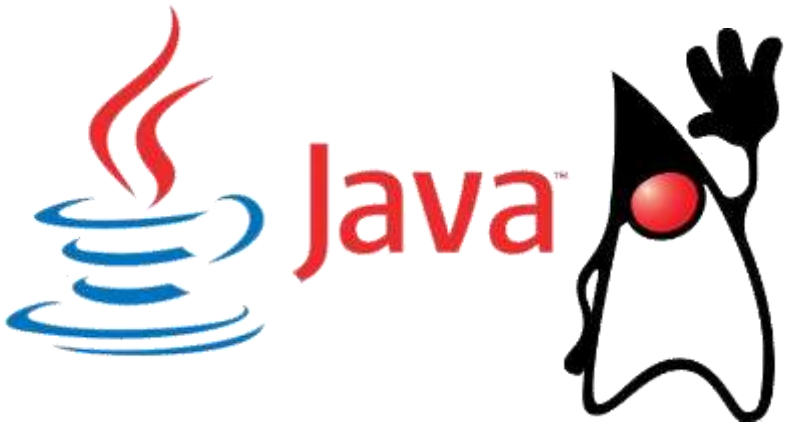


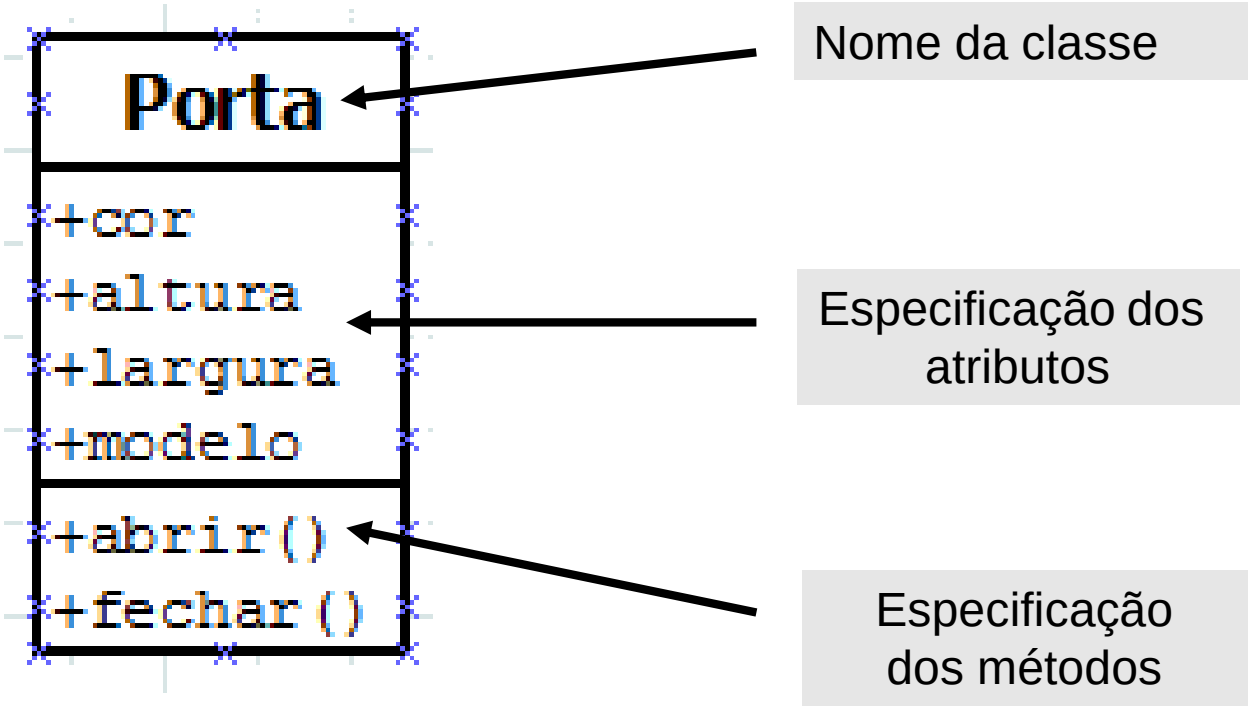
Programação Orientada a Objetos

Prof. Walter Gima

Objetivos

- ✓ Estruturas das Classes
- ✓ Atributos
- ✓ Métodos
- ✓ Visibilidade (modificadores de acesso)
  - ✓ Public
  - ✓ Private
- ✓ Encapsulamento

Classes: Representação de um objeto de mundo real.



Estruturas das Classes

Atributos: Características/qualidades de um objeto.

São declarados nas classes, sendo necessário informar a descrição do atributo e um tipo/referência.

Ex:

```
 class Porta {
        String cor;
        double altura;
        double largura;
}
```

Anhanguera

Tipo de Dados em Java x Referência

Tipo	Bits	MIN		MAX		Info	Valor Inicial
boolean	1	N/A	N/A	N/A	N/A	true/false	false
byte	8	=-2^7	-128	$2^{7} - 1$	127		(byte) 0
short	16	-2^15	-32768	$2^{15} - 1$	32767		(short) 0
char	16	0	0	$2^{15} - 1$	65535	'A', 'c', Unicode	\u0000
int	32	-2^31	-2.1E+09	$2^{31} - 1$	2.15E+09	Inteiros	0
float	32	2^{-149}		$(2-2^{-23}) \times 2^{127}$		Precisão simples / IEEE 754	0.0f
long	64	-2^63	-9.2E+18	$2^{63} - 1$	9.22E+18		0L
double	64	2^{-1074}		$(2-2^{-52}) \times 2^{1023}$		Precisão dupla / IEEE 574	0.0d

Referência: atributo utilizando como tipo uma classe criada no projeto.

Estruturas das Classes

Atributos: regras e informações para criar atributos classes Java:

- nome deve ser iniciados por letra, composto por única palavra (sem espaços, virgulas, etc).
- podem conter números na descrição.
- são iniciadas por letras minúsculas.
- não devem conter acentos e caracteres especiais.
- não podem ter descrição repetida.

Estruturas das Classes

Métodos: são as ações de um objeto.

São declarados nas classes, sendo necessário informar a tipo de retorno, descrição do método e se necessário os parâmetros.

Ex:

```
 class Porta {

        void abrir () {
                System.out.println(“Abrir porta”);
         }
}
```

Estruturas das Classes

Métodos: nomes dos métodos devem seguir as mesmas regras dos nomes de atributos.

- Não devem ser criados métodos dentro de outros métodos, me fora da classe a qual pertence.

- Caso um método não retornar nenhum valor deve ser declarado no lugar do tipo de retorno a palavra reservada void.

Ex:

```
class Porta {

        void abrir () {
                System.out.println(“Abrir porta”);
         }
}
```

Visibilidade ou Modificadores de Acesso: Public e Private

- **Private**: os atributos ou métodos declarados como private podem ser utilizados apenas dentro da implementação da própria classe.

- **Public**: os atributos ou métodos declarados como public podem ser acessados tanto dentro da própria classe ou por outra classe.

- **Protected**: pode ser acessado apenas pela própria classe ou pelas suas subclasses

Estruturas das Classes

Encapsulamento:

- Isolar implementação (funcionalidade) para deixar classe mais flexível e organizada.
- É bastante comum encapsular atribuição de valores de atributos através dos métodos get e set. Para cada atributo é criado um método **set** para informar valor de um atributo, também é criado um método **get** para obter o valor de um atributo.

Ex:

```
class ContaCorrente {
        private double saldo;

        void setSaldo (double valor) {
                this.saldo = valor;
        }
        double getSaldo() { return this.saldo; }
}
```

Exercícios

1) O que são atributos?

2) Quais as regras para se criar atributos?

3) Dê três exemplos de nomes corretos para atributos três exemplos de forma incorreta.

2) O que são métodos?

3) O que é encapsulamento de atributos? Dê 2 exemplos.

4) Descreva o objetivo dos modificadores de acesso?

5) Qual é a estrutura utilizada para se criar um método em uma classe ?

6) Quanto o método não retorna valores o que deve ser informado no método?

Atividade da Semana

Utilizando linguagem Java:

1) Criar a classe ConverteValores que converte valores em reais (R$) para dólar ($).

2) Criar a classe Temperatura que converte temperaturas Farenheit para Celsius.

3) Criar a classe CalculaIMC que recebe valores peso (Kg) e altura (mt) para calcular o índice de massa corporal.

4) Criar a classe Funcionario com os seguintes requisitos:
   - criar os atributos nome, fone, email (tipo String) e salario (double).
   - utilizar encapsulamento dos atributos.
   - criar método calcularDescontos que retorna o valor de descontos do funcionário (calcular 8% de desconto)
   - criar método getSalarioBruto, que retorna o salario sem descontos.
   - criar método getSalarioLiquido, que retorna salario subtraindo os descontos.
   - Instanciar pelo menos dois objetos funcionários com salários diferentes e exibir salario bruto, valor de desconto e valor do salario liquido.

Bibliografia

Livro Texto:

» 1) SANTOS, Rafael. Introdução à Programação Orientada a Objetos Usando Java. 2ª ed. Rio de Janeiro: Campus - Elsevier, 2013.

» 2) DEITEL, Harvery M.. Java: Como Programar. 6ª ed. São Paulo: Pearson - Prentice Hall, 2007.

» 3) SIERRA, Kathy; BATES, Bert. Use a Cabeça! Java. 1ª ed. Rio de Janeiro: Alta Books, 2005.

» 4) Serson,Roberto Rubinstein. Programação Orientada a Objetos com Java 6. Brasport, 2007.

Anhanguera

Dúvidas ?

walter.gima@anhanguera.com