Capítulo 2

Entendendo as definições de classe

Principais conceitos a serem abrangidos

- campos
- construtores
- métodos
- parâmetros
- instruções de atribuição

Máquinas de vender bilhetes – uma visão externa

- Explorando o comportamento de uma máquina simples de vender bilhetes:
  - Utilize o projeto de *máquina de vender bilhetes*.
  - As máquinas fornecem bilhetes a um preço fixo.
    - Como é que o preço é determinado?
  - Como o 'dinheiro' é inserido em uma máquina?
  - Como uma máquina controla o dinheiro que é inserido?

Máquinas de vender bilhetes

Demo

Máquinas de vender bilhetes – uma visão externa

- Interagir com um objeto nos dá pistas sobre o seu comportamento.
- Analisar as partes internas nos permite determinar como esse comportamento é fornecido ou implementado.
- Todas as classes Java têm uma visualização interna semelhante.

Estrutura de classe básica

```
public class TicketMachine
{
    A parte interna da classe omitida.
}
```

O invólucro externo de TicketMachine

```
classe pública NomeDaClasse
{
    Campos
    Construtores
    Métodos
}
```

O conteúdo de uma classe

Campos

- Os campos armazenam dados para um objeto.
- Os campos são também conhecidos como variáveis de instância.
- Utilize a opção *Inspect* para visualizar os campos de um objeto.
- Os campos definem o estado de um objeto.

```
public class TicketMachine
{
    private int price;
    private int balance;
    private int total;

    Detalhes adicionais são
    omitidos.
}
```

modificador de visibilidade

tipo

nome variável

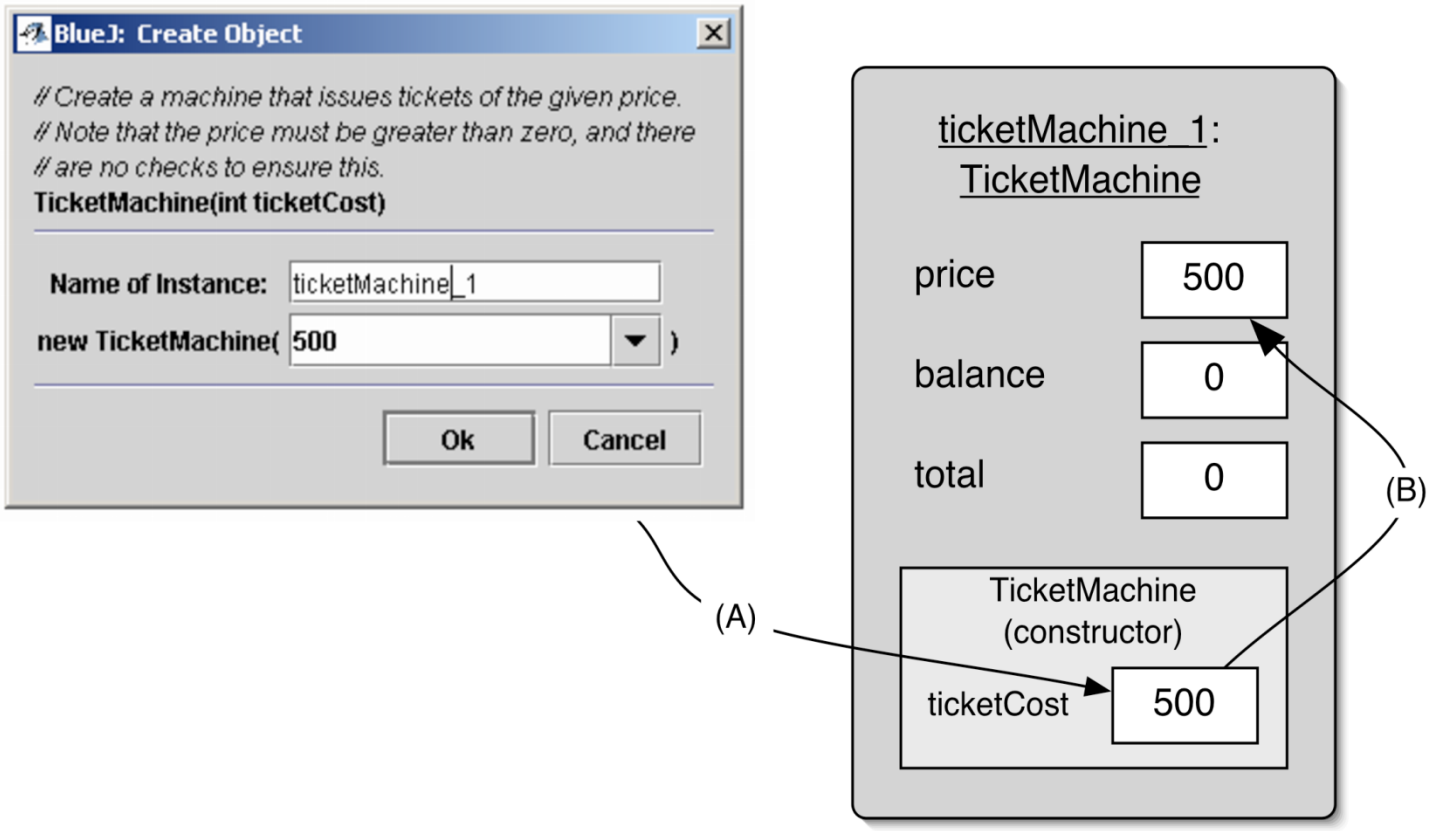
```
private int price;
```

Construtores

- Construtores inicializam um objeto.
- Têm o mesmo nome de sua classe.
- Armazenam valores iniciais para os campos.
- Costumam receber valores de parâmetro externo para isso.

```
public TicketMachine(int ticketCost)
{
    price = ticketCost;
    balance = 0;
    total = 0;
}
```

Passando dados via parâmetros



Atribuição

- Os valores são armazenados em campos (e outras variáveis), por meio de instruções de atribuição:
  - *variável = expressão;*
  - `price = ticketCost;`
- Uma variável armazena um valor único, por isso qualquer valor anterior é perdido.

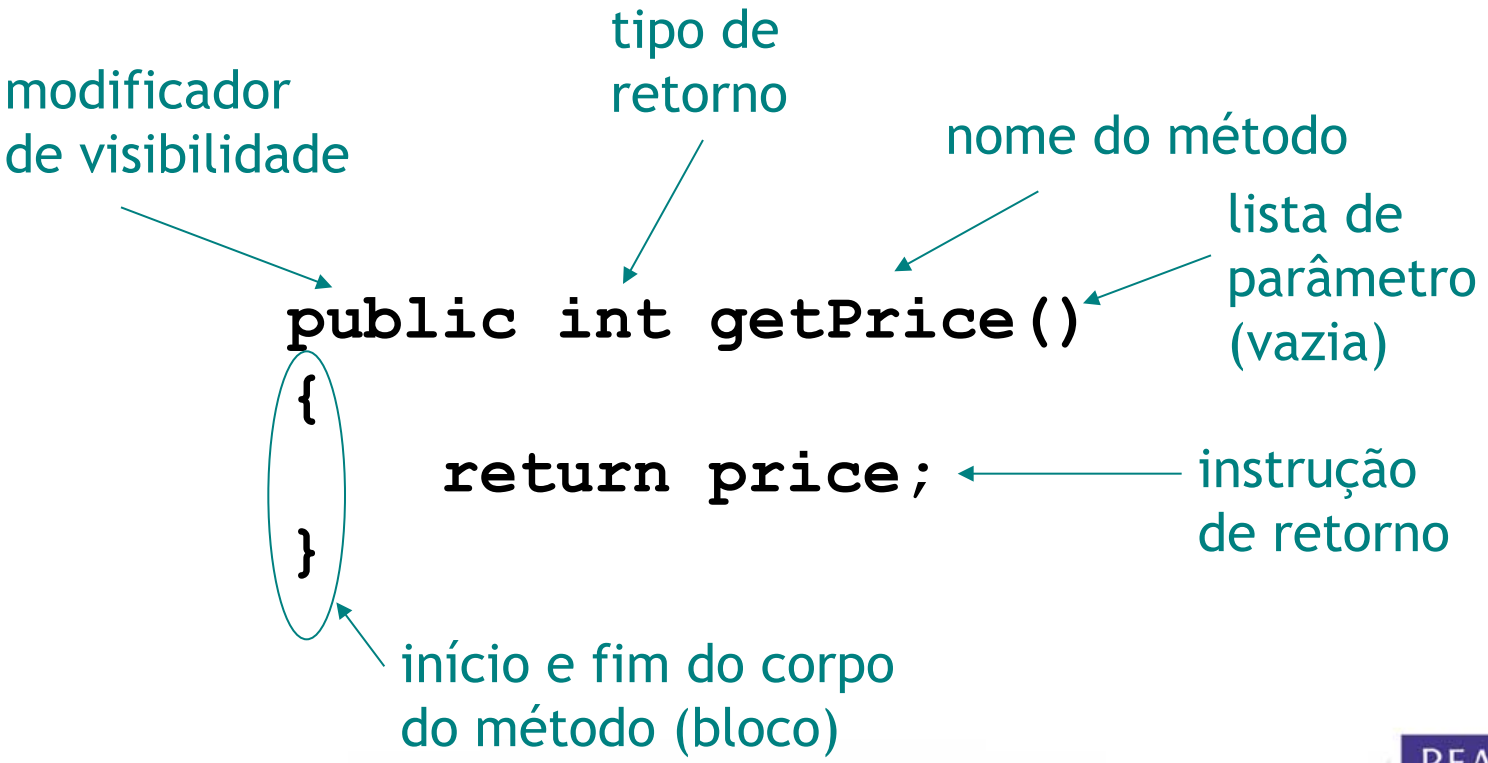
Principais conceitos a serem abrangidos

- métodos modificadores e métodos de acesso
- instruções condicionais
- variáveis locais
- concatenação de string

Métodos de acesso

- Os métodos implementam o comportamento de objetos.
- Os métodos de acesso fornecem informações sobre um objeto.
- Os métodos têm uma estrutura constituída por um cabeçalho e um corpo.
- O cabeçalho define o método da *assinatura*:
  `public int getPrice()`
- O corpo inclui instruções do método.

Métodos de acesso



Teste

```
public class CokeMachine
{
private price;

public CokeMachine()
{
   price = 300
}

public int getPrice
{
   return Price;
}
```

- O que está errado aqui?

(há cinco erros!)

Teste

```
public class CokeMachine
{
    in
private price;

public CokeMachine()
{
    price = 300
}

public int getPrice
{
    return Price;
}
}
```

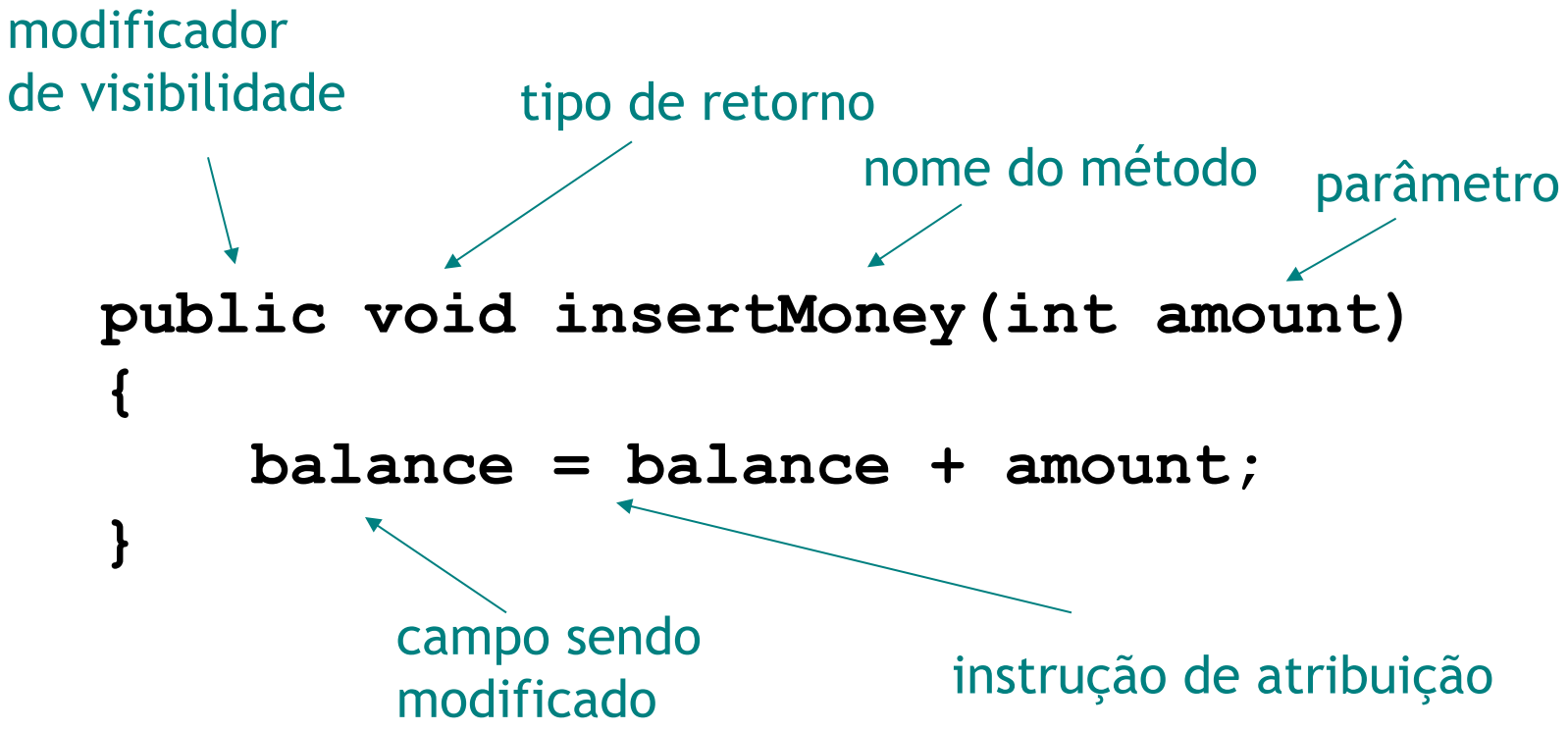
- O que está errado aqui?

(há cinco erros!)

Métodos modificadores

- Apresentam uma estrutura de método semelhante: cabeçalho e corpo.
- São usados para *modificar* o estado de um objeto.
- São alcançados através da mudança de valor de um ou mais campos.
  - Em geral, contêm instruções de atribuição.
  - Normalmente, recebem parâmetros.

Métodos modificadores



Imprimindo a partir de métodos

```
public void printTicket()
{
    // Simule a impressão de um bilhete.
    System.out.println("###################");
    System.out.println("# The BlueJ Line");
    System.out.println("# Ticket");
    System.out.println("# " + price + " cents.");
    System.out.println("###################");
    System.out.println();

    // Atualize o total coletado com o saldo.
    total = total + balance;
    // Limpa o saldo (balance).
    balance = 0;
}
```

Concatenação de string

- 4 + 5
  9
- "wind" + "ow"
  "window"
- "Resultado: " + 6
  "Resultado: 6"
- "# " + price + " cents"
  "# 500 cents"

Questionário

- System.out.println(5 + 6 + "hello");

```
11hello
```

- System.out.println("hello" + 5 + 6);

```
hello56
```

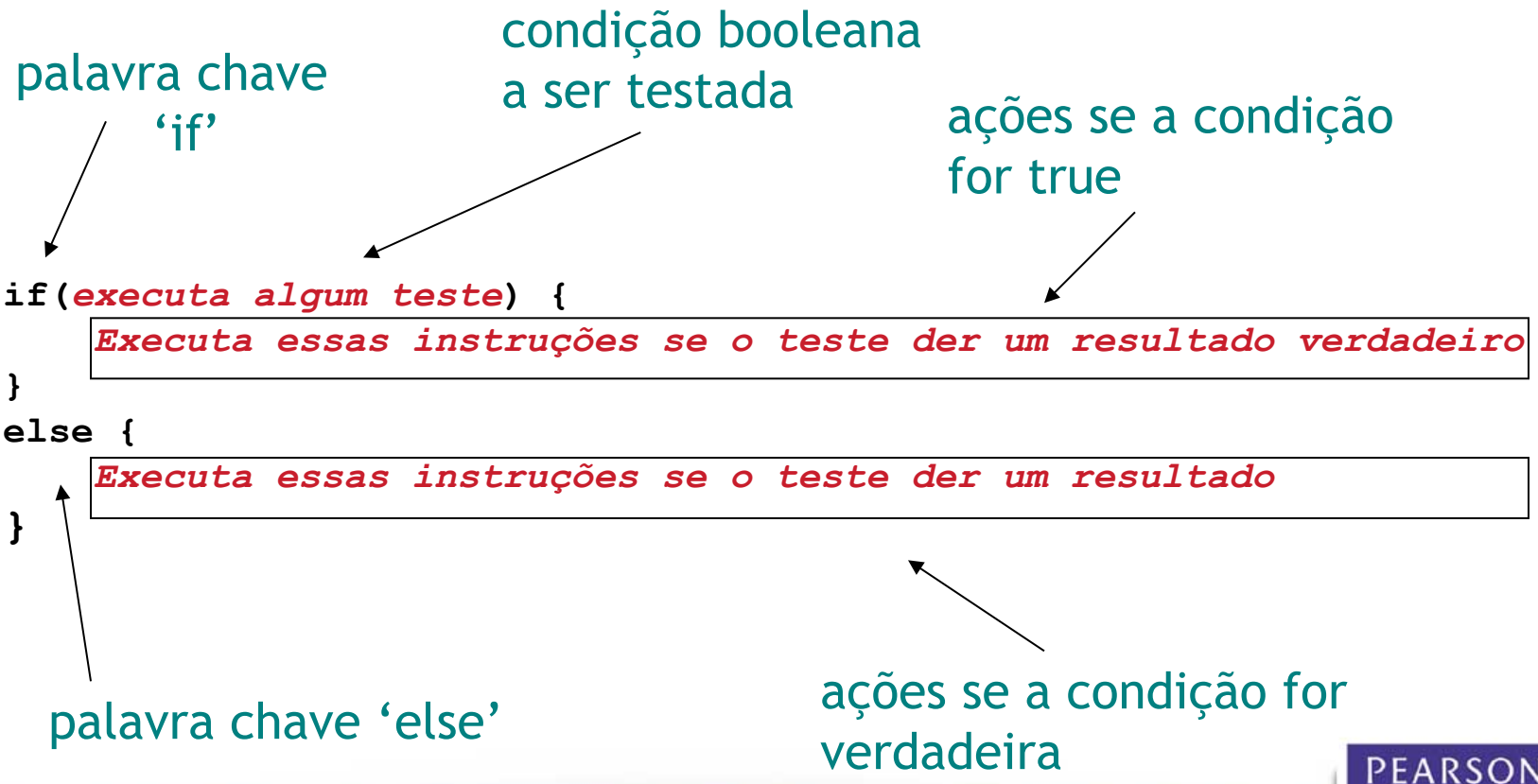
Refletindo sobre as máquinas de bilhetes

- Sob vários aspectos, seu comportamento é inadequado:
  - Sem verificação sobre as quantias inseridas.
  - Sem restituições.
  - Sem verificação para uma inicialização correta.
- Como podemos melhorar?
  - Precisamos de um comportamento mais sofisticado.

Fazendo escolhas:

```
public void insertMoney(int amount)
{
    if(amount > 0) {
        balance = balance + amount;
    }
    else {
        System.out.println("Use a positive amount: " +
                           amount);
    }
}
```

Fazendo escolhas:



Como escrever 'refundBalance'?

Variáveis locais

- Os campos são um tipo de variável:
  - Armazenam ao longo da vida de um objeto.
  - São acessíveis por toda a classe.

- Os métodos podem incluir variáveis de curta duração:
  - Existem apenas enquanto o método estiver em execução.
  - São acessíveis apenas de dentro do método.

Escopo e durabilidade

- O escopo de uma variável local é o bloco em que ela está declarada.
- O tempo de vida de uma variável local é o tempo de execução do bloco em que ela está declarada.

Variáveis locais

Uma variável local

```
public int refundBalance()
{
    int amountToRefund;
    amountToRefund = balance;
    balance = 0;
    return amountToRefund;
}
```

Revisão

- O corpo de classes contém campos, construtores e métodos.
- Os campos armazenam valores que determinam o estado de um objeto.
- Construtores inicializam objetos.
- Os métodos implementam o comportamento de objetos.

Revisão

- Os campos, os parâmetros e as variáveis locais são variáveis.
- Campos persistem durante toda a vida útil de um objeto.
- Os parâmetros são usados para receber valores em um construtor ou método.
- Variáveis locais são utilizadas para o armazenamento temporário de curta duração.

Revisão

- Os objetos podem tomar decisões por meio de instruções (if) condicionais.
- Um teste verdadeiro ou falso permite que um de dois cursos alternativos de ações seja tomado.