

Introdução à Orientação a Objetos

SCC0604 - Programação Orientada a Objetos

Prof. Fernando V. Paulovich

<http://www.icmc.usp.br/~paulovic>

paulovic@icmc.usp.br

Instituto de Ciências Matemáticas e de Computação (ICMC)
Universidade de São Paulo (USP)

2 de agosto de 2010



Sumário

- 1 Programação Estruturada (PE)
- 2 Programação Orientada a Objetos (POO)
- 3 Mecanismos Básicos da Orientação a Objetos
- 4 Princípios da Orientação a Objetos
- 5 Conceitos Básicos da Linguagem Java

Sumário

- 1 Programação Estruturada (PE)
- 2 Programação Orientada a Objetos (POO)
- 3 Mecanismos Básicos da Orientação a Objetos
- 4 Princípios da Orientação a Objetos
- 5 Conceitos Básicos da Linguagem Java

Introdução

- Normalmente os comandos de um programa são **executados sequencialmente**
- Em C (e C++) vários comandos permitem que essa sequência seja quebrada, causando o que é chamado de **transferência de controle**
- Durante a década de 60 tornou-se claro que o uso indiscriminado de transferência de controle era a raiz de muitos problemas
 - Comando **goto** foi considerado o grande culpado

Solução: Programação Estruturada (PE)

- Dijkstra (1968) provou que independente da complexidade envolvida, todo programa poderia ser escrito como uma combinação de comandos primitivos envolvendo três estruturas básicas de controle
 - 1 *estrutura de sequência*
 - 2 *estrutura de seleção*
 - 3 *estrutura de repetição*
 - Assim nascia a **Programação Estruturada**
-
- A experiência tem mostrado que a melhor forma de se desenvolver programas de grande porte é construí-los a partir de pequenas partes (**dividir para conquistar**)
 - Em programas estruturados essas partes são denominadas **funções**

Sumário

- 1 Programação Estruturada (PE)
- 2 Programação Orientada a Objetos (POO)
- 3 Mecanismos Básicos da Orientação a Objetos
- 4 Princípios da Orientação a Objetos
- 5 Conceitos Básicos da Linguagem Java

Importância

- Devido os requisitos atuais, os softwares têm se tornado cada vez mais **complexos** e **maiores**
 - Isso tem levado a busca de meios para tornar a tarefa de **programação mais produtiva**
- Ainda não existe uma resposta definitiva a essa busca, mas há um consenso de que a **Programação Orientada a Objetos (POO)** consegue produzir **resultados mais competitivos** do que as abordagens atualmente empregadas

História

- A ideia da Orientação a Objetos **surgiu a partir das técnicas empregadas para o desenvolvimento de hardwares**, onde pedaços simples de hardware (chips) eram unidos para se montar um hardware mais complexo
- Historicamente, a Orientação a Objetos foi consequência do **amadurecimento de princípios já existentes** na década de 60 (linguagem Simula, 1960-67), reimplementados na década de 70 (linguagem Smalltalk) e incorporados a novas linguagens ou linguagens ampliadas na década de 80 (linguagem C++)

Enfoque Tradicional X Enfoque Orientado a Objetos

- Na Análise Estruturada (AE)
 - Procedimentos são implementados em blocos e a comunicação entre eles se dá pela **passagem de dados**
 - Um programa estruturado, quando em execução, é caracterizado pelo **acionamento de procedimentos** cuja tarefa é a manipulação de dados
- Na Análise Orientada a Objeto (AOO)
 - **Dados e procedimentos** são encapsulados em um só elemento denominado **objeto**
 - O estabelecimento de **comunicação entre objetos** (envio e recebimento de mensagens) caracteriza a execução do **programa**

Vantagens da AOO em Relação à AE

- Como **vantagens da AOO em relação à Análise Estruturada** temos
 - maior índice de reaproveitamento de código

Vantagens da AOO em Relação à AE

- Como **vantagens da AOO em relação à Análise Estruturada** temos
 - maior índice de reaproveitamento de código
 - maior facilidade de manutenção

Vantagens da AOO em Relação à AE

- Como **vantagens da AOO em relação à Análise Estruturada** temos
 - maior índice de reaproveitamento de código
 - maior facilidade de manutenção
 - menor código gerado

Vantagens da AOO em Relação à AE

- Como **vantagens da AOO em relação à Análise Estruturada** temos
 - maior índice de reaproveitamento de código
 - maior facilidade de manutenção
 - menor código gerado
 - maior confiabilidade no código

Vantagens da AOO em Relação à AE

- Como **vantagens da AOO em relação à Análise Estruturada** temos
 - maior índice de reaproveitamento de código
 - maior facilidade de manutenção
 - menor código gerado
 - maior confiabilidade no código
 - maior facilidade de gerenciamento do código (reduz grandes problemas para menores);

Vantagens da AOO em Relação à AE

- Como **vantagens da AOO em relação à Análise Estruturada** temos
 - maior índice de reaproveitamento de código
 - maior facilidade de manutenção
 - menor código gerado
 - maior confiabilidade no código
 - maior facilidade de gerenciamento do código (reduz grandes problemas para menores);
 - maior robustez, etc.

O que Isso Significa?

Alguns Dados Importantes (*Gartner Institute*)

- 74% de todos os projetos de TI **falham**, custam **mais caro** no final ou **não obedecem o prazo final**

O que Isso Significa?

Alguns Dados Importantes (*Gartner Institute*)

- 74% de todos os projetos de TI **falham**, custam **mais caro** no final ou **não obedecem o prazo final**
- 28% falham em **todos** esses requisitos

O que Isso Significa?

Alguns Dados Importantes (*Gartner Institute*)

- 74% de todos os projetos de TI **falham**, custam **mais caro** no final ou **não obedecem o prazo final**
- 28% falham em **todos** esses requisitos
- 52,7% custam 189% do preço original

O que Isso Significa?

Alguns Dados Importantes (*Gartner Institute*)

- 74% de todos os projetos de TI **falham**, custam **mais caro** no final ou **não obedecem o prazo final**
- 28% falham em **todos** esses requisitos
- 52,7% custam 189% do preço original
- Todo ano, US\$75B são gastos com **projetos que falharam** (35% do PIB Argentino em 2007)

Sumário

- 1 Programação Estruturada (PE)
- 2 Programação Orientada a Objetos (POO)
- 3 Mecanismos Básicos da Orientação a Objetos**
- 4 Princípios da Orientação a Objetos
- 5 Conceitos Básicos da Linguagem Java

Objeto

- Um **objeto** é uma entidade que formaliza o modo pelo qual **compreendemos algo no domínio do problema**
 - Reflete a capacidade do sistema de **guardar informações** sobre o elemento abstraído, **interagir** com ele, ou ambas as coisas

Objeto

- Um **objeto** é uma entidade que formaliza o modo pelo qual **compreendemos algo no domínio do problema**
 - Reflete a capacidade do sistema de **guardar informações** sobre o elemento abstraído, **interagir** com ele, ou ambas as coisas
 - Entidade o **mais próximo possível das entidades do mundo real** – aquilo que é tangível ou visível

Objeto

- Um **objeto** é uma entidade que formaliza o modo pelo qual **compreendemos algo no domínio do problema**
 - Reflete a capacidade do sistema de **guardar informações** sobre o elemento abstraído, **interagir** com ele, ou ambas as coisas
 - Entidade o **mais próximo possível das entidades do mundo real** – aquilo que é tangível ou visível
 - Dessa forma, os **objetos são os substantivos** do domínio do problema

Objeto

- A um objeto sempre estarão associados
 - O **estado** do objeto é definido pelas propriedades que ele possui e pelos valores que elas estão assumindo

Objeto

- A um objeto sempre estarão associados
 - O **estado** do objeto é definido pelas propriedades que ele possui e pelos valores que elas estão assumindo
 - O **comportamento** do objeto é definido pela forma como ele age e reage, em termos de mudança de seu estado e o relacionamento com os demais objetos do sistema

Objeto

- A um objeto sempre estarão associados
 - O **estado** do objeto é definido pelas propriedades que ele possui e pelos valores que elas estão assumindo
 - O **comportamento** do objeto é definido pela forma como ele age e reage, em termos de mudança de seu estado e o relacionamento com os demais objetos do sistema
 - A **identidade** do objeto é a propriedade pela qual ele se distingue dos demais

Classe

- Uma **classe** descreve um **conjunto de objetos semelhantes**
 - atributos e métodos que resumem as características comuns de vários objetos
- A **diferença entre classe e objeto** está no fato de um objeto constituir uma entidade concreta com tempo e espaço de existência, enquanto a classe é tão-somente uma abstração
- Em termos de programação, definir uma **classe significa formalizar um tipo de dado** e todas as operações associadas a esse tipo, enquanto declarar **objetos significa criar variáveis do tipo definido**

Classe (Membros)

- Uma classe possui **dados** que definem suas propriedades e **procedimentos** que devem ser executados sobre esses dados (**membros da classe**)

Classe (Membros)

- Uma classe possui **dados** que definem suas propriedades e **procedimentos** que devem ser executados sobre esses dados (**membros da classe**)
- Os dados caracterizam **atributos** e os procedimentos **métodos** (ou serviços) da classe

Classe (Atributos)

- Na AOO uma das características de um objeto é seu **estado**. Um estado é representado pelos **valores de seus atributos**

Classe (Atributos)

- Na AOO uma das características de um objeto é seu **estado**. Um estado é representado pelos **valores de seus atributos**
- Os **atributos** devem ser **manipulados exclusivamente por serviços associados à classe** a que pertencem ou a classes a ela relacionadas

Classe (Atributos)

- Na AOO uma das características de um objeto é seu **estado**. Um estado é representado pelos **valores de seus atributos**
- Os **atributos** devem ser **manipulados exclusivamente por serviços associados à classe** a que pertencem ou a classes a ela relacionadas
- **Diferentes objetos** de uma mesma classe **não compartilham os mesmos atributos**, cada um possui sua própria cópia do atributo

Classe (Métodos ou Serviços)

- Os métodos são **operações que podem ser executadas pelos objetos**

Classe (Métodos ou Serviços)

- Os métodos são **operações que podem ser executadas pelos objetos**
- Serviço é um comportamento específico, residente no objeto, que define como ele deve agir quando exigido

Classe (Acesso a Membros)

- Uma classe pode definir o tipo de acesso que seus membros permitirão às demais partes do sistema

Classe (Acesso a Membros)

- Uma classe pode definir o tipo de acesso que seus membros permitirão às demais partes do sistema
 - **Público:** O atributo ou método da classe pode ser acessado por todas as demais entidades do sistema

Classe (Acesso a Membros)

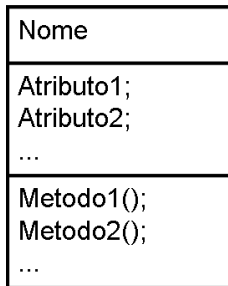
- Uma classe pode definir o tipo de acesso que seus membros permitirão às demais partes do sistema
 - **Público:** O atributo ou método da classe pode ser acessado por todas as demais entidades do sistema
 - **Protegido:** O atributo ou método da classe pode ser acessado somente por classes da mesma hierarquia

Classe (Acesso a Membros)

- Uma classe pode definir o tipo de acesso que seus membros permitirão às demais partes do sistema
 - **Público:** O atributo ou método da classe pode ser acessado por todas as demais entidades do sistema
 - **Protegido:** O atributo ou método da classe pode ser acessado somente por classes da mesma hierarquia
 - **Privado:** O atributo ou método da classe pode ser acessado somente por métodos da própria classe

Classe (Notação Gráfica)

- A notação gráfica de uma classe permite visualizar uma abstração independente de qualquer linguagem de implementação específica, dando ênfase às partes mais importantes de uma abstração: seu nome, atributos e métodos (operações)



Exercício

Modelagem da Faculdade Tabajara Inc.

A Faculdade Tabajara Inc., situada às margens da rodovia dos Imigrantes, é um centro de excelência que pode transformar seu sonho em realidade. Hoje contamos com quatro cursos, oferecendo uma infraestrutura que une ensino prático e teórico, respondendo as necessidades do mercado de trabalho.

Organizacionalmente, cada curso conta com seu próprio departamento, responsável pelo gerenciamento das disciplinas oferecidas, de forma a permitir que cada curso possa funcionar de forma independente, como uma unidade auto-gerenciável.

Nosso quadro docente conta com professores altamente capacitados alocados aos diversos departamentos, ministrando disciplinas conforme suas especialidades. Além disso, para tornar a administração departamental mais próxima aos anseios dos alunos, temos para cada departamento um professor responsável por sua administração.

Exercício

Modelagem Disk Pizza

Quando um cliente telefona para o disk-pizza, primeiro é verificado, através de seu telefone, se o mesmo já está cadastrado no sistema. Caso o cliente não conste no cadastro da empresa, o mesmo deve ser cadastrado por meio de seus dados pessoais: nome, endereço, telefone e CPF.

O disk-pizza oferece uma variedade de pizzas e bebidas, sendo que o cliente pode pedir de uma só vez, várias pizzas e bebidas por meio de um número de identificação da pizza ou bebida (números do cardápio), ou por meio de seus nomes. Vale salientar que, no momento do pedido, os clientes podem querer saber quais são os ingredientes que compõem uma determinada pizza, sendo possível a retirada de algum ingrediente caso o cliente assim desejar.

Além de poder pedir pizzas que constam no cardápio, um cliente pode requerer a “montagem” de uma pizza com certos ingredientes. Nesse caso, o preço da pizza será cobrado com base nos ingredientes que irão compor a mesma.

Por fim, as bebidas são fornecidas por diversas distribuidoras, sendo que o sistema deve verificar constantemente o estoque de forma a saber se novos pedidos devem ser feitos. Essas distribuidoras devem ser cadastradas no sistema por meio de um nome, pessoa para contato, CGC, inscrição estadual e municipal, telefone, etc

Exercício

Modelagem Biblioteca

A biblioteca da Faculdade Tabajara empresta aos alunos, funcionários, professores e outras pessoas cadastradas os seguintes tipos de obras: livros e fitas de vídeo. Para se cadastrar, um usuário necessita comprovar vínculo com a instituição, mesmo que seja um grau de parentesco (por exemplo: filhos de professores podem usar a biblioteca). Tais usuários são cadastrados através de seu RG, nome, endereço, telefone, entre outros atributos.

Após ser cadastrado na biblioteca, para usar os serviços, o usuário recebe um cartão magnético (contendo os seus dados de usuário). Caso ele deseje emprestar um livro que esteja disponível, isto é, não emprestado, deve usar o seu cartão magnético. No instante do empréstimo, o funcionário verifica se existe alguma pendência. Caso exista, o empréstimo não pode ser feito e pior, caso haja multa, sua carteira magnética só é devolvida após a quitação do débito. O empréstimo tanto de livros, como de fitas de vídeo se dá por meio de um período máximo de sete dias.

Além disso, caso o usuário queira fazer uma consulta às obras cadastradas na biblioteca, ele pode fazê-la através em um terminal de computador, interno à biblioteca, pesquisando pelo título de uma obra, autor de uma obra, etc.

Uma situação muito importante nessa biblioteca é o cadastramento das obras. Quando as mesmas chegam, elas não podem ser emprestadas até que os bibliotecários efetivem o seu tombamento. Os bibliotecários não sabem qual o intervalo de período que os livros chegam, pois os pedidos partem do setor de compras da faculdade, no entanto, quando as obras chegam, os bibliotecários são responsáveis por cadastrar os dados da chegada, tais como: nota fiscal, fornecedor, data de chegada e condição de entrega.

Sumário

- 1 Programação Estruturada (PE)
- 2 Programação Orientada a Objetos (POO)
- 3 Mecanismos Básicos da Orientação a Objetos
- 4 Princípios da Orientação a Objetos**
- 5 Conceitos Básicos da Linguagem Java

Princípios da Orientação a Objetos

- O paradigma orientado a objetos define alguns **princípios básicos** que devem ser seguidos

Princípios da Orientação a Objetos

- O paradigma orientado a objetos define alguns **princípios básicos** que devem ser seguidos
 - Abstração

Princípios da Orientação a Objetos

- O paradigma orientado a objetos define alguns **princípios básicos** que devem ser seguidos
 - Abstração
 - Encapsulamento

Princípios da Orientação a Objetos

- O paradigma orientado a objetos define alguns **princípios básicos** que devem ser seguidos
 - Abstração
 - Encapsulamento
 - Herança

Princípios da Orientação a Objetos

- O paradigma orientado a objetos define alguns **princípios básicos** que devem ser seguidos
 - Abstração
 - Encapsulamento
 - Herança
 - Polimorfismo

Princípios da Orientação a Objetos

- O paradigma orientado a objetos define alguns **princípios básicos** que devem ser seguidos
 - Abstração
 - Encapsulamento
 - Herança
 - Polimorfismo
 - Modularidade

Princípios da Orientação a Objetos

- O paradigma orientado a objetos define alguns **princípios básicos** que devem ser seguidos
 - Abstração
 - Encapsulamento
 - Herança
 - Polimorfismo
 - Modularidade
 - Mensagens

Abstração

- Consiste em **identificar artefatos** de software na modelagem de um domínio – ignorar aspectos não-relevantes, concentrando-se apenas nos assuntos principais do problema

Abstração

- Consiste em **identificar artefatos** de software na modelagem de um domínio – ignorar aspectos não-relevantes, concentrando-se apenas nos assuntos principais do problema
- **Classes** são abstrações de conceitos

Abstração

- Consiste em **identificar artefatos** de software na modelagem de um domínio – ignorar aspectos não-relevantes, concentrando-se apenas nos assuntos principais do problema
- **Classes** são abstrações de conceitos
- Consiste basicamente no processo de retirar do domínio do problema os **detalhes relevantes** e representá-los não mais na linguagem de domínio, e sim na linguagem de solução (ex.: Java)

Encapsulamento

- A propriedade de **implementar dados e procedimentos correlacionados em uma mesma entidade** recebe o nome de Encapsulamento

Encapsulamento

- A propriedade de **implementar dados e procedimentos correlacionados em uma mesma entidade** recebe o nome de Encapsulamento
- A ideia por trás do encapsulamento é a de que um sistema orientado a objetos **não deve depender de sua implementação interna**, e sim de sua interface

Herança

- Permite a **hierarquização** das classes em um sistema

Herança

- Permite a **hierarquização** das classes em um sistema
- Uma classe mais especializada (sub-classe ou classe-derivada) **herda as propriedades** (métodos e atributos) e semântica de uma classe mais geral (super-classe ou classe-base)

Herança

- Permite a **hierarquização** das classes em um sistema
- Uma classe mais especializada (sub-classe ou classe-derivada) **herda as propriedades** (métodos e atributos) e semântica de uma classe mais geral (super-classe ou classe-base)
- Uma sub-classe pode **sobrescrever o comportamento** de uma super-classe

Herança

- Permite a **hierarquização** das classes em um sistema
- Uma classe mais especializada (sub-classe ou classe-derivada) **herda as propriedades** (métodos e atributos) e semântica de uma classe mais geral (super-classe ou classe-base)
- Uma sub-classe pode **sobrescrever o comportamento** de uma super-classe
- É um relacionamento “**é-um(a)**”

Herança

- Permite a **hierarquização** das classes em um sistema
- Uma classe mais especializada (sub-classe ou classe-derivada) **herda as propriedades** (métodos e atributos) e semântica de uma classe mais geral (super-classe ou classe-base)
- Uma sub-classe pode **sobrescrever o comportamento** de uma super-classe
- É um relacionamento “**é-um(a)**”
- Promove **re-uso**

Polimorfismo

- Possibilita que **métodos** diferentes, implementados em diferentes níveis da hierarquia de classes, tenham o mesmo nome (**homônimos**)

Polimorfismo

- Possibilita que **métodos** diferentes, implementados em diferentes níveis da hierarquia de classes, tenham o mesmo nome (**homônimos**)
- Possibilita que um método possa apresentar **diferentes comportamentos**

Modularidade

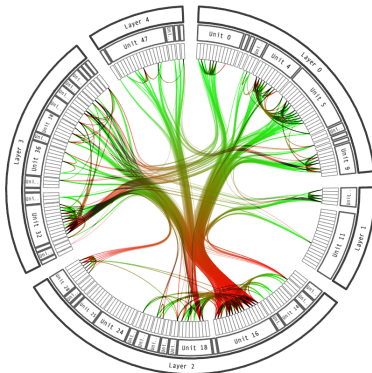
- Uma das capacidades de um sistema orientado a objetos é a separação de **conjunto de módulos**, cada um contendo classes com **independência de funcionamento**

Modularidade

- Uma das capacidades de um sistema orientado a objetos é a separação de **conjunto de módulos**, cada um contendo classes com **independência de funcionamento**

Modularidade

- Uma das capacidades de um sistema orientado a objetos é a separação de **conjunto de módulos**, cada um contendo classes com **independência de funcionamento**



Mensagens

- Objetos se comunicam por meio do **envio/recebimento de mensagens**

Mensagens

- Objetos se comunicam por meio do **envio/recebimento de mensagens**
- Uma mensagem contém alguma forma de **informação**

Sumário

- 1 Programação Estruturada (PE)
- 2 Programação Orientada a Objetos (POO)
- 3 Mecanismos Básicos da Orientação a Objetos
- 4 Princípios da Orientação a Objetos
- 5 Conceitos Básicos da Linguagem Java

Introdução a Linguagem Java

Histórico

- Na invenção de Java, a *National Public Radio* apresentou uma história de 10 minutos sobre a linguagem Java

Introdução a Linguagem Java

Histórico

- Na invenção de Java, a *National Public Radio* apresentou uma história de 10 minutos sobre a linguagem Java
- Os criadores de Java apontam a mesma para a criação de **novos aplicativos** (para web, celulares, PDAs, etc.) e **não para a reescrita do que já existe**

Introdução a Linguagem Java

Histórico

- Na invenção de Java, a *National Public Radio* apresentou uma história de 10 minutos sobre a linguagem Java
- Os criadores de Java apontam a mesma para a criação de **novos aplicativos** (para web, celulares, PDAs, etc.) e **não para a reescrita do que já existe**

Introdução a Linguagem Java

Histórico

- Na invenção de Java, a *National Public Radio* apresentou uma história de 10 minutos sobre a linguagem Java
- Os criadores de Java apontam a mesma para a criação de **novos aplicativos** (para web, celulares, PDAs, etc.) e **não para a reescrita do que já existe**

Vantagens da Linguagem Java

- **Independência de plataforma** - necessário para aplicações Web

Introdução a Linguagem Java

Histórico

- Na invenção de Java, a *National Public Radio* apresentou uma história de 10 minutos sobre a linguagem Java
- Os criadores de Java apontam a mesma para a criação de **novos aplicativos** (para web, celulares, PDAs, etc.) e **não para a reescrita do que já existe**

Vantagens da Linguagem Java

- **Independência de plataforma** - necessário para aplicações Web
- **Totalmente orientada a objetos**, diferente do C++ - é muito **mais fácil** escrever código Java do que C++

Conceitos Errados sobre Java

Conceitos errados sobre Java

- Java é uma extensão do HTML

Conceitos Errados sobre Java

Conceitos errados sobre Java

- Java é uma extensão do HTML
- Java é fácil de aprender

Conceitos Errados sobre Java

Conceitos errados sobre Java

- Java é uma extensão do HTML
- Java é fácil de aprender
- Java vai se tornar uma linguagem universal

Conceitos Errados sobre Java

Conceitos errados sobre Java

- Java é uma extensão do HTML
- Java é fácil de aprender
- Java vai se tornar uma linguagem universal
- Java é apenas outra linguagem de programação

Conceitos Errados sobre Java

Conceitos errados sobre Java

- Java é uma extensão do HTML
- Java é fácil de aprender
- Java vai se tornar uma linguagem universal
- Java é apenas outra linguagem de programação
- Java é interpretada, portanto lenta

Conceitos Errados sobre Java

Conceitos errados sobre Java

- Java é uma extensão do HTML
- Java é fácil de aprender
- Java vai se tornar uma linguagem universal
- Java é apenas outra linguagem de programação
- Java é interpretada, portanto lenta
- As *applets* Java são um risco a segurança

Conceitos Errados sobre Java

Conceitos errados sobre Java

- Java é uma extensão do HTML
- Java é fácil de aprender
- Java vai se tornar uma linguagem universal
- Java é apenas outra linguagem de programação
- Java é interpretada, portanto lenta
- As *applets* Java são um risco a segurança
- O *JavaScript* é uma versão mais simples da linguagem Java

Compilando um Programa

Compilar

- Antes de executar um programa, o mesmo necessita ser compilado
- Para isso o comando *javac* deve ser usando, passando como argumento a classe a ser compilada

Compilando um Programa

Compilar

- Antes de executar um programa, o mesmo necessita ser compilado
- Para isso o comando *javac* deve ser usando, passando como argumento a classe a ser compilada

```
javac Principal.java
```

Executando um Programa

Executar

- Em Java, para se executar um programa (depois de compilá-lo), o comando *java* é usado, passando como argumento a classe que contém o método *main* a ser executado

Executando um Programa

Executar

- Em Java, para se executar um programa (depois de compilá-lo), o comando *java* é usado, passando como argumento a classe que contém o método *main* a ser executado

Executando um Programa

Executar

- Em Java, para se executar um programa (depois de compilá-lo), o comando *java* é usado, passando como argumento a classe que contém o método *main* a ser executado

```
java Principal
```

CLASSPATH

C:/bibliotecas/locadora/modelo

C:/bibliotecas/locadora/controle

C:/bibliotecas/locadora/visao

CLASSPATH=C:/bibliotecas/locadora;

import modelo.*; //para package modelo;

import controle.*; //para package controle;

import visao.*; //para package visao;