

Objetos e Classes

Uma **classe** é um modelo usado para definir vários objetos com características semelhantes. Um programa é constituído de uma classe ou de um conjunto de classes.

Membros da classe (2 categorias):

- As variáveis, que especificam o estado da classe ou de um objeto instância desta classe.
- Os métodos, que especificam os mecanismos pelos quais a classe ou um objeto instância desta classe podem operar.

Classes

Esqueleto de uma classe:

```
class NomeDaClasse{  
    ...  
    TipoDaVariavel1 variavel1;  
    TipoDaVariavel2 variavel2;  
    ...  
    TipoDeRetorno1 metodo1(){  
        ...  
    }  
    TipoDeRetorno2 metodo2(){  
        ...  
    }  
    ...  
}
```

Classes

Esqueleto de uma classe:

```
public class NomeDaClasse{  
    ...  
    TipoDaVariavel1 variavel1;  
    TipoDaVariavel2 variavel2;  
    ...  
    TipoDeRetorno1 metodo1() {  
        ...  
    }  
    TipoDeRetorno2 metodo2() {  
        ...  
    }  
    ...  
}
```

Classes

Um objeto é uma instância de uma classe.

Um objeto precisa ser criado. Para que seja possível acessar as variáveis e os métodos de um objeto, é preciso atribuir uma referência ao objeto. O tipo de uma referência, ou seja, a classe à qual pertence o objeto ao qual ela vai se referir precisa ser declarada.

Declaração: `NomeDaClasse nomeDoObjeto;`

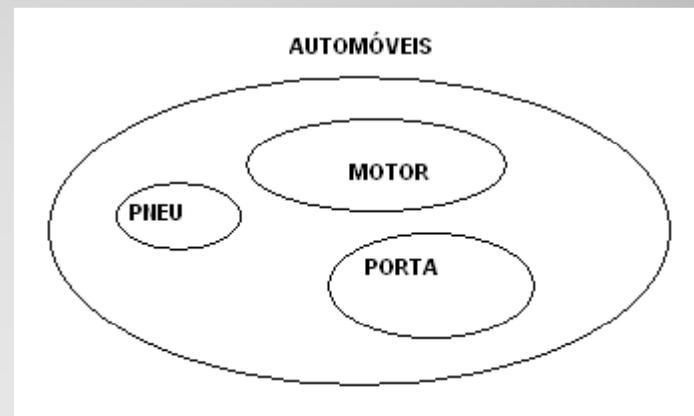
Criação: `nomeDoObjeto = new NomeDaClasse();`

`NomeDaClasse nomeDoObjeto = new NomeDaClasse();`

- `algumMehtodo(new AlgumaClasse());`

Objetos

```
Motor  
public int velocidade;  
public int endereco;  
public void altera_velocidade(int novav);
```



Classes e Objetos

Variáveis

- Variável de instância (default): valor é específico ao objeto e não à classe; possui um valor diferente pra cada objeto da classe.
- Variável de classe (static): valor comum a todos objetos representantes da classe.

**Variáveis de instancia e
variáveis de classe**

```
/* a classe definida neste arquivo pertence ao pacote
   chamado algumPacote */
package algumPacote;

public class nomeDaClasse{
/* a variável w é acessível por qualquer classe */
public int w;
/* a variável x é acessível pelos métodos da classe
   nomeDaClasse e das suas subclasses */
protected int x;
/* a variável y é acessível pelos métodos da classe
   nomeDaClasse e das outras classes que pertencem ao
   pacote chamado algumPacote */
int y;
/* a variável z é acessível somente pelos métodos da
   classe nomeDaClasse */
private int z;
...
}
```

Restrições de Acesso

```
TipoDaVariavel var;  
var = nomeDoObjeto.variavel;
```

```
TipoDaVariavel var = algumValor;  
nomeDoObjeto.variavel = var;
```

```
TipoDaVariavel var;  
var = nomeDoObjeto.variavel.outraVariavel;
```

**Acessando variáveis de outro
objeto**

```
TipoDaVariavel var;  
var = nomeDaClasse.variavel;  
  
public class nomeDaClasse{  
    private TipoDaVariavel variavel;  
    ...  
    public TipoDaVariavel getVariavel(){  
        return variavel;  
    }  
    public void setVariavel( TipoDaVariavel valor ){  
        /* aqui pode-se colocar testes para determinar se o  
        valor é aceitável */  
        ...  
        variavel = valor;  
    }  
}
```

**Acessando variáveis de outro
objeto**

```
TipoDaVariavel var, varp;  
var = nomeDoObjeto.getVariavel();  
...  
nomeDoObjeto.setVariavel( varp );
```

**Acessando variáveis de outro
objeto**

```
class nomeDaClasse{
TipoDaVariavel variavel1;
...
TipoDeRetorno nomeDoMetodo( )
{
TipoDaVariavel variavel2;
for( int i = 0; i < 10; i++ ){
    ...
}
}
}
```

Variáveis globais e locais

```
class nomeDaClasse{
  TipoDaVariavel variavel; // variável
  global
  ...
  TipoDeRetorno nomeDoMetodo()
  {
    TipoDaVariavel variavel; // variável
    local
    ...
    variavel2 = variavel;
    variavel3 = this.variavel;
  }
}
```

Variáveis globais e locais

Esta expressão um tanto paradoxal se refere a variáveis cujo valor não pode mudar durante a execução do programa. Por convenção recebem usualmente nomes escritos inteiramente em maiúsculas.

Por exemplo, na classe Math do pacote lang, encontramos

```
public static final double PI;
```

Variáveis constantes