

Programação em Java

Eduardo Figueiredo

<http://www.dcc.ufmg.br/~figueiredo>

Implementando um Projeto

- Assim como em outras engenharias, o projeto não é suficiente
 - Ninguém pode dirigir o desenho de um carro
 - Ninguém pode executar o projeto de um sistema
 - Por enquanto...

Do Projeto para Implementação

- Antes do carro ficar pronto, ele deve ser projetado

Carro
cor
velocidadeAtual
acelerar()
frear()

Projeto



```
class Carro {
    String cor;
    int velocidadeAtual;

    void acelerar() {}
    void frear() {}
}
```

Implementação

Classe Pública em Java

- Declaração de uma classe pública (**public**) deve ser feita em um arquivo de mesmo nome
 - Com a extensão ".java"
- Modificador **public** permite acesso / chamadas de qualquer outra classe do programa

```
public class Carro {
    String cor;
    int velocidadeAtual;

    void acelerar() {}
    void frear() {}
}
```

Arquivo Carro.java

Corpo de Classes e Métodos

- O corpo de uma classe é delimitado por chaves { }
- O corpo de um método também é delimitado por chaves { }

```
public class Carro {
    String cor;
    int velocidadeAtual;

    void acelerar() {...}
    void frear() {...}
}
```

Convenção de Nomes

- Nome de classes
 - Substantivo
 - Iniciar com letra maiúscula
 - CamelCase (MeuCarro)
- Nome de métodos
 - Verbo
 - Iniciar com minúsculo
 - camelCase (acelerarCarro)

```
public class Carro {
    String cor;
    int velocidadeAtual;

    void acelerar() {...}
    void frear() {...}
}
```

[Assinatura de Métodos]

- Modificador + Tipo de Retorno + Nome do Método + Lista de Parâmetros
 - **Modificador:** public | private | protected | ...
 - **Tipo de Retorno:** void | int | String | ...
 - **Nome:** acelerar | frear | ...
 - **Parâmetros:** delimitado por parênteses

```
public class Carro {  
    private String cor;  
    private int velocidadeAtual;  
  
    public void acelerar() {}  
    public void frear() {}  
}
```

[Implementando Métodos]

```
public class Carro {  
    String cor;  
    int velocidadeAtual = 0;  
  
    public void acelerar() {  
        velocidadeAtual = velocidadeAtual + 1;  
    }  
  
    public void frear() {  
        velocidadeAtual = velocidadeAtual - 1;  
    }  
}
```

[Executar a classe Carro ...]


```
public class Carro {  
    String cor;  
    int velocidadeAtual = 0;  
  
    public void acelerar() {  
        velocidadeAtual = velocidadeAtual + 1;  
    }  
  
    public void frear() {  
        velocidadeAtual = velocidadeAtual - 1;  
    }  
}
```

**Exception in
thread main ...**

Um programa Java começa
pelo método **main**

[O Método Main]

```
public class TesteCarro {  
  
    public static void main(String[] args) {  
        ...  
    }  
}
```

É uma boa prática colocar o método
main em uma classe separada 

[O Método Main]

```
public class TesteCarro {  
  
    public static void main(String[] args) {  
        ...  
    }  
}
```

O método **main** é automaticamente
chamado pela máquina virtual Java

[Implementando Métodos]

```
public class Carro {  
    String cor;  
    int velocidadeAtual = 0;  
  
    public void acelerar() {  
        velocidadeAtual = velocidadeAtual + 1;  
    }  
  
    public void frear() {  
        velocidadeAtual = velocidadeAtual - 1;  
    }  
  
    public void mostrarVelocidade() {  
        System.out.println("Minha velocidade é " +  
            velocidadeAtual);  
    }  
}
```

[Testar a Classe Carro]

```
public class TesteCarro {  
  
    public static void main(String[] args) {  
        Carro meuCarro = new Carro();  
        meuCarro.mostrarVelocidade();  
    }  
}
```

[Compilando o Programa]

```
public class TesteCarro {  
  
    public static void main(String[] args) {  
        Carro meuCarro = new Carro();  
        meuCarro.mostrarVelocidade();  
    }  
}
```

Compilar

```
javac Carro.java TesteCarro.java  
ou  
javac *.java
```

[Executando o Programa]

```
public class TesteCarro {  
  
    public static void main(String[] args) {  
        Carro meuCarro = new Carro();  
        meuCarro.mostrarVelocidade();  
    }  
}
```

Executar

```
java TesteCarro
```

[Resultado do Teste]

```
public class TesteCarro {  
  
    public static void main(String[] args) {  
        Carro meuCarro = new Carro();  
        meuCarro.mostrarVelocidade();  
    }  
}
```



```
Minha velocidade é 0
```

[Acelerando o Carro]

```
public class TesteCarro {  
  
    public static void main(String[] args) {  
        Carro meuCarro = new Carro();  
        meuCarro.mostrarVelocidade();  
        meuCarro.acelerar();  
        meuCarro.mostrarVelocidade();  
    }  
}
```

[Compilar e Executar]

```
public class Teste {  
  
    public static void main(String[] args) {  
        Carro meuCarro = new Carro();  
        meuCarro.mostrarVelocidade();  
        meuCarro.acelerar();  
        meuCarro.mostrarVelocidade();  
    }  
}
```

Compilar

Executar

[Resultado do Teste]

```
public class Teste {  
    public static void main(String[] args) {  
        Carro meuCarro = new Carro();  
        meuCarro.mostrarVelocidade();  
        meuCarro.acelerar();  
        meuCarro.mostrarVelocidade();  
    }  
}
```



Minha velocidade é 0
Minha velocidade é 1

Parâmetros e Instâncias

[Ainda a Classe Carro]

```
public class Carro {  
    String cor;  
    int velocidadeAtual = 0;  
  
    public void acelerar() {  
        velocidadeAtual = velocidadeAtual + 1;  
    }  
  
    public void frear() {  
        velocidadeAtual = velocidadeAtual - 1;  
    }  
  
    public void mostrarVelocidade() {  
        System.out.println("Minha velocidade é " +  
            velocidadeAtual);  
    }  
}
```

[Métodos para Cor]

```
public class Carro {  
    String cor;  
    int velocidadeAtual = 0;  
  
    public void acelerar() {...}  
    public void frear() {...}  
    public void mostrarVelocidade() {...}  
  
    public void definirCor(String novaCor) {  
        cor = novaCor;  
    }  
  
    public String retornarCor() {  
        return cor;  
    }  
}
```

[Métodos para Cor]

```
public class Carro {  
    String cor;  
    int velocidadeAtual = 0;  
  
    public void acelerar() {...}  
    public void frear() {...}  
    public void mostrarVelocidade() {...}  
  
    public void definirCor(String novaCor) {  
        cor = novaCor;  
    }  
  
    public String retornarCor() {  
        return cor;  
    }  
}
```

→ Método
"set"

→ Método
"get"

[Parâmetros de Métodos]

```
public class Carro {  
    String cor;  
    int velocidadeAtual = 0;  
  
    public void acelerar() {...}  
    public void frear() {...}  
    public void mostrarVelocidade() {...}  
  
    public void definirCor(String novaCor) {  
        cor = novaCor;  
    }  
  
    public String retornarCor() {  
        return cor;  
    }  
}
```

→ Método com um
parâmetro

→ Método sem
parâmetro

[Ocultando Atributos]

```
public class Carro {  
    private String cor;  
    private int velocidadeAtual = 0;  
  
    public void acelerar() {...}  
    public void frear() {...}  
    public void mostrarVelocidade() {...}  
  
    public void definirCor(String novaCor) {  
        cor = novaCor;  
    }  
  
    public String retornarCor() {  
        return cor;  
    }  
}
```

Atributos são geralmente privados (*private*) ou protegidos (*protected*)

[O método definirCor]

```
public void definirCor(String novaCor) {  
    cor = novaCor;  
}
```

- Não retorna nada quando é chamado
 - Tipo de retorno: void
- Recebe um parâmetro
 - Parâmetro: String novaCor

[O Método retornarCor]

```
public String retornarCor() {  
    return cor;  
}
```

- Retorna um objeto
 - Tipo de retorno: String
- Não recebe nenhum parâmetro
 - Parâmetro: vazio

[Testando definirCor]

```
public class TesteCarro {  
  
    public static void main(String[] args) {  
        Carro meuCarro = new Carro();  
        meuCarro.definirCor("Vermelho");  
    }  
}
```

[Passagem de Parâmetros]

```
public class TesteCarro {  
  
    public static void main(String[] args) {  
        Carro meuCarro = new Carro();  
        meuCarro.definirCor("Vermelho");  
    }  
}
```

Passando um parâmetro (String) para o método definirCor

[Criando Dois Carros]

```
public class TesteCarro {  
  
    public static void main(String[] args) {  
        Carro meuCarro = new Carro();  
        meuCarro.definirCor("Vermelho");  
  
        Carro seuCarro = new Carro();  
        seuCarro.definirCor("Azul");  
    }  
}
```

Instância "seuCarro" tem cor Azul

Instância "meuCarro" tem cor Vermelho

[Testando retornarCor]

```
public class TesteCarro {  
  
    public static void main(String[] args) {  
        Carro meuCarro = new Carro();  
        meuCarro.definirCor("Vermelho");  
  
        Carro seuCarro = new Carro();  
        seuCarro.definirCor("Azul");  
  
        System.out.println("Meu carro é " + meuCarro.retornarCor());  
        System.out.println("Seu carro é " + seuCarro.retornarCor());  
    }  
}
```

Cada instância da classe Carro
(meuCarro e seuCarro) tem sua cor.

[Compilando e Executando]

```
public class TesteCarro {  
  
    public static void main(String[] args) {  
        Carro meuCarro = new Carro();  
        meuCarro.definirCor("Vermelho");  
  
        Carro seuCarro = new Carro();  
        seuCarro.definirCor("Azul");  
  
        System.out.println("Meu carro é " + meuCarro.retornarCor());  
        System.out.println("Seu carro é " + seuCarro.retornarCor());  
    }  
}
```

Compilar

Executar

[Resultado do Teste]

```
public class TesteCarro {  
  
    public static void main(String[] args) {  
        Carro meuCarro = new Carro();  
        meuCarro.definirCor("Vermelho");  
  
        Carro seuCarro = new Carro();  
        seuCarro.definirCor("Azul");  
  
        System.out.println("Meu carro é " + meuCarro.retornarCor());  
        System.out.println("Seu carro é " + seuCarro.retornarCor());  
    }  
}
```

Meu carro é Vermelho
Seu carro é Azul

[Bibliografia]

- DEITEL, H. M.; DEITEL P. J. **Java: Como Programar**, 8a. Edição. Pearson, 2010.
 - Seções 1.5 a 1.10
 - Capítulo 3
- BOOCH, G., RUMBAUGH, J., JACOBSON, I. **UML, Guia do Usuário**. Rio de Janeiro: Campus, 2000.