Metodologia e Linguagem de Programação II

PROF: RODRIGO DA CRUZ FUJIOKA

Check List – Ementa

1. ~~Histórico e evolução da linguagem Java~~
2. ~~Arquitetura da tecnologia Java~~
3. ~~Características da linguagem~~
4. ~~Estado da arte em ambientes de desenvolvimento e execução~~
5. ~~Escrevendo, compilando e executando aplicações Java~~
6. ~~Entrada padrão de dados~~
7. ~~Saída padrão de dados~~
8. ~~Entrada/Saída de dados GUI~~

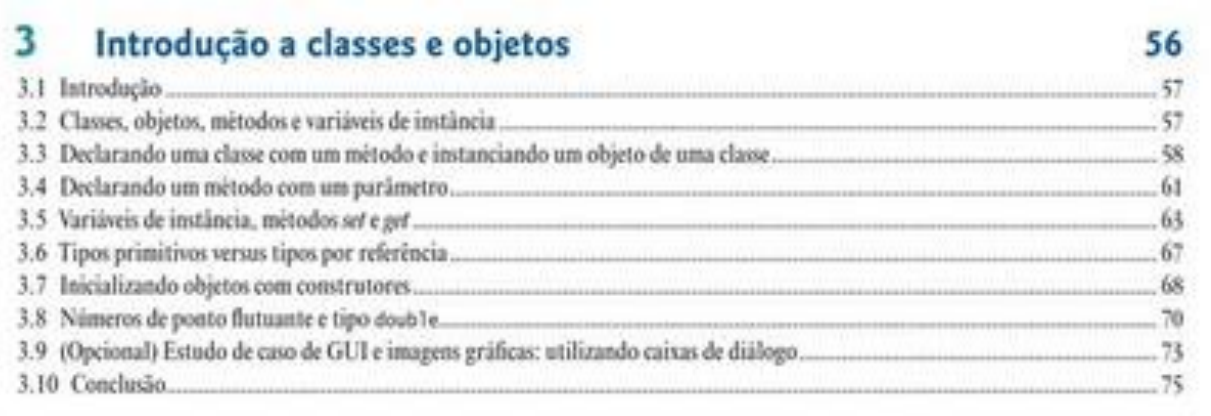
Segunda parte.

- Visão geral de conceitos de POO
- ~~Abstração~~
- Objetos
- Ciclo de vida dos objetos (instanciação à destruição)
- Classes
- Membros de uma classe: atributos e métodos
- Interface de objetos
- Construtores e suas características
- Encapsulamento
- Varáveis e métodos de classe
- Herança e Polimorfismo
- Sobrecarga e sobreposição de métodos
- Documentando programas Java

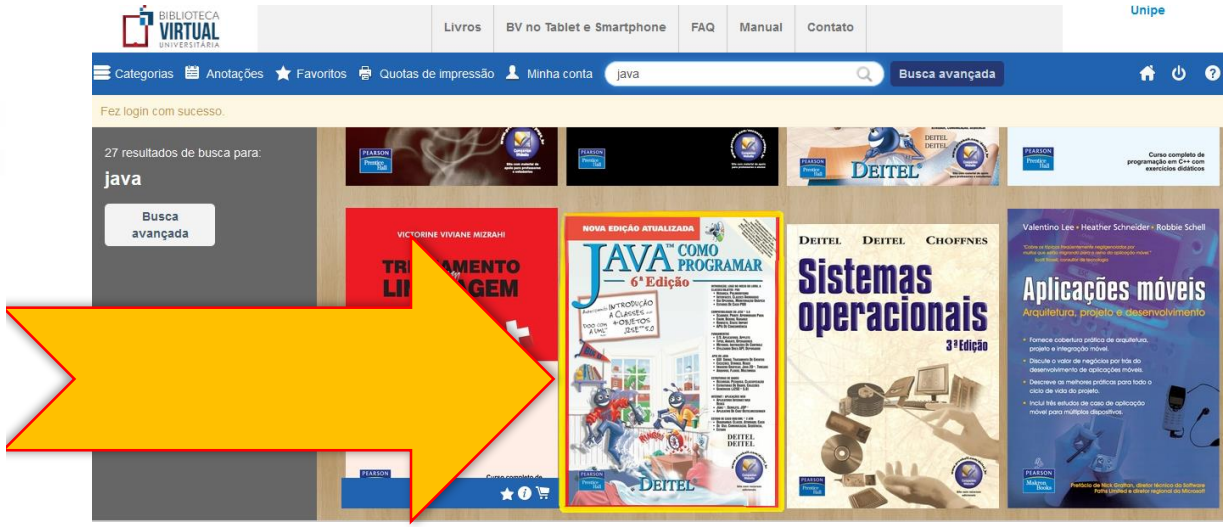
Assunto pode ser encontrado em:

Entendo um pouco mais:

Livro: Java Como Programa 8° Edição
Link: Esta no ambiente de EAD.
Páginas: 55 - 75

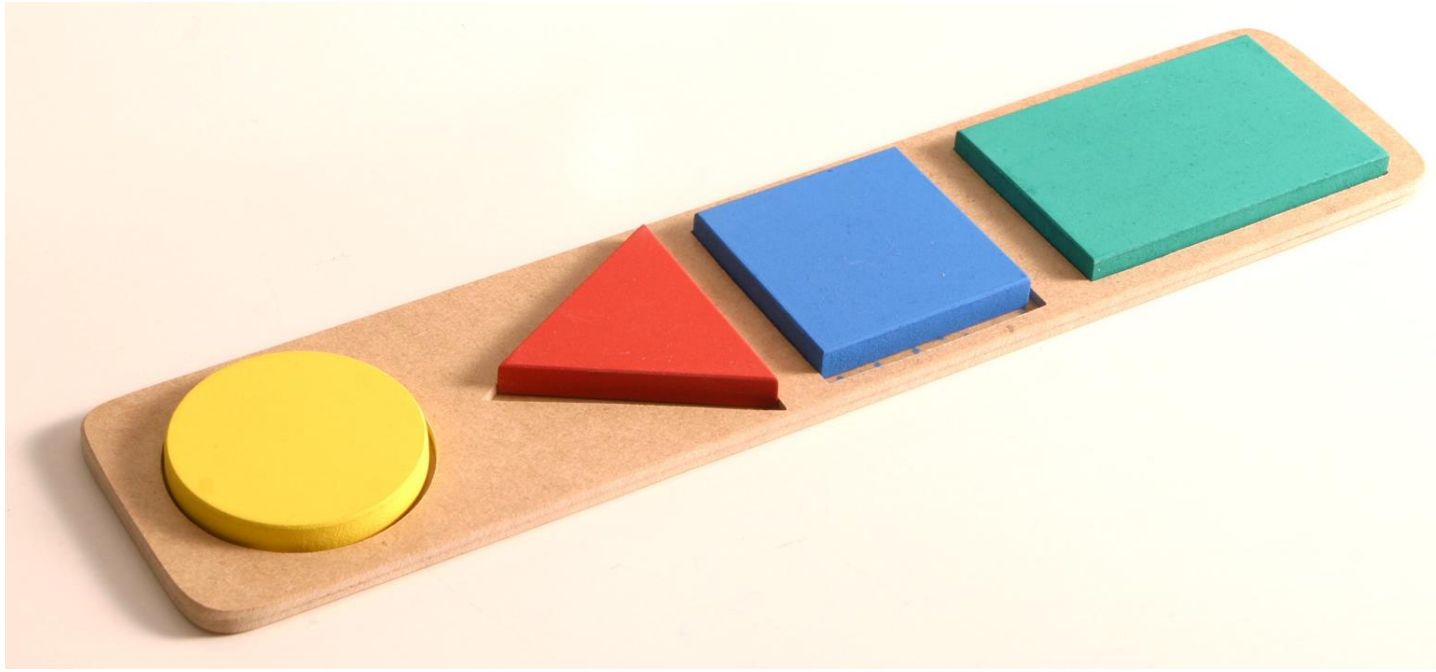


3 Introdução a classes e objetos 56



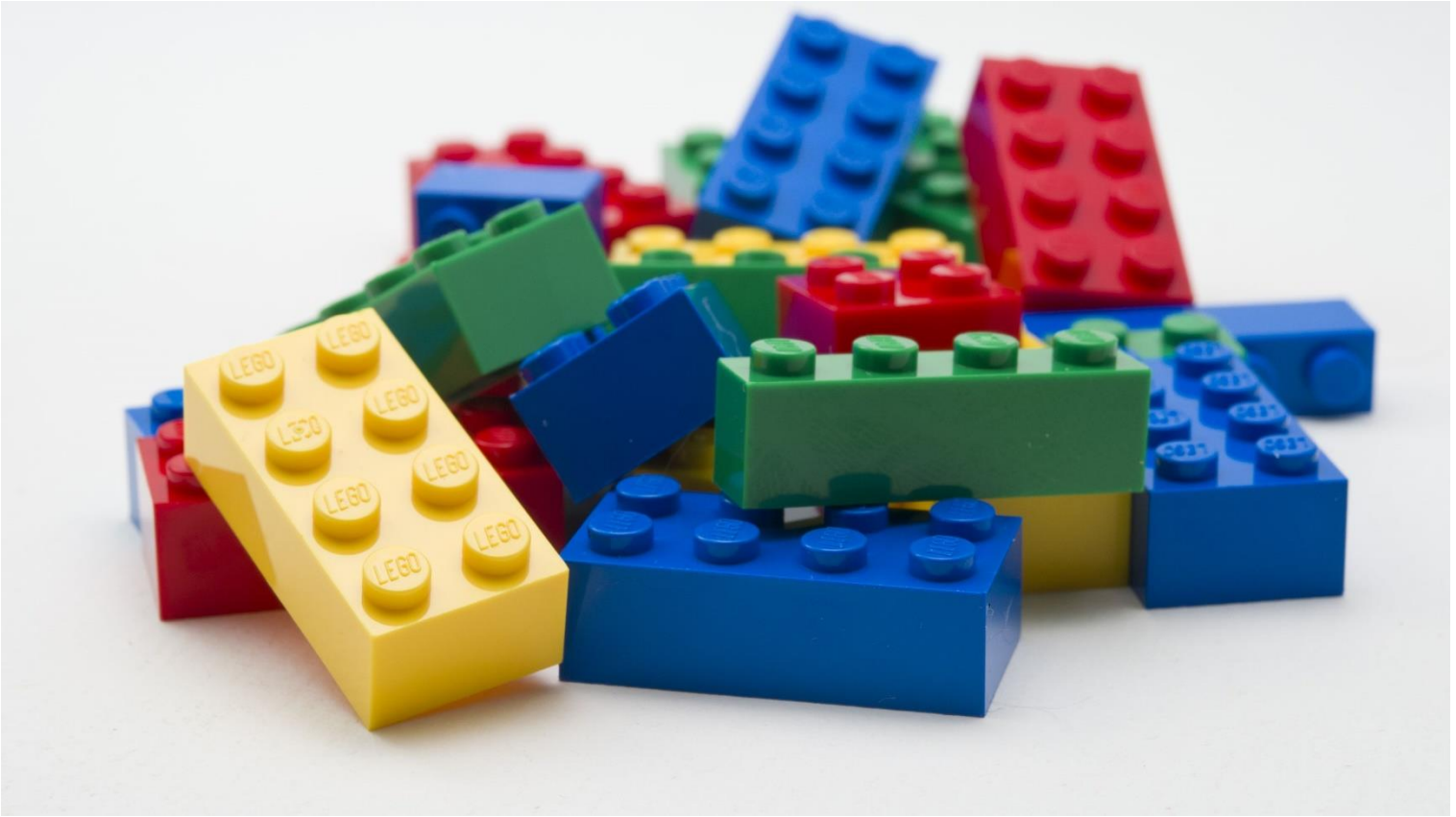
Visão geral da POO (Programação Orientada a Objetos)

O que é um objeto?



Visão geral da POO (Programação Orientada a Objetos)

O que é um objeto?

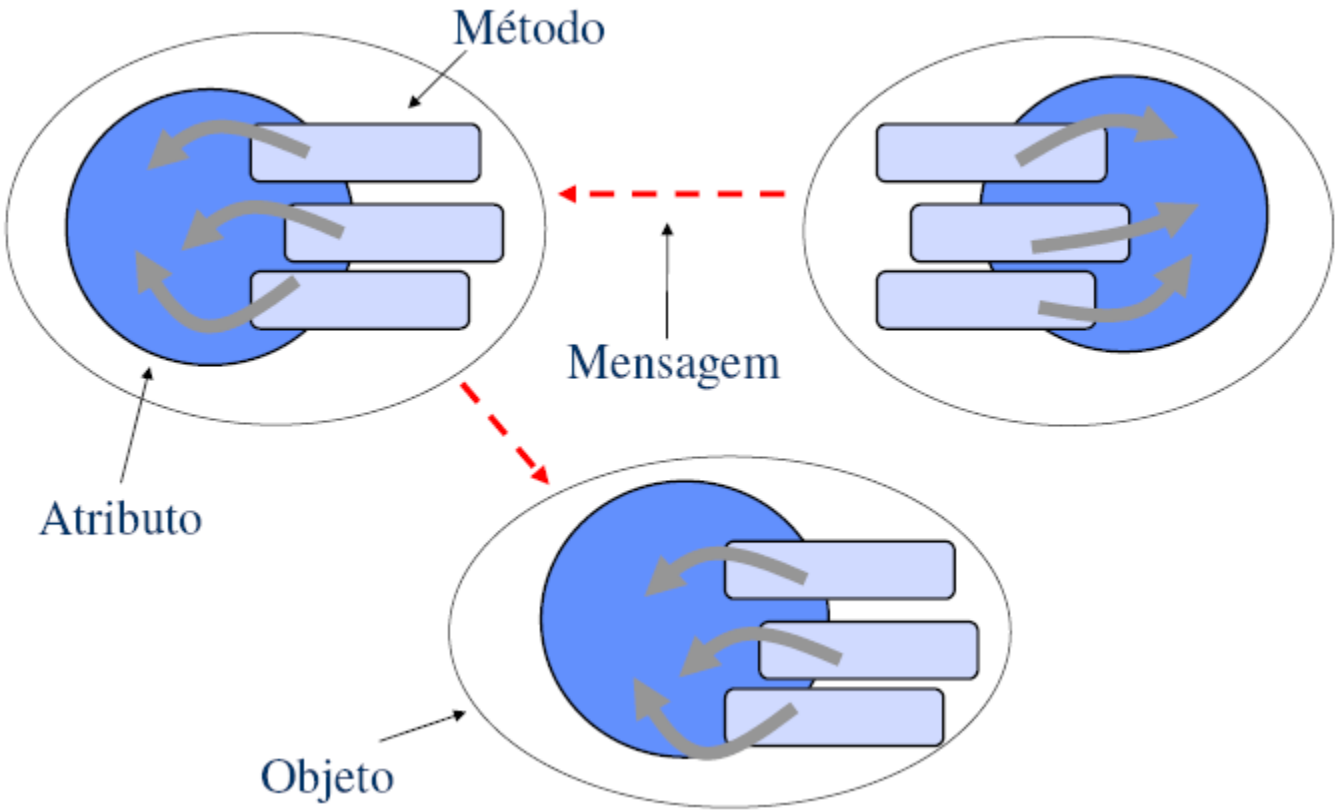


Visão geral da POO (Programação Orientada a Objetos)

- Descrição de um Objeto
  - Um objeto pode ser descrito por um conjunto de atributos e comportamentos

Atributos	Comportamentos
Motor	Avançar
Cor	Retroceder
Potência	Parar
Rodas	Abastecer
Velocidade	

Visão geral da POO (Programação Orientada a Objetos)



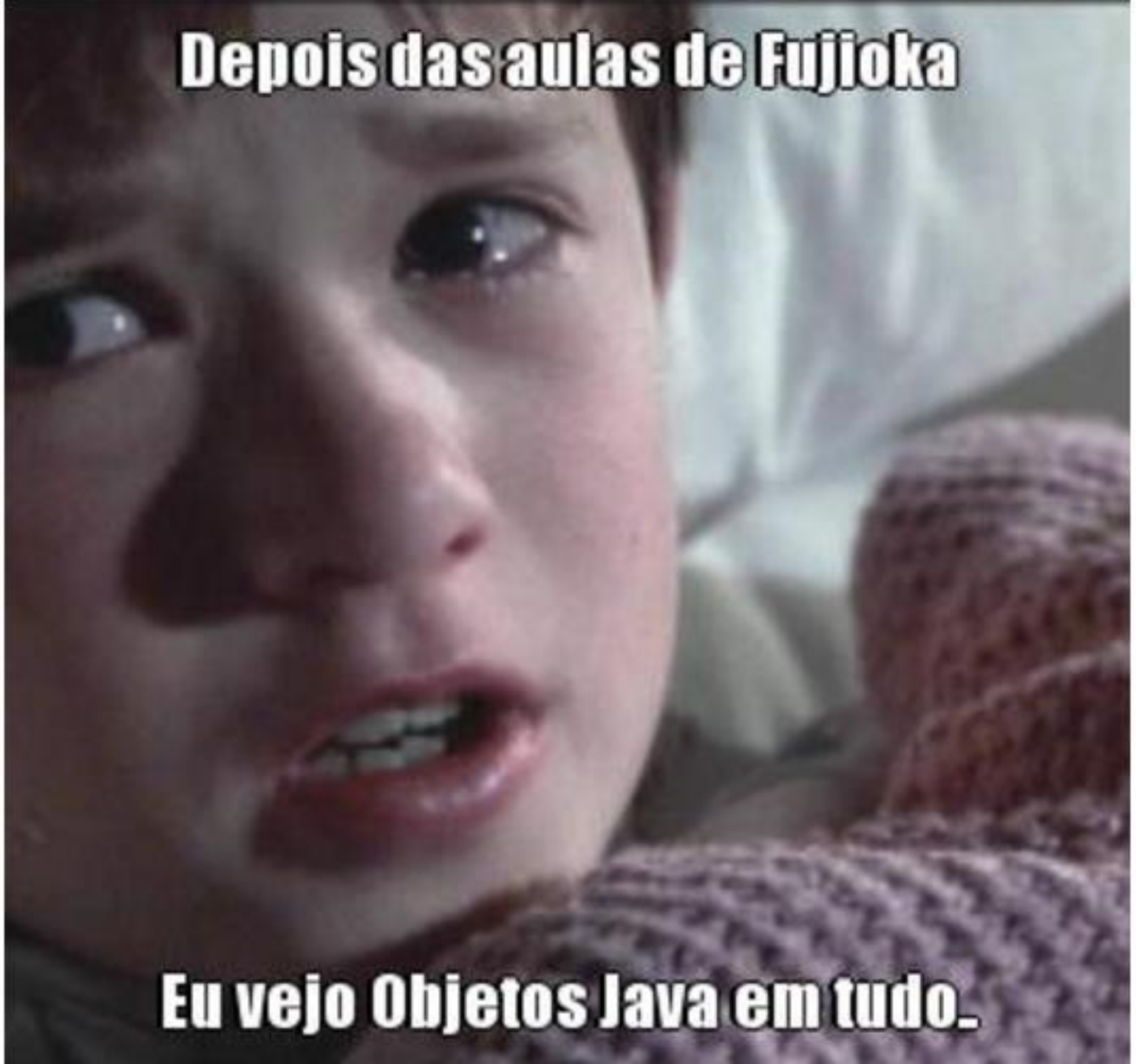
Classe Java

```java
public class ObjetoJapones {

    private String nomeOjeto;
    private String corObjeto;
    private String materialObjeto;

    public String getNomeOjeto() {
        return nomeOjeto;
    }
    public void setNomeOjeto(String nomeOjeto) {
        this.nomeOjeto = nomeOjeto;
    }
    public String getCorObjeto() {
        return corObjeto;
    }
    public void setCorObjeto(String corObjeto) {
        this.corObjeto = corObjeto;
    }
    public String getMaterialObjeto() {
        return materialObjeto;
    }
    public void setMaterialObjeto(String materialObjeto) {
        this.materialObjeto = materialObjeto;
    }
}
```

Conceitos de POO


Depois das aulas de Fujioka
Eu vejo Objetos Java em tudo..

O operador “.”

Ele é utilizado para acessar os **dados** ou **invocar** os **métodos** de um objeto;

Quando após o operador houver apenas um nome, ele está acessando os dados (as variáveis de instância);

Quando após o operador houver um nome e parênteses **()** no fim ele está invocando as ações (os métodos);

O que é o “new”?

É o operador utilizado para criar objetos;

Após o “new” deve vir o nome de uma classe;

O operador new retorna um objeto contruido para ser utilizado;

```java
public class ObjetoJapones {

    private String nomeObjeto;
    private String corObjeto;
    private String materialObjeto;

    public String getNomeObjeto() {
        return nomeObjeto;
    }
    public void setNomeObjeto(String nomeObjeto) {
        this.nomeObjeto = nomeObjeto;
    }
    public String getCorObjeto() {
        return corObjeto;
    }
    public void setCorObjeto(String corObjeto) {
        this.corObjeto = corObjeto;
    }
    public String getMaterialObjeto() {
        return materialObjeto;
    }
    public void setMaterialObjeto(String materialObjeto) {
        this.materialObjeto = materialObjeto;
    }
}
```

```java
ObjetoJapones variavelReferenciaUm = new ObjetoJapones();
ObjetoJapones variavelReferenciaDois = new ObjetoJapones();
ObjetoJapones variavelReferenciaTres = new ObjetoJapones();
```

Programação Orientada a Objetos

- Quando executado, um método pode alterar o estado (atributos) do objeto.
- Um objeto pode ser composto de outros objetos.
- Um programa corresponde a um conjunto de objetos enviando mensagens uns aos outros, para juntos atingirem um comportamento de mais alto nível.
  - Ex: Avião (motor e asa não voam)
- Todo objeto possui um tipo.
- Todos os objetos de um tipo particular podem receber as mesmas mensagens.

Programação Orientada a Objetos

- O que é um Objeto?
  - "Alguma coisa que faz sentido no domínio da aplicação"
  - Uma abstração
- Utilidade:
  - Facilita a compreensão
  - Oferece base real para implementação no computador

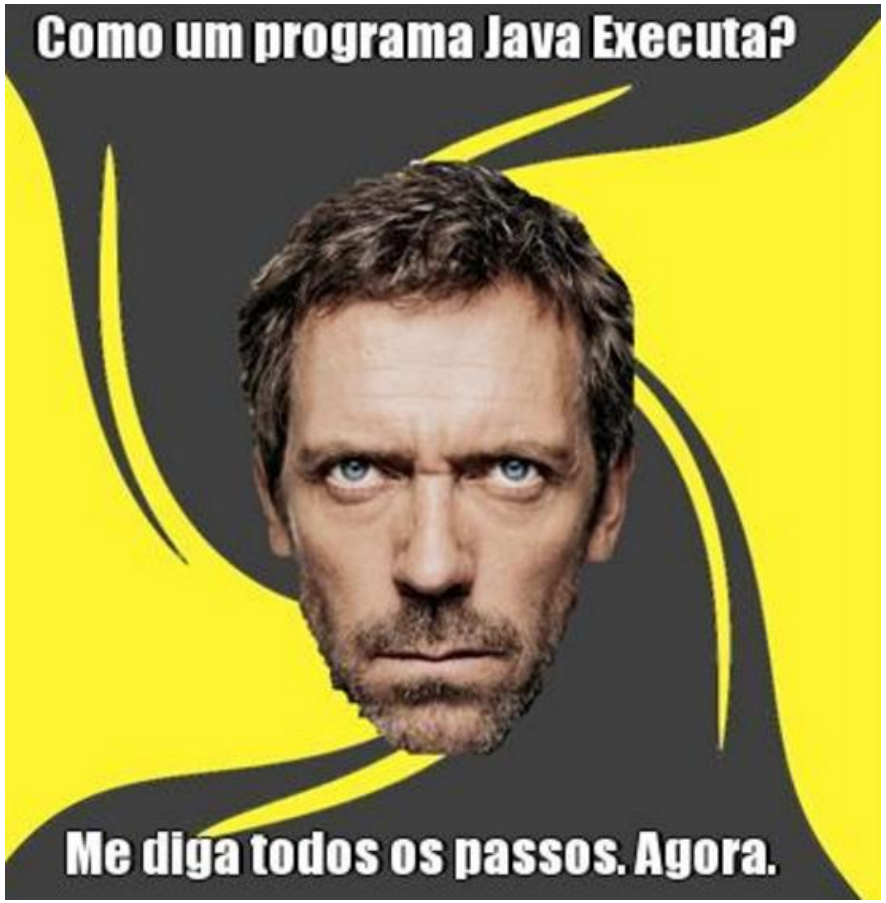
- Descrição de um Objeto
  - Um objeto pode ser descrito por um conjunto de atributos e comportamentos

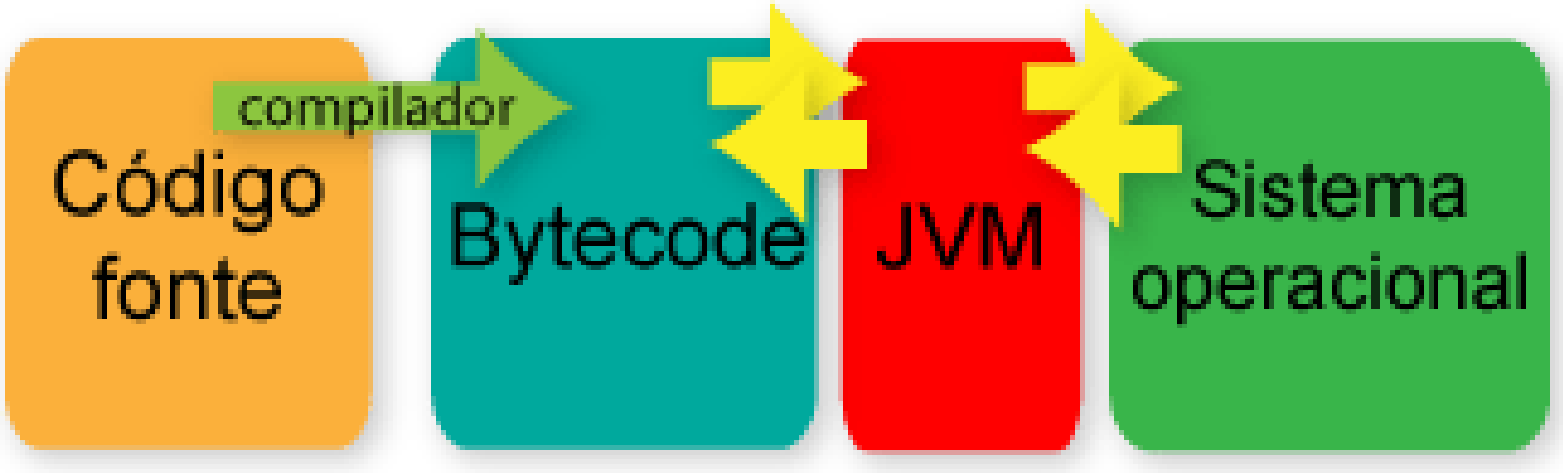
Atributos	Comportamentos
Motor	Avançar
Cor	Retroceder
Potência	Parar
Rodas	Abastecer
Velocidade	

Dúvidas?


Então?
Sacou os paranauê?

Java – Compilador JIT (Just-in-Time)

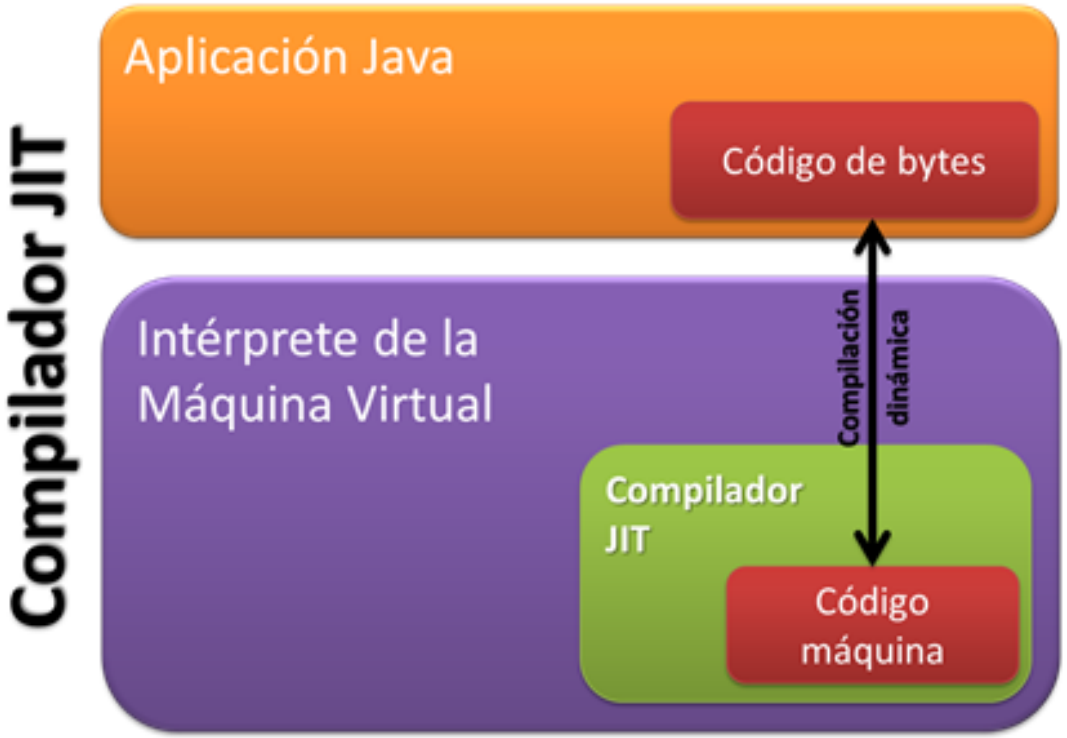



compilador
Código fonte
Bytecode
JVM
Sistema operacional

Compilador JIT (Just-in-Time)

“Compiladores Just-In-Time foram o próximo passo no desenvolvimento do Java. Compiladores Just-In-time são compiladores que traduziam enquanto o programa estava sendo executado. Esses compiladores nada mais fazem do que traçar o perfil do programa em execução para descobrir onde estão os métodos principais do programa, e depois os compilam para o conjunto de instruções nativo em que a máquina virtual está executando. A parte compilada é salva para a próxima vez em que o programa for executado, de modo que a execução possa ser feita mais rapidamente da próxima vez em que ele for executado”

Leia mais em: Processo de tradução e execução de programas Java http://www.devmedia.com.br/processo-de-traducao-e-execucao-de-programas-java/26915#ixzz41WlAjdaN



Atributos e Métodos.

Métodos com Retorno e Sem Retorno

Objetos

Um objeto representa uma entidade real ou abstrata, com um papel bem definido no domínio do problema que se quer resolver / solucionar

- Os objetos podem ser físicos...
  - *Bicicleta, máquina de escrever, pessoa, sensor,...*
- Ou abstratos...
  - *Fluxograma, venda, ...*

Objetos

- Estado:

– Propriedades: característica que contribui para distinguir um objeto dos demais.

– Valores: todas as propriedades possuem valores que podem variar ao longo do tempo.

Objetos

- Comportamento:

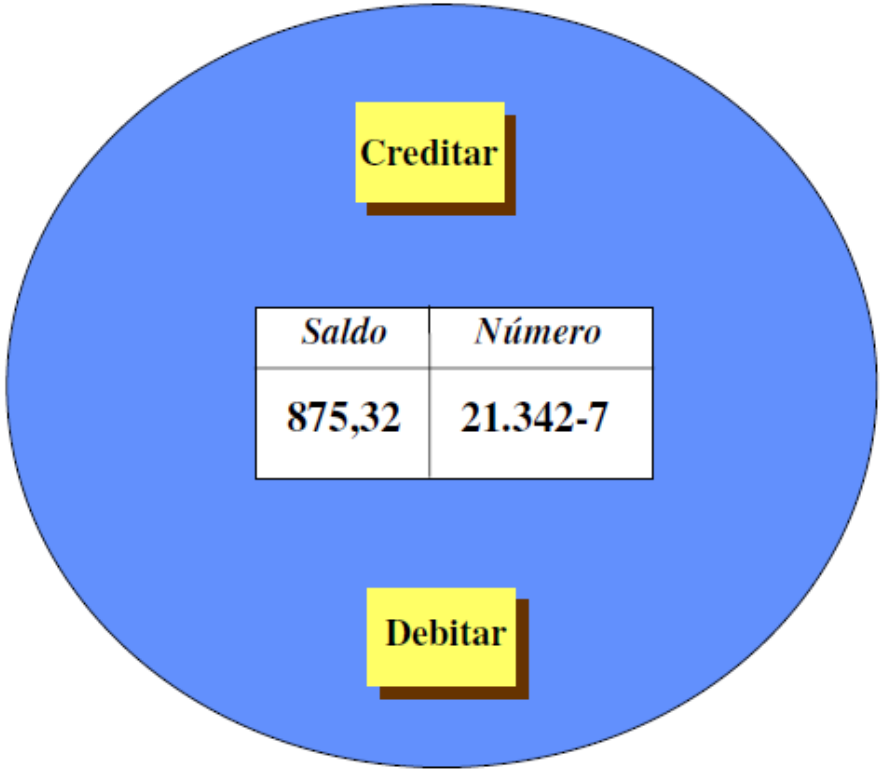
– O comportamento de um objeto é caracterizado pelo conjunto de operações que o objeto é capaz de executar

– Podemos imaginar os objetos como entidades capazes de prestar serviços

– Para um objeto solicitar um serviço de outro objeto, deve enviar-lhe uma mensagem

– Se o objeto receptor for capaz de prestar este serviço, ele será executado

Objetos



Objetos - Propriedades

- Propriedades do objeto ônibus:
  - Velocidade
  - Número de Lugares
  - Marca
- Propriedades do objeto poltrona:
  - Cor
  - Número de Lugares
  - Marca

Objetos - Comportamento

- Comportamentos do objeto ônibus:
  - Acelerar
  - Frear
  - Receber passageiro
  - Virar a direita
- Comportamentos do objeto poltrona:
  - Mover
  - Acomodar Pessoa
  - Reclinar Encosto

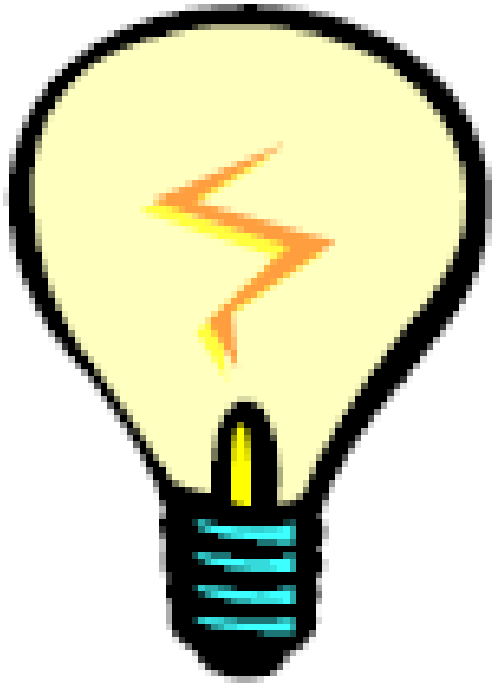
- Exemplo:

– Suponha uma abstração para uma lâmpada que deva exibir o seguinte comportamento:

- ✓ Informar se está acesa
- ✓ Acender
- ✓ Apagar

– O estado de uma lâmpada diz se ela está ou apagada.

Objeto - exemplo



Objetos - Exemplo

```
class Lampada {

    boolean estaAcesa = false;

    //Método que informa se esta acesa ou apagada.
    boolean getAcesa() {
        return estaAcesa;
    }

    //Método acender
    void acender() {
        estaAcesa = true;
    }
    //Método apagar
    void apagar() {
        estaAcesa = false;
    }
}
```

Objetos - Exemplo

```
public class UsaLampada {

    public static void main(String[] args) {

        Lampada l1, l2;
        l1 = new Lampada();
        l2 = new Lampada();

        l1.acender();
        l2.apagar();

        if (l1.getAcesa()) {
            System.out.println("Lampada l1 esta ligada");

        } else {

            System.out.println("Lampada l1 esta apagada");
        }

    }

}
```

Objetos - Exemplo

```
public class UsaLampada {

    public static void main(String[] args) {

        Lampada l1, l2;
        l1 = new Lampada();
        l2 = new Lampada();

        l1.acender();
        l2.apagar();

        if (l1.getAcesa()) {
            System.out.println("Lampada l1 esta ligada");

        } else {

            System.out.println("Lampada l1 esta apagada");
        }

    }

}
```

Objetos - Exemplo

- Identidade

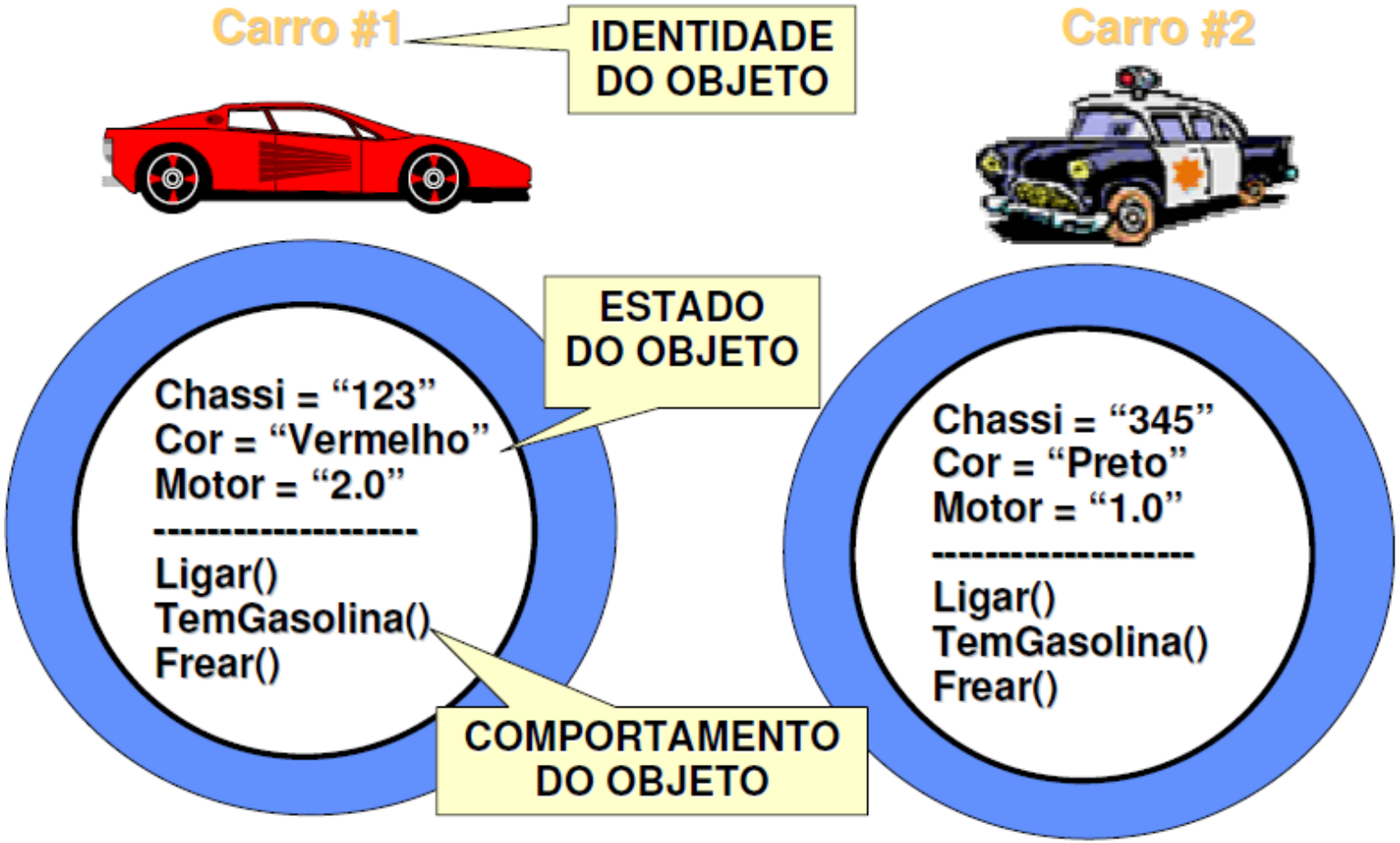
– Cada objeto é único e distinguível dos demais independentemente dos valores dos seus atributos

– Para manipular os objetos, utilizamos referências (Exemplo do controle remoto) para os mesmos

– Uma referência nada mais é do que uma variável que “aponta” para um objeto do tipo declarado

– Objetos são criados através de um processo chamado de instanciação, que, por sua vez, devolve um “ponteiro”
para o objeto criado.

Objetos - Identidade



Objetos – Ciclo de vida

1) Instanciação: o objeto é criado e passa a ser referenciado por um nome (sua referência)

– Referências
acessados e
e “tipificadas” em tempo de compilação.

Objetos – Ciclo de vida

2) Uso: o objeto recebe mensagens de outros objetos e, com isso, executa parte da funcionalidade do sistema.

3) Destruição: a área de memória ocupada pelo objeto é liberada quando não existem mais referências apontando para o objeto

– Em Java a destruição de um objeto é realizada automaticamente pelo coletor de lixo!

– Minimiza o trabalho do desenvolvedor

Objetos - Instanciação

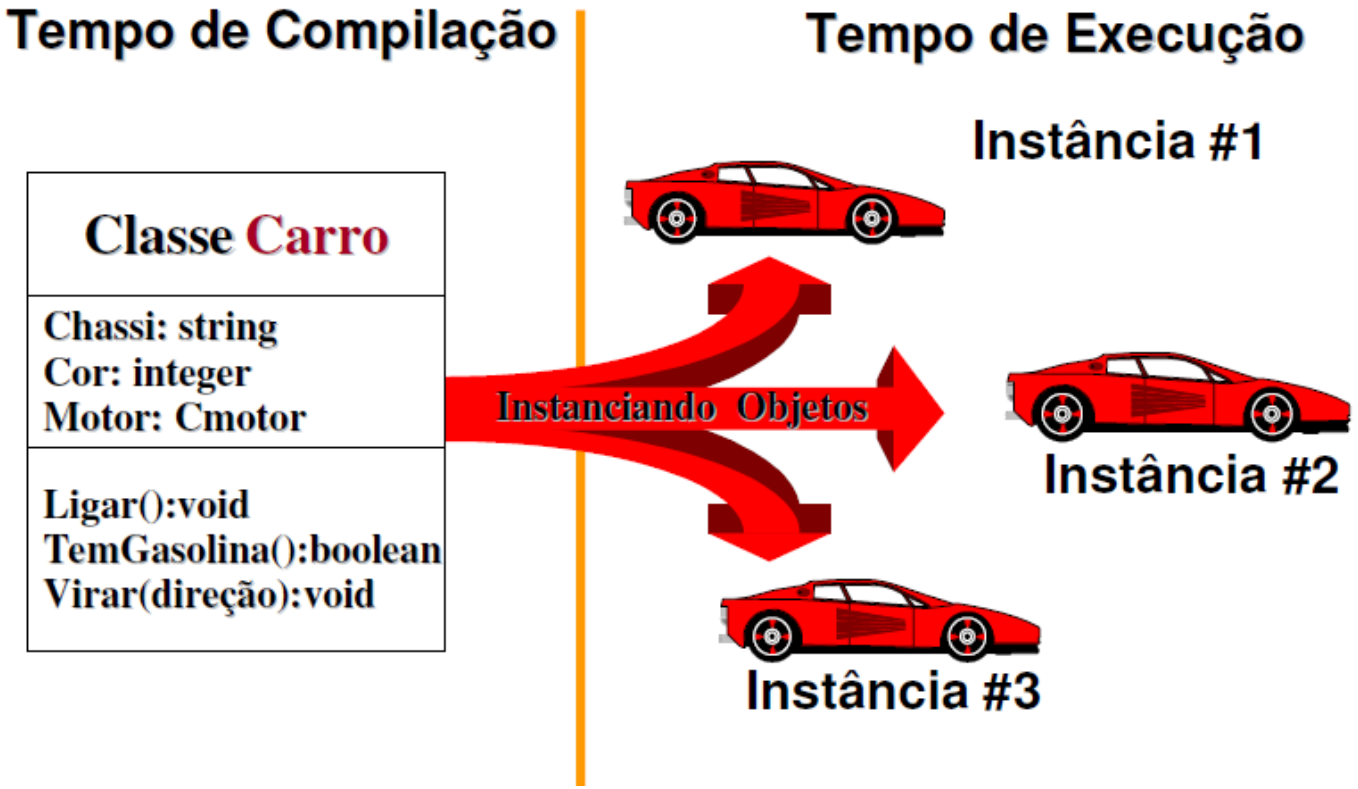
Definição

- Operação através da qual um objeto é criado e passa a existir na memória do computador.

A alocação de objetos na memória ocorre dinamicamente.

A classe é o modelo utilizado para criar o objeto.

Objetos - Instanciação



Objetos – Ciclo de vida

```
Quadrado q;

q = new Quadrado();

q.moverDireita(20);
q.mudarTamanho(5);
q.ficarVisivel(false);
q.mudarCor("red");

q = null;
```

1)

2)

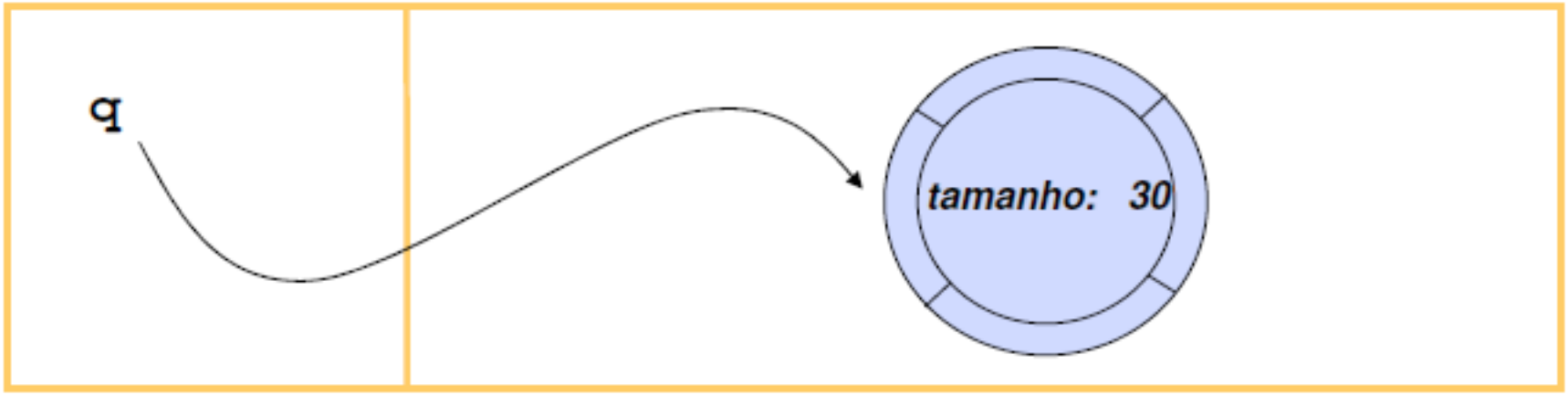
3)

Objetos – Referencias

```
Quadrado q;                 // declaração
q = new Quadrado();         // instanciação
q.mudeTamanho(50);          // uso
q = null;                   // destruição
```

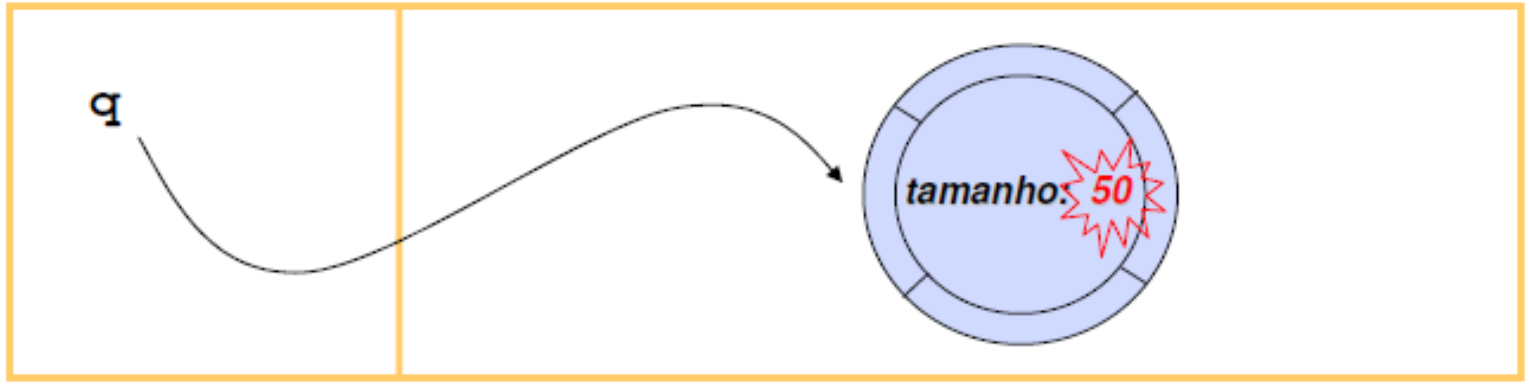
Obietos – Referencias

```
Quadrado q;                 // declaração
q = new Quadrado();         // instanciação
q.mudeTamanho(50);          // uso
q = null;                   // destruição
```



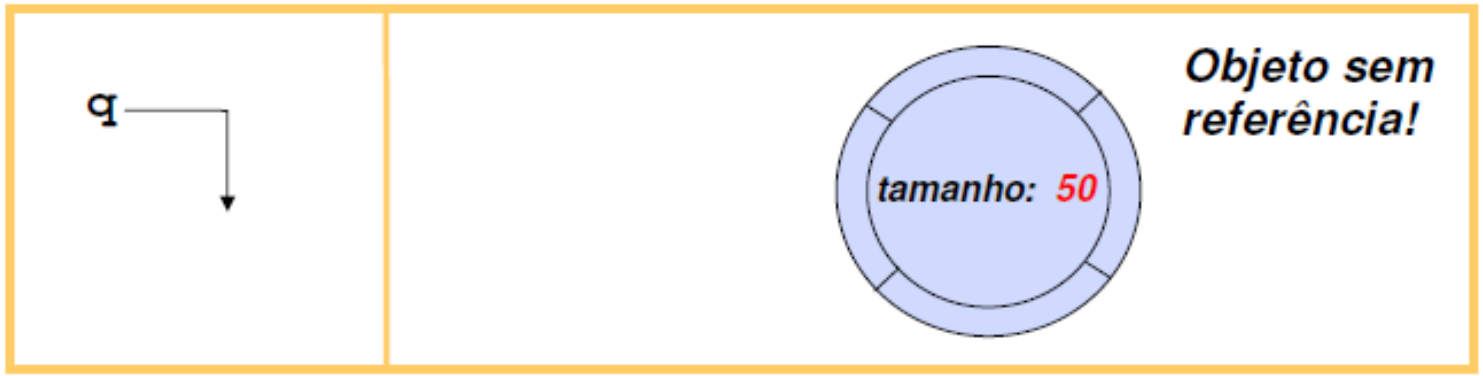
Objetos – Referencias

```
Quadrado q;              // declaração
q = new Quadrado();      // instanciação
q.mudeTamanho(50);       // uso
q = null;                // destruição
```



Objetos – Referencias

```
Quadrado q;              // declaração
q = new Quadrado();      // instanciação
q.mudeTamanho(50);       // uso
q = null;                // destruição
```



Dúvidas ?

? ? ? ? ? ? ? ?

? ? ? ? ? ? ? ?

? ? ? ? ? ? ? ?

? ? ? ? ? ? ? ?

? ? ? ? ? ? ? ?

Classes

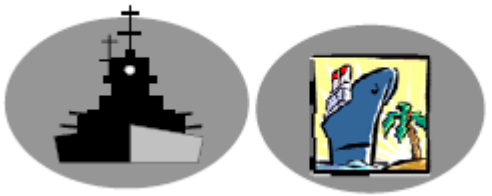
- Definição:

– Local onde encontram-se definidas as propriedades (variáveis) e o comportamento (métodos) que uma categoria de objetos deve possuir

Classes

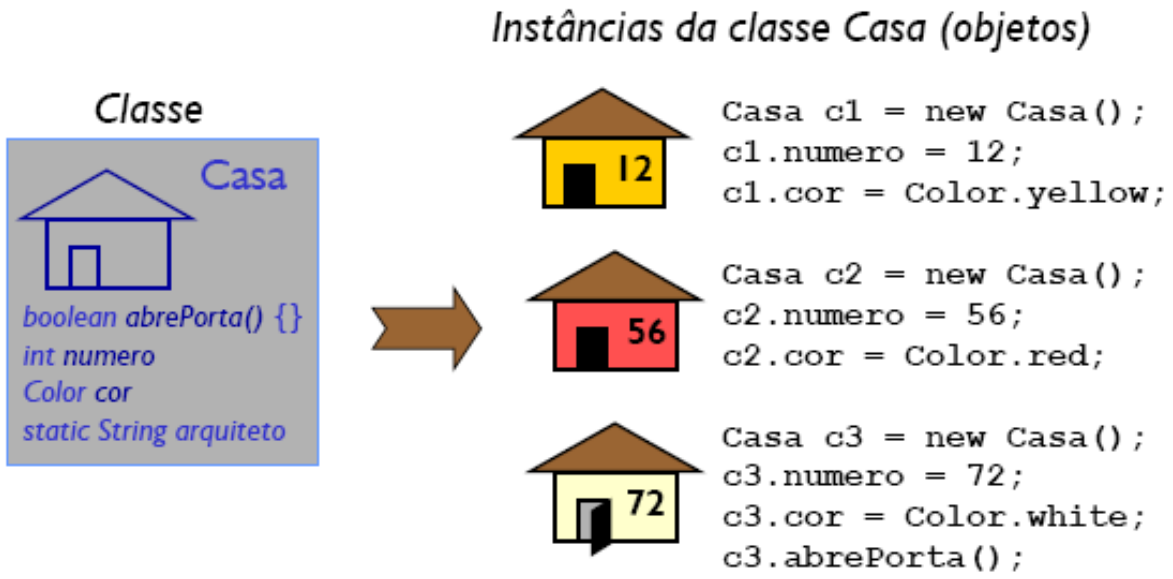
- Definição:

– Local onde encontram-se definidas as propriedades (variáveis) e o comportamento (métodos) que uma categoria de objetos deve possuir



Classes

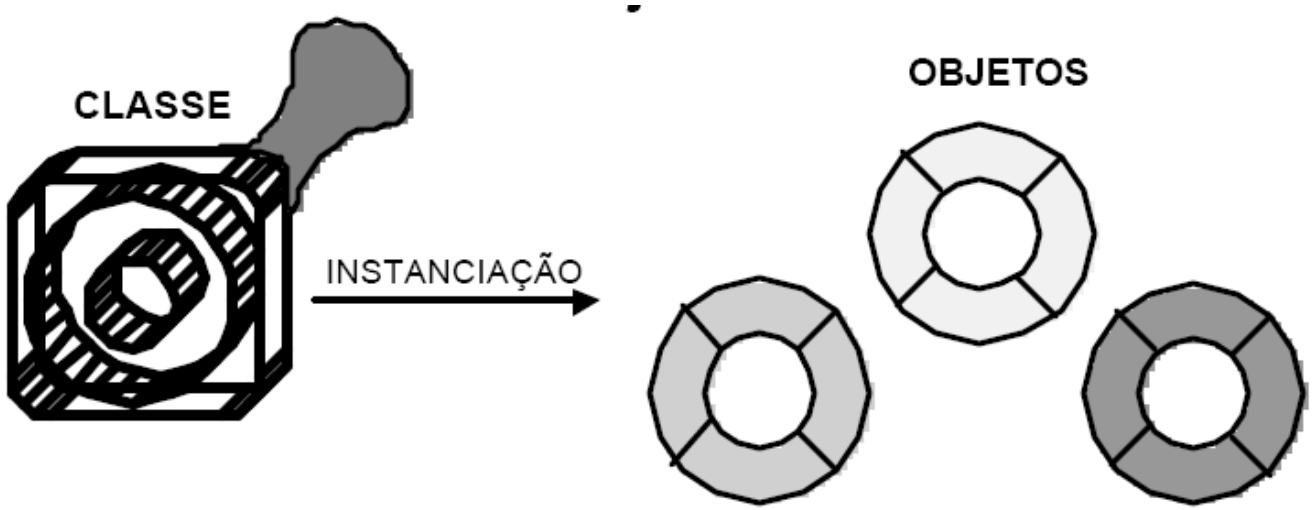
- *Classes são uma especificação para objetos*
- *Uma classe representa um tipo de dados complexo*
- *Classes descrevem*
  - *Tipos dos dados que compõem o objeto (o que podem armazenar)*
  - *Procedimentos que o objeto pode executar (o que podem fazer)*



Classes

Toda classe define **um novo tipo** de dados

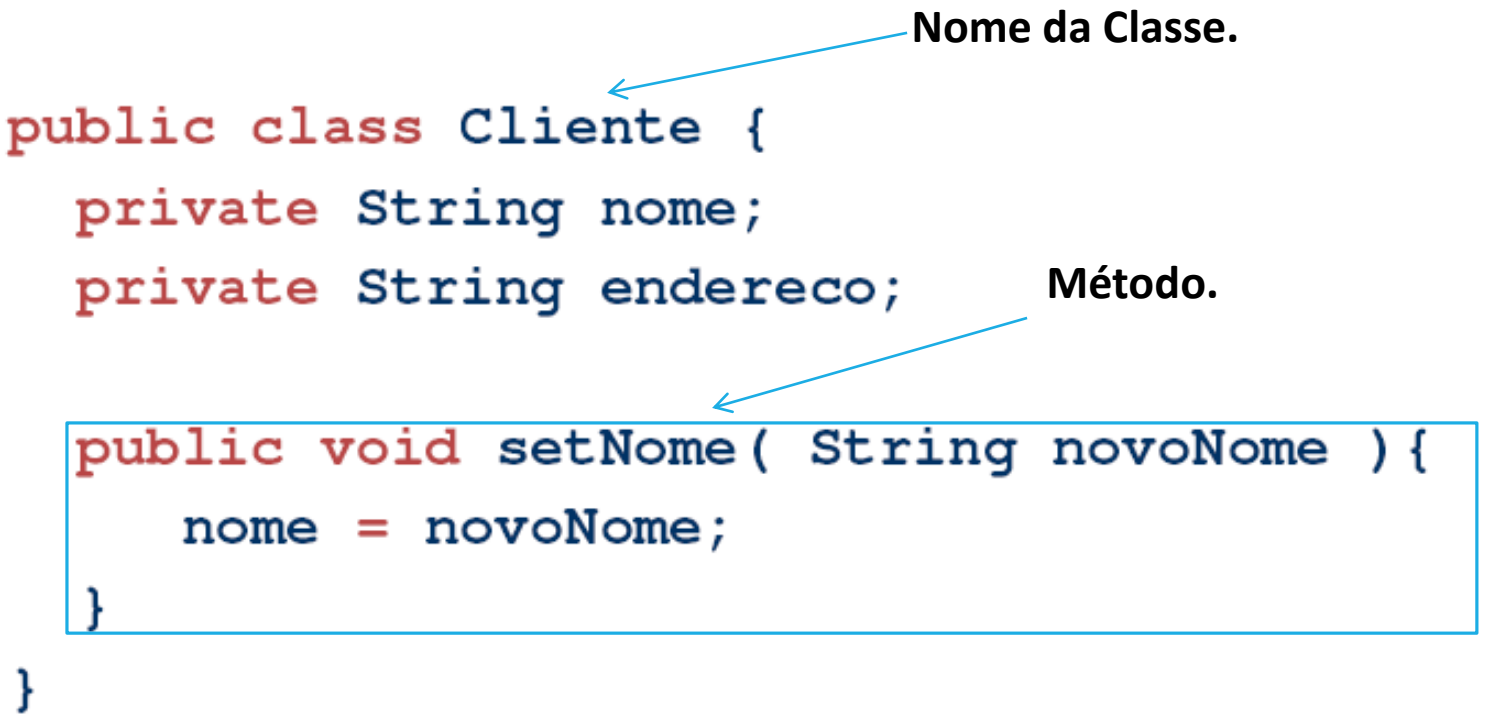
Valores de um tipo definido por uma classe recebem o nome de **objetos**



Classes

- Estrutura fundamental de programação em Java!
  - Todo e qualquer programa Java deve definir pelo menos uma classe
  - Não há como escrever código Java sem que haja a definição de classes

Classes – Exemplo

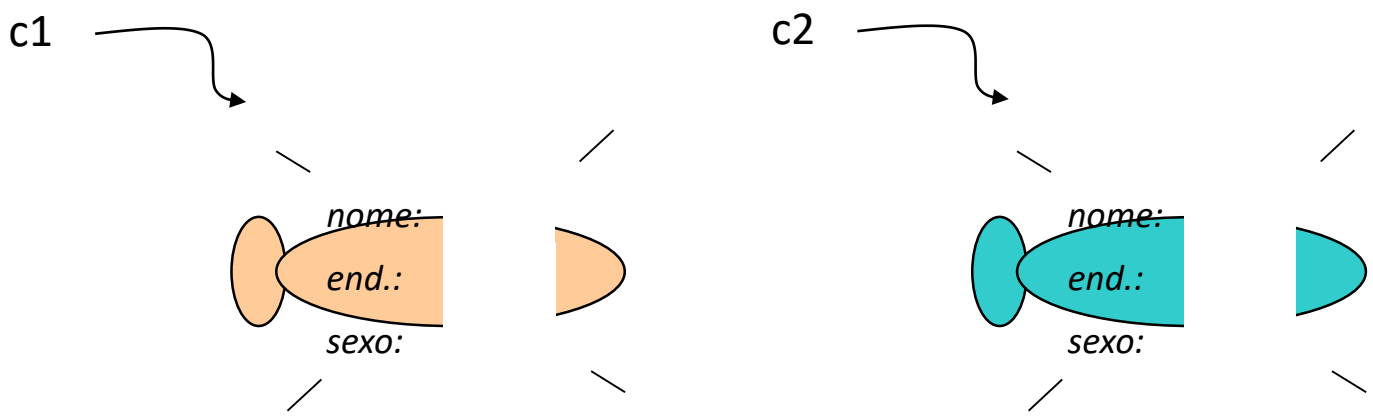



```
public class Cliente {
   private String nome;
   private String endereco;

   public void setNome( String novoNome ){
      nome = novoNome;
   }
}
```

Instanciação da classe cliente:

```
Cliente c1 = new Cliente();
Cliente c2 = new Cliente();
```



Classes – Padrão Nomenclatura

- O nome da classe deve começar com letra maiúscula.
  - Ex: Conta, Cliente, Banco, Endereco.
- Nomes compostos não são separados por _. A primeira letra da palavra seguinte é maiúscula.
  - Ex: PessoaJuridica, PessoaFisica
- Evite abreviações no nome e use nomes com alguma relação com o que a classe modela

Métodos

- Realizam leitura (getX()) do valor dos atributos de um objeto

- Alteram o valor dos atributos (setX())

- A implementação de um método pode utilizar as estruturas lógicas da programação estruturada: decisão, laços, desvio de
fluxo, atribuição, etc.

Métodos

Um método é um procedimento ou função que permite aos objetos de uma classe executarem serviços

É como o objeto implementa suas funcionalidades

O envio de uma mensagem para um objeto faz o método correspondente (de mesmo nome) ser executado

A mensagem é a ativação de um método sobre o objeto

Métodos

Métodos podem ter parâmetros

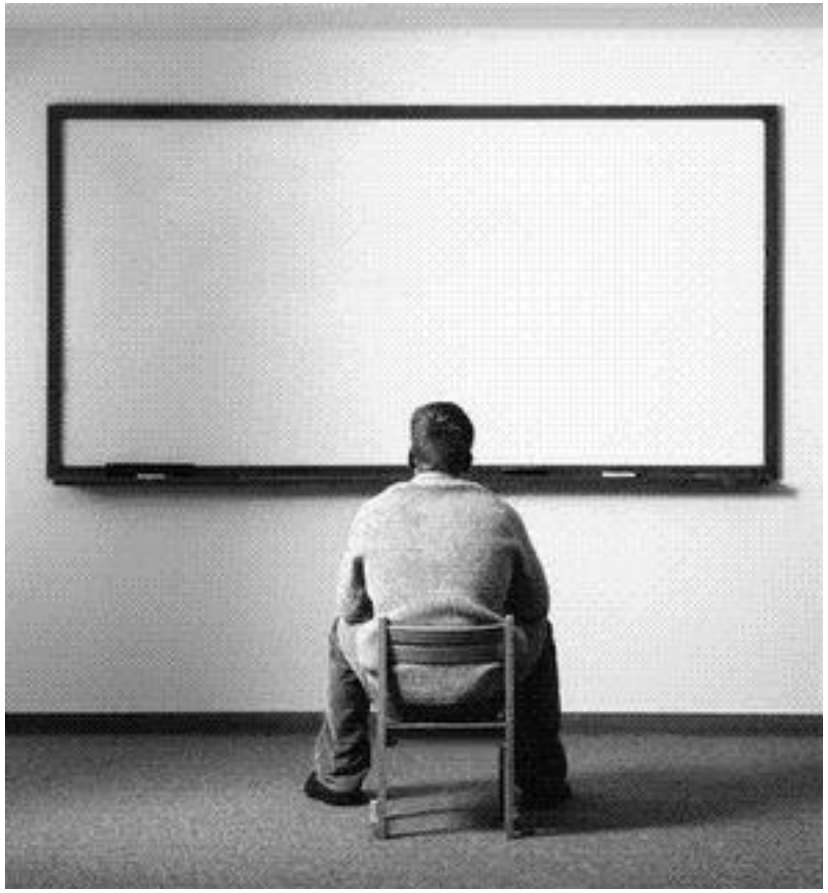
Métodos poder ler (getXXX()) ou alterar (setXXX()) os valores das propriedades dos objetos

- Modificam, portanto, o estado de um objeto
- Estes métodos, juntos, representam o comportamento de um objeto ou o protocolo de uma classe

A implementação de um método utiliza todas as estruturas lógicas da programação estruturada: decisão, laços, desvio de fluxo, atribuição, etc.

Métodos: **é onde as coisas acontecem na POO!**

Interfaces de Objetos



Segunda parte.

- ~~Visão geral de conceitos de POO~~
- ~~Abstração~~
- ~~Objetos~~
- ~~Ciclo de vida dos objetos (instanciação à destruição)~~
- ~~Classes~~
- ~~Membros de uma classe: atributos e métodos~~
- ~~Interface de objetos~~
- Construtores e suas características
- ~~Encapsulamento~~
- Varáveis e métodos de classe
- Herança e Polimorfismo
- Sobrecarga e sobreposição de métodos
- Documentando programas Java

Parabéns agora você já sabe algo de Java.

Resolva a lista e exercício no Ambiente Virtual.

Material Antigo, serve para os que faltam as primeiras aulas também.