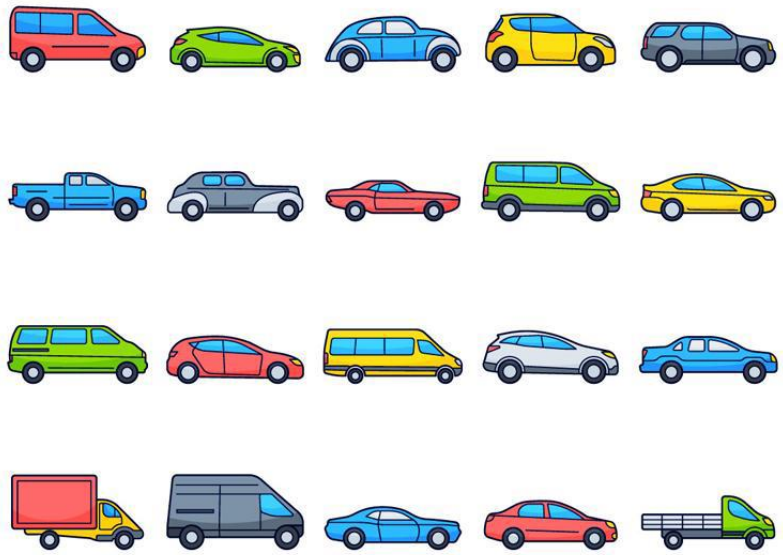
Paradigma Orientado a Objetos





Abstração

- Construção de um modelo para representar algo da realidade;
- Foco apenas em aspectos essenciais;
- Preservação da simplicidade do projeto.



Modularidade

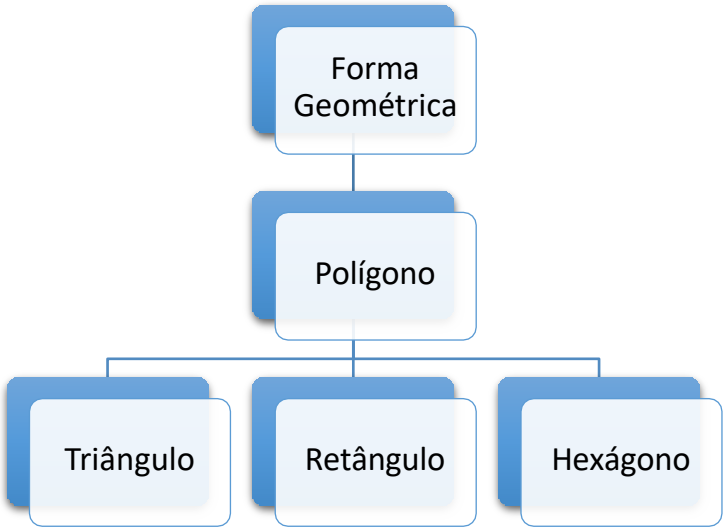
- Quebrar algo complexo em partes menores;
- Facilita o entendimento;
- Cada parte pode ser desenvolvida separadamente.

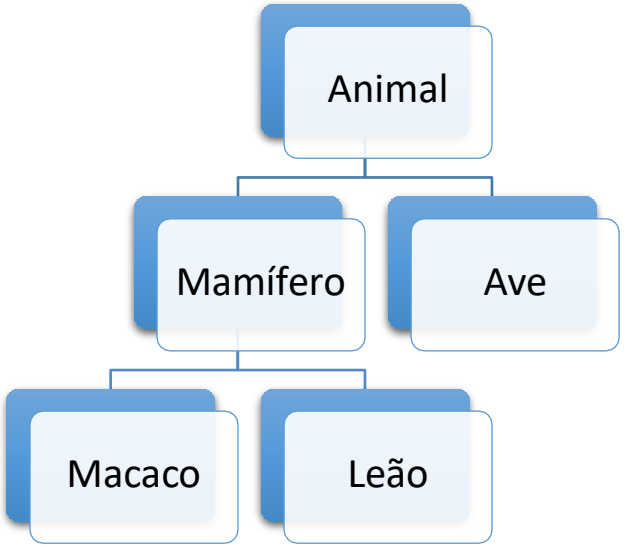
Encapsulamento

- As informações do objeto ficam encapsuladas, como se fosse uma caixa protegida;
- Evita que as informações sejam corrompidas por entidades externas;
- Mudanças internas não impactam os clientes;
- Manutenções mais baratas e fáceis.

Hierarquia

- Define níveis de abstração;
- Base conceitual para que o software seja extensível;
- Permite reuso de código/comportamento.





Aumenta a abstração

Diminui a abstração

Revisão

- Conceito de objeto, entidade mais próxima do mundo real;
- Objeto deve ser abstrato, modular, encapsulado, hierárquico;
- **Abstrato**, foca em aspectos essenciais ao projeto, e nada mais;
- **Modular**, pois o complexo é quebrado em partes menores;
- **Encapsulado**, pois o objeto é responsável por seus próprios dados;
- **Hierárquico**, pois permite a construção de modelos extensíveis;

Paradigma Funcional

Definição

- No paradigma funcional, o software é construído em torno de funções matemáticas puras. O código é escrito como uma série de funções que recebem entradas e produzem saídas sem efeitos colaterais. Linguagens como Haskell, Lisp e Clojure são exemplos de linguagens funcionais.

Vantagens

- Evita efeitos colaterais, facilitando a depuração e teste.
- Ênfase em funções puras torna o código mais previsível.
- Suporta programação concorrente de forma mais segura.

Desvantagens

- Nem todos os problemas se encaixam bem no modelo funcional.
- Curva de aprendizado íngreme para desenvolvedores acostumados com paradigmas imperativos.
- Menos eficiente em tarefas que envolvem mutação de estado.

Paradigma Funcional

Classes

Definição

- As classes são estruturas fundamentais na programação orientada a objetos. Elas definem os atributos e comportamentos de objetos.

Importância

- Classes oferecem reutilização eficiente e organização de código, facilitando a manutenção e o desenvolvimento de sistemas.

Métodos

Significado

- Métodos representam o comportamento dos objetos, definindo as ações que podem ser realizadas com eles.

Relevância

- Os métodos permitem a modularidade, reutilização e organização do código, tornando os sistemas mais eficientes.

Pacotes

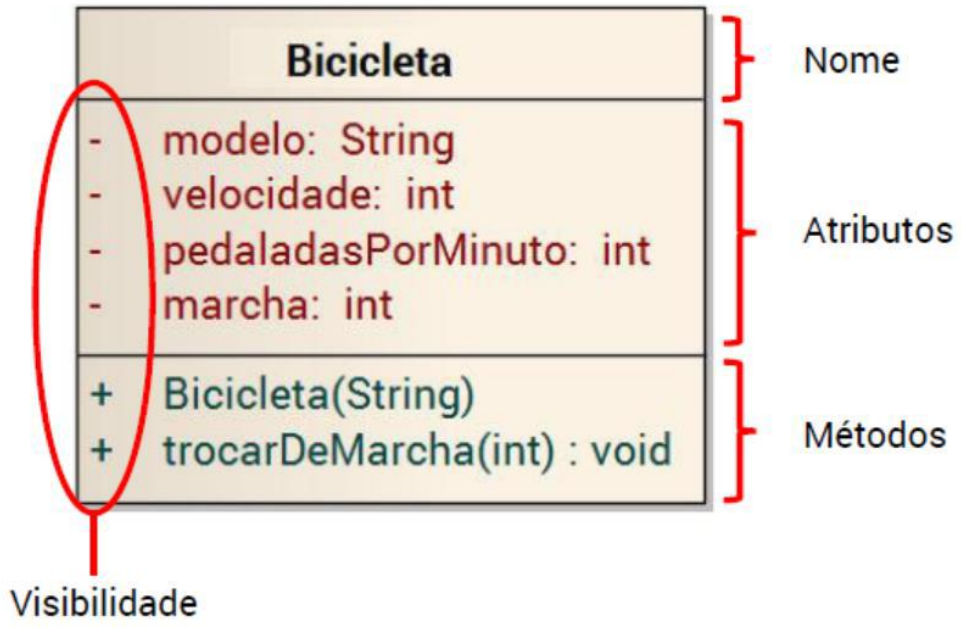
Definição

- Pacotes são espaços de nomes que agrupam classes relacionadas, organizando-as em módulos funcionais e reutilizáveis.

Vantagens

- Pacotes promovem a encapsulação, modularização e reutilização de código, contribuindo para a escalabilidade e manutenibilidade dos sistemas.

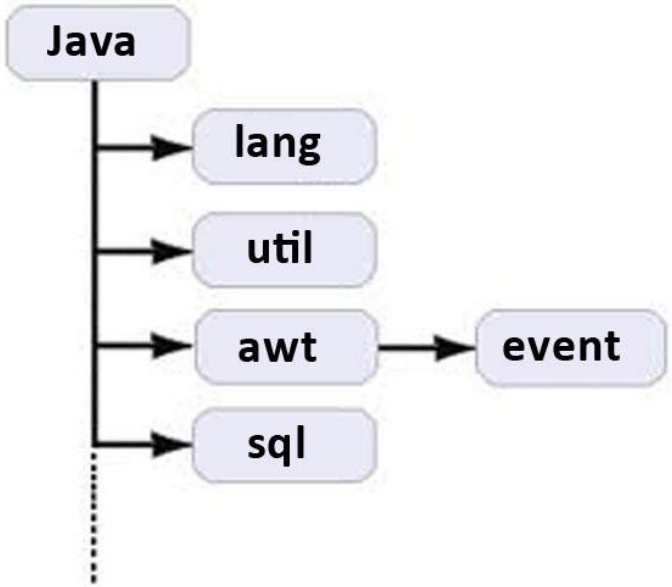
Classe (é um tipo evoluído personalizado)




```java
class Bicicleta {
    private String modelo;
    private int velocidade =  0;
    private int pedaladasPorMinuto =  0;
    private int marcha =  1;

    Bicicleta(String modelo) {
        this.modelo = modelo;
    }
    void trocarDeMarcha(int novaMarcha) {
        marcha = novaMarcha;
    }
}
```

Pacotes



br.edu.up.calculadora

Exercícios

Métodos

- Refatorar as listas de exercícios 1 e 2 para que os exercícios fiquem em pacotes distintos;
- Criar uma classe Menu para apresentar a escolha das listas e logo após o exercício a ser executado;
- Desenvolver a lógica para que após executar o exercício o usuário possa voltar para a lista de opções e/ou encerrar a aplicação.

Exercício 8

- Crie uma classe chamada Carro com os seguintes atributos públicos: modelo, ano, cor. Crie também um método público chamado mostrar_informacoes() que imprime as informações do carro.

Exercício 9

- Crie uma classe chamada Retangulo com os seguintes atributos públicos: largura e altura. Adicione métodos públicos para calcular a área e o perímetro do retângulo.

Exercício 10

- Crie uma classe chamada ContaBancaria com os seguintes atributos públicos: titular, saldo. Adicione métodos públicos para depositar e sacar dinheiro da conta.

Exercício 11

- Crie uma classe chamada Produto com os seguintes atributos públicos: nome, preco, quantidade. Adicione métodos públicos para atualizar a quantidade e mostrar o valor total dos produtos em estoque.

Exercício 12

- Crie uma classe chamada Pessoa com os seguintes atributos públicos: nome, idade, altura. Adicione métodos públicos para imprimir as informações da pessoa e para verificar se ela é maior de idade.

OBRIGADO!

Desenvolvimento de Software

Professor Rhafael Freitas
da Costa, M. Sc.