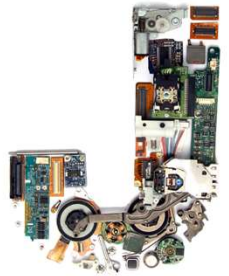
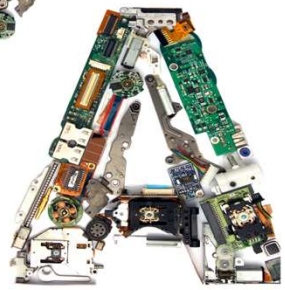
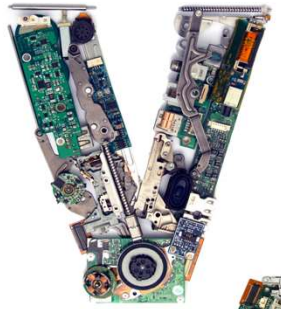





Programação Orientada a Objetos com Java e WEB

Introdução à Programação Orientada a Objetos

Introdução


VIDADEPROGRAMADOR
.COM.BR
/* HISTÓRIA REAL ENVIADA POR WAGNER */
#344
CARA... O QUE É "ORIENTADO A OBJETOS"?
OH, COFFEE MY COFFEE
É UMA FORMA DE PROGRAMAR QUE FOI CRIADA PARA TENTAR APROXIMAR O MUNDO REAL DO MUNDO VIRTUAL USANDO PROPRIEDADES, OBJETOS, MÉTODOS, ETC...
OH, COFFEE MY COFFEE
AHHHH TÁ... É QUE EU ACHO QUE FAÇO ORIENTADO A OBJETOS E NEM SEI...
PLAFT
OH, COFFEE MY COFFEE

Introdução

A programação nada mais é do que o desenvolvimento de programas que implementam determinadas funcionalidades.

Um programa é um conjunto organizado de instruções que operam sobre um conjunto de dados, processando-os, afim de realizar alguma funcionalidade e produzir alguma saída.

Para um número pequeno de instruções essa tarefa é relativamente simples, mas torna-se complexa quando o número de instruções aumenta.

Introdução

Programas maiores são mais complicados de se criar, são mais difíceis de organizar as instruções de forma a otimizar os recursos do sistema e também há a tendência de apresentarem erros, dentre outros fatores.

Erros de software podem ter elevado custo de correção, resultar em situações indesejáveis, como a indisponibilidade de sistema e até mesmo colocar vidas em perigo.

Introdução

A **programação orientada a objetos** fornece uma nova forma para tratar a complexidade quando programas tornam-se maiores e mais complexos

A **grande meta da orientação a objetos** é obter programas que sejam mais confiáveis e de fácil manutenção

Programação Estruturada

A **programação estruturada** foca na resolução do problema, sem ter uma preocupação maior com a estruturação dos dados que serão processados, ou seja, o foco está na resolução do problema e não nos dados.

Os programas escritos usando o paradigma da programação estruturada são organizados da seguinte forma:

- ❑ O programa tem um projeto **modular** (**métodos** / **funções** / **rotinas**).
- ❑ O programa tem um módulo principal que gerencia a chamada dos outros módulos.
- ❑ Os módulos são projetados utilizando-se as três estruturas de controle básicas: sequência, seleção e repetição (ausência total de instruções do tipo **GOTO**).

Programação Orientada a Objetos

Como o próprio nome indica, os "**atores**" principais do paradigma de **programação orientada a objetos** são os **objetos**.

Um **objeto** se origina a partir de uma **classe**, que é uma especificação tanto dos campos, também chamados de **variáveis de instância**, que um objeto contém, como dos **métodos** (operações) que pode executar.

Cada classe apresenta para o mundo exterior uma visão concisa e consistente dos objetos que são instâncias dessa classe, sem detalhes ou acesso às estruturas internas dos objetos.

Fonte: Goodrich, Michael T.; Tamassia, Roberto. Estruturas de Dados e Algoritmos em Java (p. 60). Edição do Kindle.

POO x PE

1

Os dados constituem a principal razão de um sistema, ou seja, os dados são a essência de um sistema. POO foca na estrutura dos dados (objetos)

2

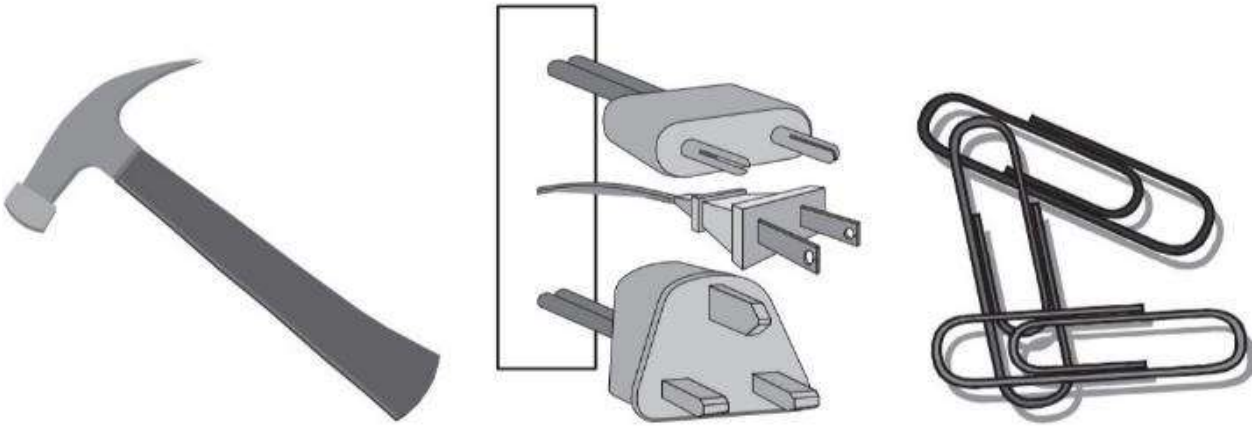
A programação orientada a objetos (POO) têm por finalidade descrever **objetos físicos** do mundo real (**livro**, **aluno**, **professor**, **quadrado**, etc), por meio de entidades denominadas **objetos na programação**

3

Na **programação estruturada** (PE) **a ênfase não está nas "coisas"** (objetos) e nos dados, mas sim em como fazer as "coisas" funcionarem, ou seja, como ler dados digitados pelo usuários, como ordenar um conjunto de dados, testar valores das variáveis, etc. **O foco está nas funções (métodos)!!!**

Objetivos do Projeto Orientado a Objetos

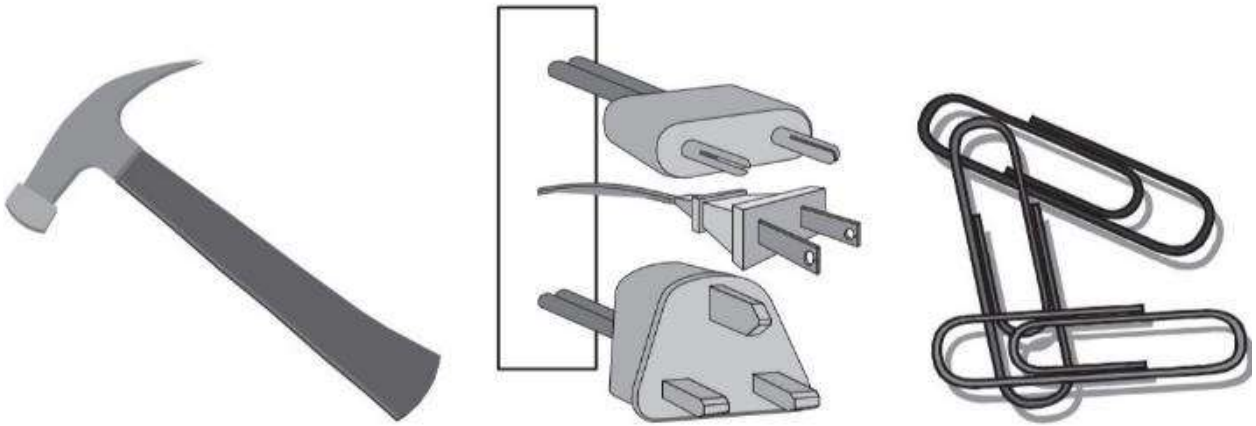
A implementação de um software deve buscar **robustez**, **adaptabilidade** e **reusabilidade**.



Robustez Adaptabilidade Reusabilidade

Fonte: Goodrich, Michael T.; Tamassia, Roberto. Estruturas de Dados e Algoritmos em Java (p. 60). Edição do Kindle.

Objetivos do Projeto Orientado a Objetos



Robustez | Adaptabilidade | Reusabilidade

Software robusto é capaz de lidar com entradas inesperadas que não estavam previstas na aplicação

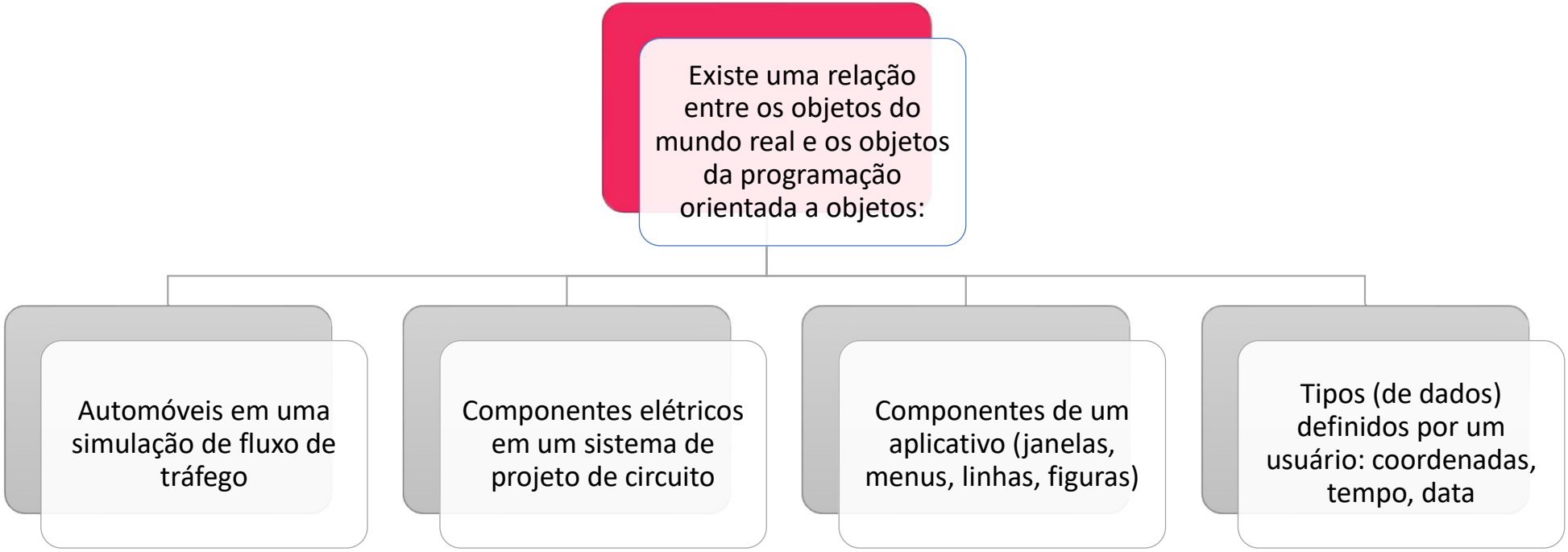
O software deve ser capaz de evoluir ao longo do tempo para de adaptar as condições do seu meio (**adaptabilidade**). Deve ser capaz de ser executado em diversas plataformas (**portabilidade**).

É desejável que possa ser **reutilizável**, ou seja, que seu código possa ser utilizado como componente em várias aplicações.

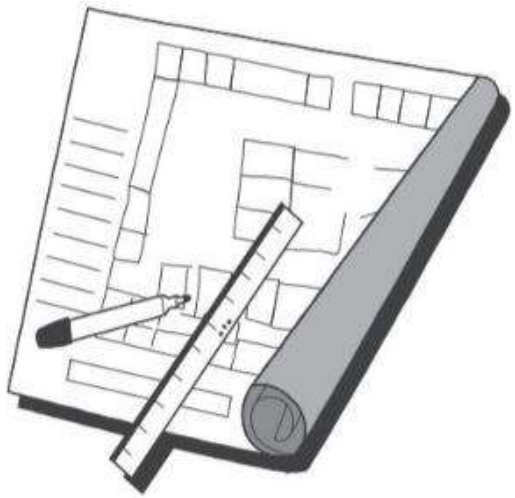
Fonte: Goodrich, Michael T.; Tamassia, Roberto. Estruturas de Dados e Algoritmos em Java (p. 60). Edição do Kindle.

Raciocínio Orientado a Objetos

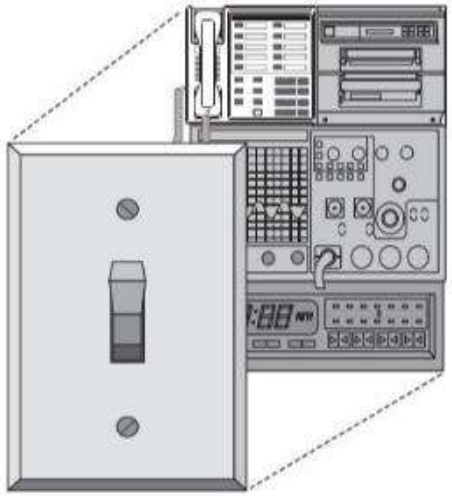
Quando um problema é tratado por uma linguagem orientada a objetos, o programa é dividido em **objetos**



Princípios do Projeto Orientado a Objetos



Abstração

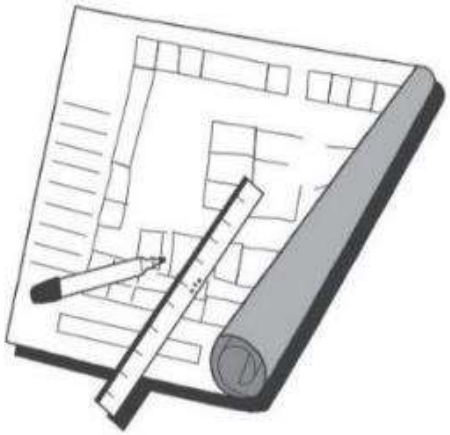


Encapsulamento

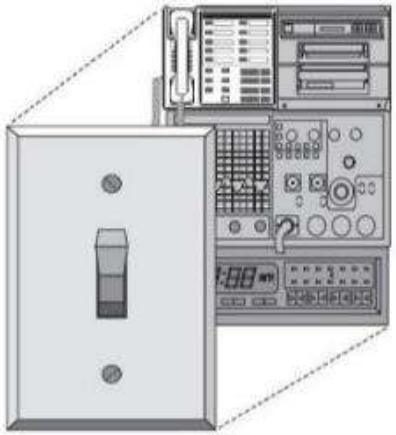
Modularidade

Fonte: Goodrich, Michael T.; Tamassia, Roberto. Estruturas de Dados e Algoritmos em Java (p. 61). Edição do Kindle.

Princípios do Projeto Orientado a Objetos



Abstração



Encapsulamento

Modularidade

Abstração: decompor um sistema complicado em partes fundamentais e descrevê-la em uma linguagem simples e precisa.

Encapsulamento: estabelece que os diferentes componentes do sistema não devem revelar detalhes da sua implementação.

Modularidade: sistemas modernos de software são compostos por vários componentes diferentes que devem interagir corretamente, fazendo com que o sistema como um todo funcione adequadamente.

Fonte: Goodrich, Michael T.; Tamassia, Roberto. Estruturas de Dados e Algoritmos em Java (p. 61). Edição do Kindle.

Conceito de Abstração

1

É a capacidade de identificação de coisas semelhantes quanto à forma e ao comportamento permitindo assim a organização em **classes**

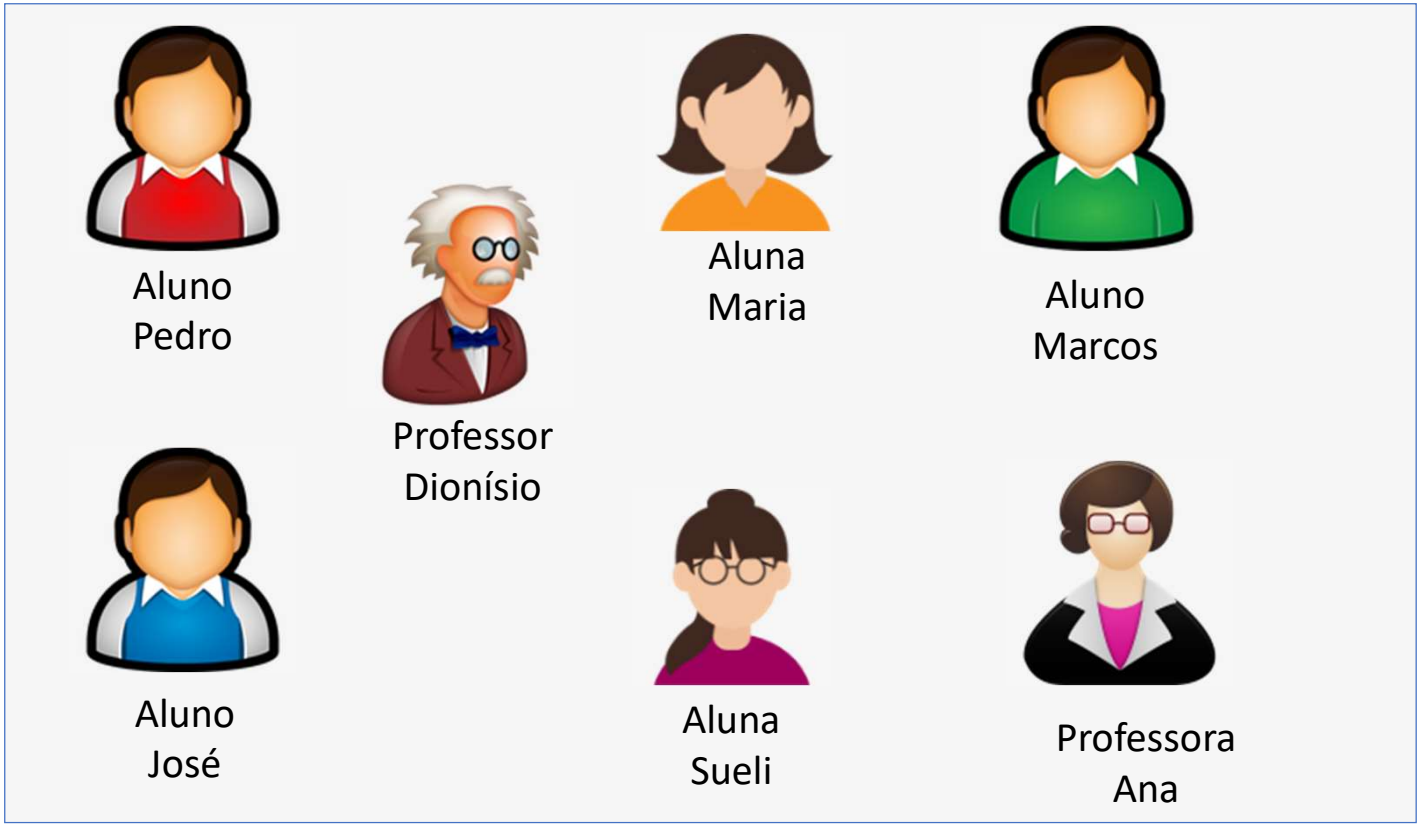
2

Fundamenta-se na **busca dos aspectos relevantes dentro do domínio do problema** e na omissão daquilo que não seja importante naquele contexto

3

A questão chave na modelagem orientada a objetos consiste na **identificação de abstrações que melhor descreve o domínio do problema**

Conceito de Abstração



Domínio universitário (entidades)

Conceito de Classe

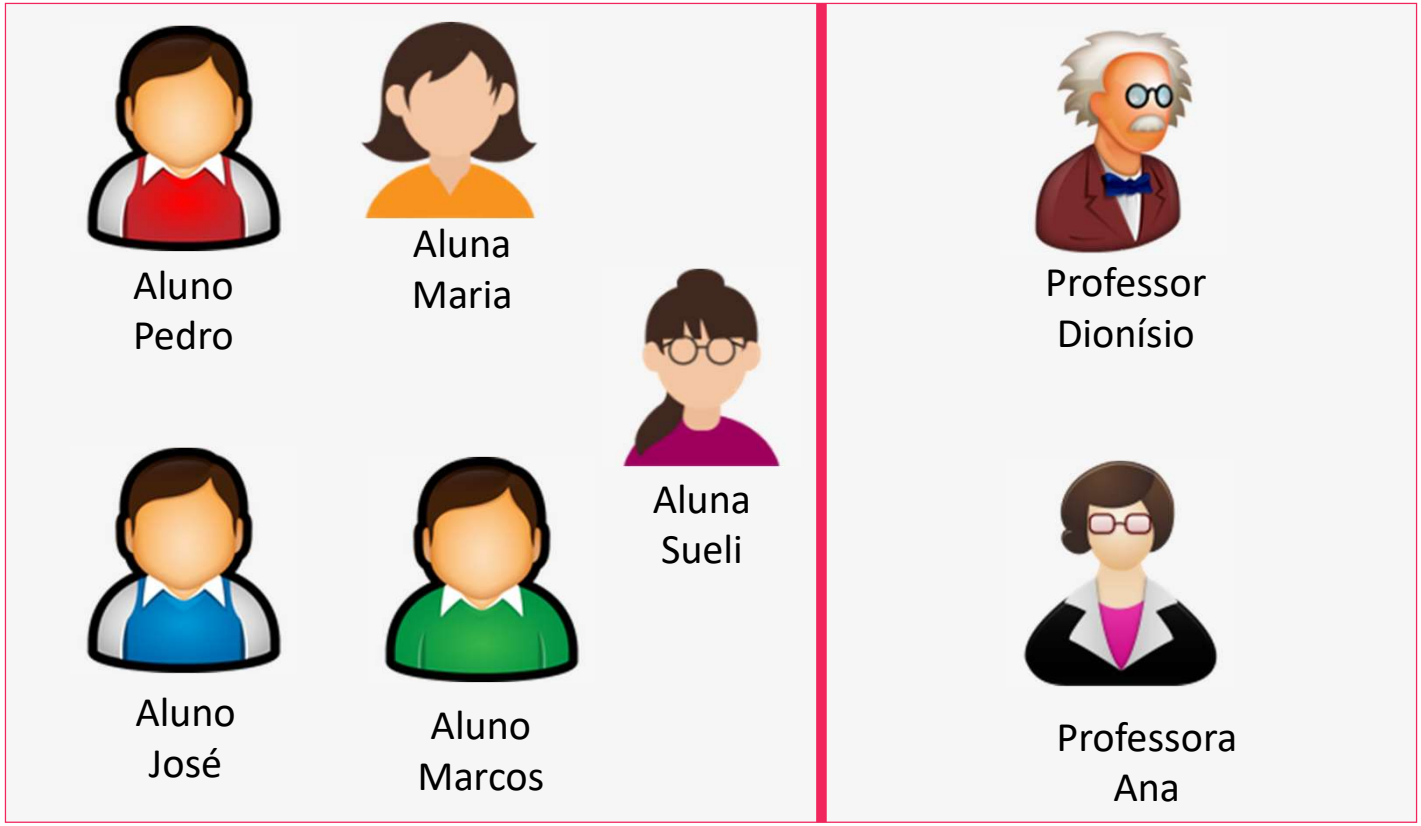
As abstrações são representadas pelas **classes**

Pode ser definida como um **modelo** que descreve um conjunto de elementos que compartilham as mesmas características

Uma classe serve como um *padrão*, *modelo* ou *template*. *A classe especifica quais dados e quais funções* (*métodos*) *serão incluídas nos objetos daquela classe*

Definir uma classe não cria quaisquer objetos

Conceito de Classe



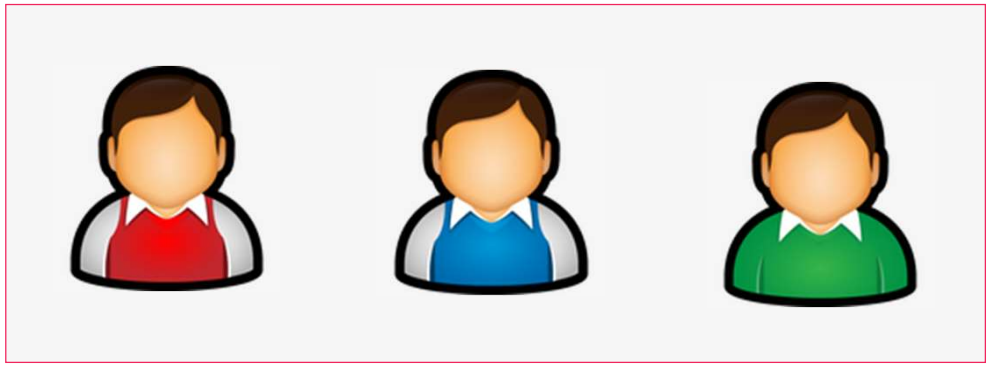
Conceito de Classe

Deve conter apenas os elementos necessários para resolver um aspecto bem definido do sistema

Nas classes são encontrados **atributos** (**ou propriedades ou campos**) e **métodos** (**ou serviços**) que resumem as características comuns de vários objetos

Conceito de Classe – Exemplo

Classe Aluno

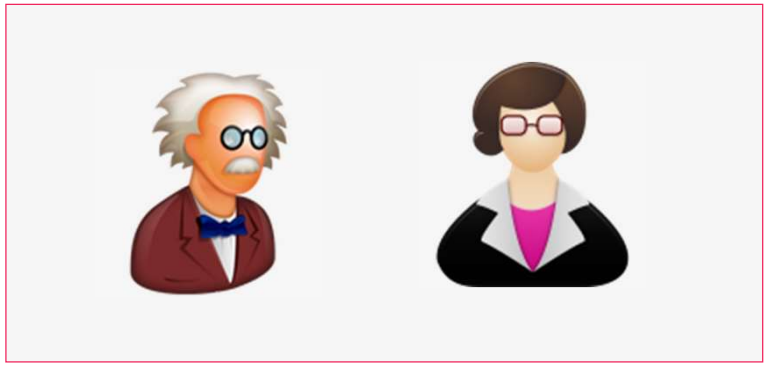


Objeto: aluno1

Objeto: aluno2

Objeto: aluno3

Classe Professor



Objeto: professor1

Objeto: professor2

Atributos

1

Características particulares de um objeto

Exemplo: indivíduo tem nome, gênero, idade, altura, etc.

2

Atributos também são chamados de propriedades ou campos ou variáveis de instância.

3

Dado ou informação de **estado** (valores dos atributos), para o qual cada objeto em uma classe tem seu próprio valor.

Objetos

Um elemento representante de uma classe é uma *instância da classe* ou *objeto*

Um objeto é uma entidade concreta (apesar de sua concepção abstrata) que possui:

Identidade: característica que o distingue dos demais objetos	**Estado**: conjunto de valores de seus atributos em um determinado instante	**Comportamento**: reação apresentada às solicitações feitas por outros objetos com os quais se relaciona

Objetos

Informalmente um objeto representa uma entidade, tanto física quanto conceitual ou de software

Podemos afirmar que um objeto é um conceito, abstração, ou entidade com limites bem definidos e um significado para aplicação. Exemplos:

Entidade Física: caminhão, carro, bicicleta, etc	**Entidade Conceitual:** processo químico, matrícula, etc	**Entidade de Software**: lista, arquivo, etc

Objetos – Exemplos

Objeto 1	Objeto 2	Objeto 3

nome = “maria” **rm** = 302569 **idade** = 15	**nome** = “joão” **rm** = 25824 **idade** = 25	**nome** = “ana” **rm** = 40123 **idade** = 35

Atributos? Identidade? Estado? Comportamento?

Diferença entre Classe e Objeto

1

A diferença fundamental entre classes e objetos está no fato de que um **objeto constitui uma entidade concreta com tempo e espaço de existência (ocupa espaço na memória)**, enquanto a classe é somente uma ***abstração*** (***tipo de dado definido pelo usuário***)

2

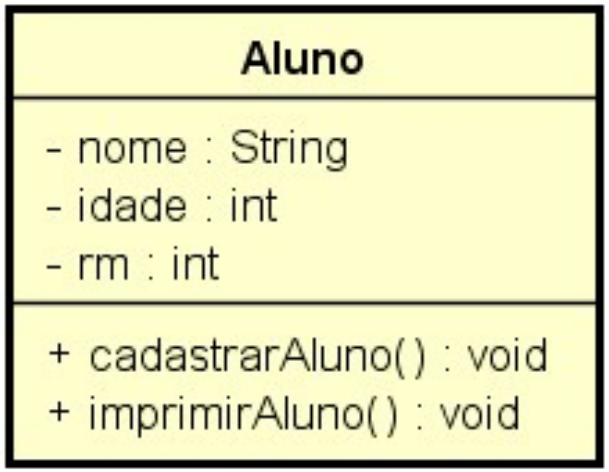
Em termos de programação, ***definir uma classe significa formalizar um tipo de dado*** e as operações associadas a esse tipo. **Quando criamos uma classe criamos um novo tipo de dado.**

3

Classe representa uma categoria, e os objetos são os membros ou representantes dessa categoria

Sistemas de Informação | FIAP
Prof. Dr. Antonio Marcos SELMINI – selmini@fiap.com.br

Representação de Classes na UML (Linguagem de Modelagem Unificada)



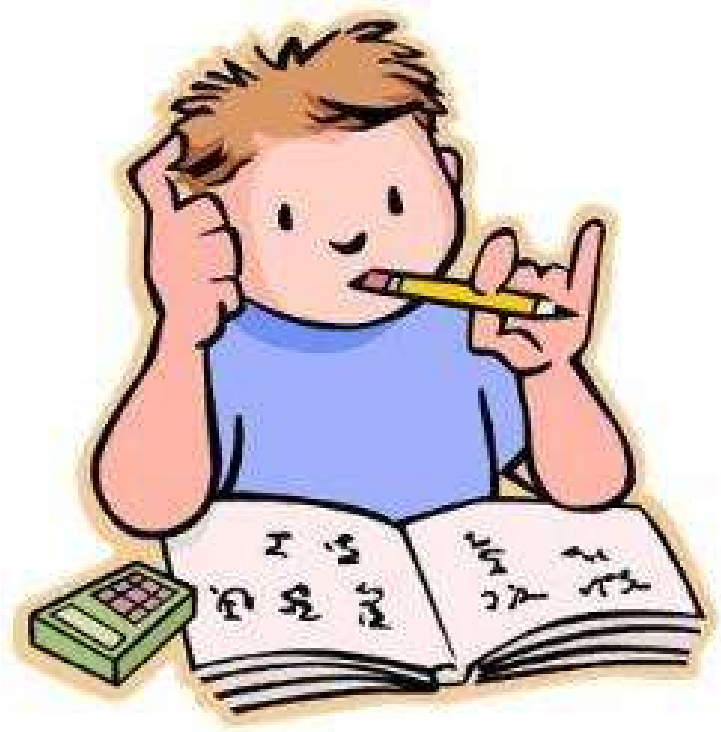
→ Nome da classe

→ Atributos da classe
(características gerais definidas na classe.
São também chamados de membros da classe).

→ Métodos ou serviços da classe
(operações realizadas sobre os atributos da classe.)

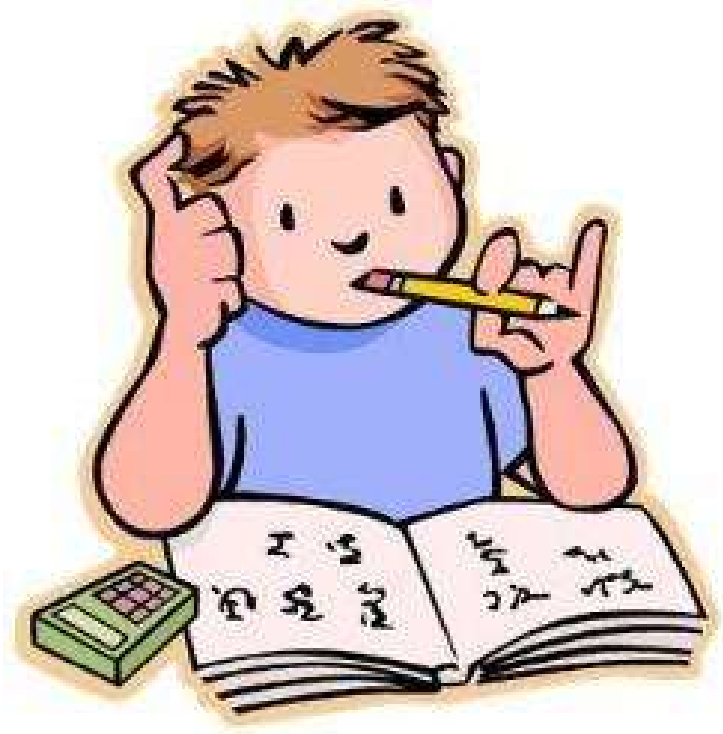
Exercício 1

- Identifique os objetos no problema abaixo:
  - Um cinema pode ter muitas salas, sendo necessário, portanto, registrar informações a respeito de cada sala, como sua capacidade (número de lugares disponíveis).
  - O cinema apresenta vários filmes. Um filme tem informações como título e duração. Sempre que um filme for adquirido deverá ser registrado;
  - Um filme pode ter vários atores.



Exercício 1

- Identifique os objetos no problema abaixo:
  - Um clube tem muitos sócios e precisa manter informações referente a eles, como o número do seu cartão de sócio, endereço, telefone e e-mail.
  - Um sócio pode ter nenhum ou vários dependents.
  - Um sócio deve pagar mensalidades para poder frequentar o clube. Serão cobrados juros sobre o valor da mensalidade relativos ao atraso do pagamento. As informações pertinentes a cada mensalidade são a data de pagamento, o valor, a data em que foi efetivamente paga e juros aplicados.



Projeto Orientado a Objetos

As principais etapas envolvidas são:

1

Identificação dos objetos envolvidos com o sistema a ser desenvolvido e sua representação em forma de **classes**

2

Identificação de suas características relevantes e sua representação em forma de **atributos**

3

Identificação de ações realizadas por esses objetos e sua representação em forma de **métodos**

Projeto Orientado a Objetos

A unidade fundamental de programação no Java é a *classe*

Classes fornecem a estrutura para os *objetos* e os mecanismos para "fabricar" objetos a partir de uma definição de classe

Classes definem os *atributos* e *métodos* (*membros*):

Atributos: características relevantes dos objetos	**Métodos**: coleção de código executável que é o foco da computação e que manipulam os dados armazenados nos objetos	**Métodos** fornecem o **comportamento** dos objetos de uma classe

Sistemas de Informação | FIAP
Prof. Dr. Antonio Marcos SELMINI – selmini@fiap.com.br

Projeto Orientado a Objetos

Objetos

Qualquer entidade do mundo real que apresente algum significado, **mesmo que não se constitua em algo concreto.**

Todo objeto possui características próprias → **atributos.**

Atributos permitem distinguir objetos de classes diferentes.

Objetos manifestam comportamentos → **métodos.**

- Objetos podem ser:
  - **Concretos**: pessoas, carros, etc.
  - **Abstratos**: círculo, elipse, etc.

Projeto Orientado a Objetos

Classes

Uma vez identificados os objetos relevantes e seus atributos, estes devem ser agrupados em **classes.**

- **Quando um conjunto de objetos possuem atributos comuns significa que eles pertencem a uma mesma categoria (mesma classe).**

Classe é um sinônimo de **categoria.**

Classe representa um grupo de objetos com características comuns e compõem-se de atributos e métodos.

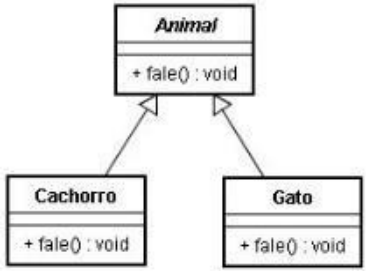
Projeto Orientado a Objetos – resumo

abstração

Projeto orientada a objetos: diagramar o modelo dos objetos observados

representação

UNIFIED MODELING LANGUAGE



Análise orientada a objetos: analisar o problema e decompor em estruturas abstratas

Programação orientada a objetos: "implementar as estruturas da modelagem em código de computador usando uma linguagem que suporta a orientação a objetos"

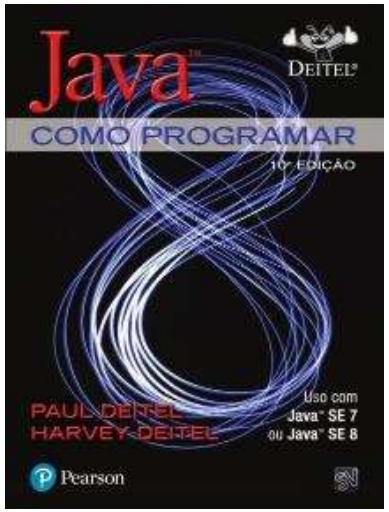
```
public class Cachorro {
//Atributos ou propriedades
//Métodos ou operações
}
```


Prof. Dr. Antonio Marcos SELMINI – selmini@fiap.com.br

Bibliografia

- DEITEL, H. M., DEITEL, P. J. **JAVA como programar**. 10ª edição. São Paulo: Prentice-Hall, 2010.

- SCHILDT, H. **Java para Iniciantes – Crie, Compile e Execute Programas Java Rapidamente**. 6ª Edição, Editora Bookman, Porto Alegre, RS, 2015.



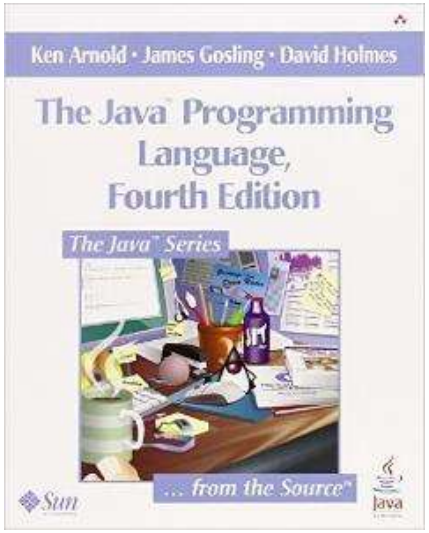
Bibliografia

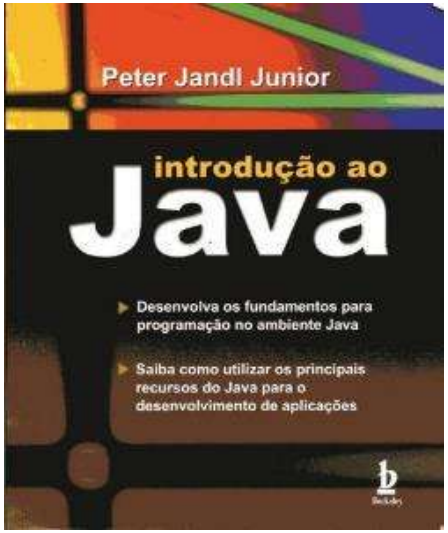
- KNUDSEN, J., NIEMEYER, P. **Aprendendo Java**. Rio de Janeiro: Editora Elsevier Campus, 2000.
- FLANAGAN, D. **Java – o guia essencial**. Porto Alegre: Editora Bookman, 2006.

Bibliografia

- ARNOLD, K., GOSLING, J., HOLMES, D., **Java programming language**. 4th Edition, Editora Addison-Wesley, 2005.

- JANDL JUNIOR, P. **Introdução ao Java**. São Paulo: Editora Berkeley, 2002.





REFERÊNCIAS

- ARNOLD, K., GOSLING, J., HOLMES, D., **Java programming language**. 4th Edition, Editora Addison-Wesley, 2005.

- JANDL JUNIOR, P. **Introdução ao Java**. São Paulo: Editora Berkeley, 2002.