

INE5603 Introdução à POO

Prof. A. G. Silva

09 de agosto de 2017

Tópicos

- Algoritmo
- Classe e objeto
- Mensagem
- Identificadores em Java
- Tipos e valores
- Objetos e variáveis

Conceitos básicos de programação imperativa: algoritmos e programas

- **Algoritmo:**

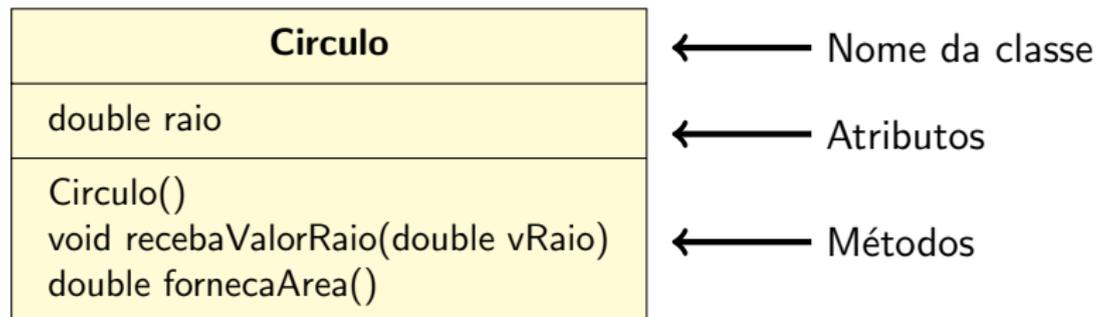
- ▶ Conjunto de passos discretos especificando como realizar uma tarefa
- ▶ Descrito de forma finita e não-ambígua
- ▶ Executável em tempo finito
- ▶ Usa recursos “tão grandes quanto necessários”

- **Programa:**

- ▶ Conjunto estruturado de instruções
- ▶ Capacitam uma máquina a aplicar sucessivamente
 - ★ Operações básicas
 - ★ Testes
- ▶ Possuem uma estrutura de controle

Classe – representação

- Exemplo: classe `Circulo`



Classe – código-fonte

```
public class Circulo {
    protected double raio;

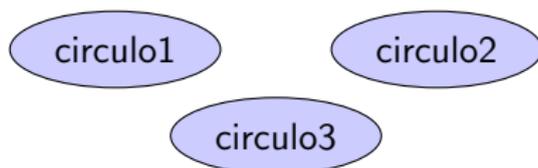
    public Circulo() {
        raio = 0.0;
    }

    public void recebaValorRaio(double vRaio) {
        raio = vRaio;
    }

    public double forneceArea() {
        double area;
        area = 3.1415 * raio * raio;
        return area;
    }
}
```

Objeto

- Exemplo: representação de objetos do tipo `Circulo`



- Exemplo: código-fonte para criação (instanciação) de objetos do tipo `Circulo`

```
public class Programa1 {  
    public static void main(String[] args) {  
        Circulo circulo1, circulo2, circulo3;  
        circulo1 = new Circulo();  
        circulo2 = new Circulo();  
        circulo3 = new Circulo();  
    }  
}
```

Mensagem

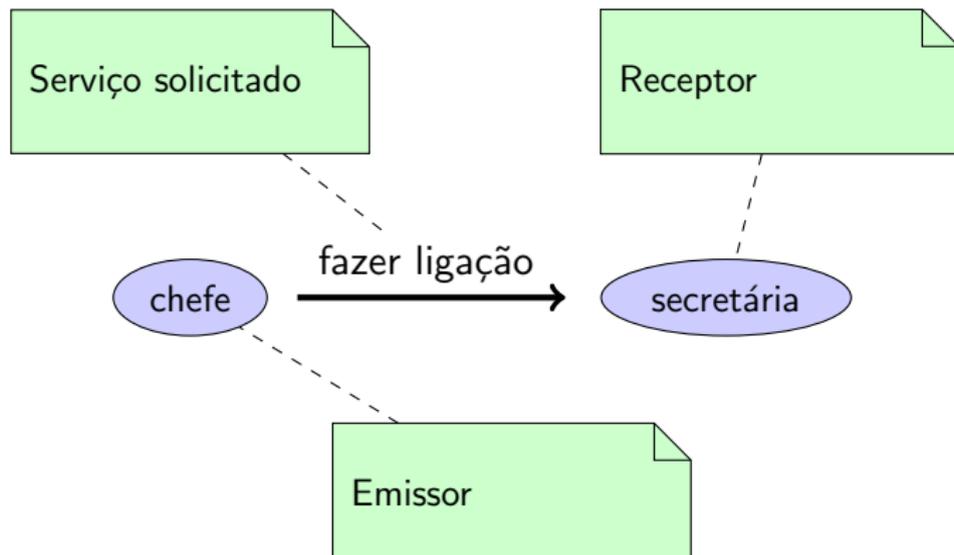
- Um objeto só executará um serviço se receber uma solicitação
- Essa solicitação é, muitas vezes, encaminhada por outro objeto
- Quando algum objeto necessitar que um outro objeto faça algum serviço, o primeiro deve se comunicar com o segundo, por meio de uma mensagem
- Objetos comunicam-se por meio do envio de mensagens, envolvendo três elementos:
 - ▶ Objeto emissor – encaminha a mensagem (sua classe é cliente)
 - ▶ Objeto receptor – recebe a mensagem e a especificação do serviço a ser executado (objeto executor)

Mensagem

- No envio, especifica-se o nome do objeto executor e qual o método a ser executado (também denominado “seletor”)
- Na especificação do método, devem ser especificados os recursos necessários à execução
- O objeto executor eventualmente tem de devolver alguma informação resultante

Mensagem

- Comunicação por envio de mensagem:



Exemplos

Exemplo 1

- O objeto umProfessor solicita ao objeto umAluno que estude
- A classe Aluno deve instanciar um objeto umAluno e este executar o serviço definido pelo método estude():

```
umAluno.estude();
```

Exemplo 2

- O objeto umaPessoa receba uma mensagem solicitando que modifique o valor de seu atributo idade:

```
umaPessoa.recebaValorIdade(25);
```

Exemplo 3

- Um hotel (umHotel) determinando a conta de um cliente (umCliente):
valorConta = umHotel.fornecaValorConta(umCliente);

Identificadores em Java

- Cada objeto ou classe deve ter uma identificação
- Atributos e métodos também devem ter uma identificação
- É interessante que identificadores sejam significativos (tenha relação com o que se identifica)

Identificadores em Java

- Em Java, as seguintes regras devem ser observadas na formação de um identificador:
 - ▶ O identificador pode ser formado por letras, dígitos, cifrão (\$) e sublinhado (-)
 - ▶ O primeiro caracter não pode ser um dígito
 - ▶ *Case-sensitive*
 - ▶ Não há limitação quanto ao número de caracteres
 - ▶ Não pode ser uma palavra-chave (*public*, *protect*, *return*, *int*, etc)

Tipos de valores

- Atributos de um objeto podem ser outros objetos ou valores (variáveis)
- A classe Circulo especificou o atributo raio como um valor que, por sua vez, foi definido (declarado) como do tipo double

Tipos de valores

- Em Java, os valores podem ser classificados nos seguintes tipos básicos:
 - ▶ Tipo `int`: números inteiros. Exemplos: 12, 2007, -9, +17, 0
 - ▶ Tipo `double`: números com uma parte inteira e uma parte fracionária, incluindo notação de potências de 10. Exemplos: 4.003, -0.37, 0.0, 4., -429.7, .0235, +2.0, -.5 5e-03, , -73.093E4. Também pode ser representado com um sufixo *D* ou *d*: 0.34D, 7.23d
 - ▶ Tipo `char`: expressam um único caractere. Representados entre apóstrofos. Exemplos: 'A', 's', '^', 'a', '*', '2', ' '
 - ▶ Tipo `boolean`: expressam apenas dois valores, true e false. Usado em condições ou expressões lógicas. Exemplo: `25 > 7` representa true

Objetos e variáveis

- Para que seja possível o envio da mensagem a umaCaneta.escreva(), na classe a qual pertence o objeto umaCaneta, deve-se definir o método escreva
- Um objeto só pode executar um serviço se o referido método existir em sua classe e esse objeto existir (tiver sido construído)
- Um objeto só passa a existir se for explicitamente construído. Ao ser construído, ocupa espaço de memória

Objetos e variáveis

- Diferentemente de objeto, uma variável constitui-se em uma posição de memória que armazena um determinado valor por vez (pode ser modificado durante a execução do programa)
- Uma variável pode ser declarada e usada isoladamente, constituindo-se em um elemento estático, e sempre com um tipo associado (*double*, *int*, etc)
- Um atributo representado por um valor é, de fato, uma variável

Primeiros programas em Java

- Programa em Java que calcule a soma de dois valores:

```
public class Programa2
{
    public static void main(String [] args)
    {
        int valorA, valorB, soma;

        valorA = 10; //armazena 10 na variavel valorA
        valorB = 15; //armazena 15 na variavel valorB

        soma = valorA + valorB;

        System.out.println(soma);
    }
}
```