

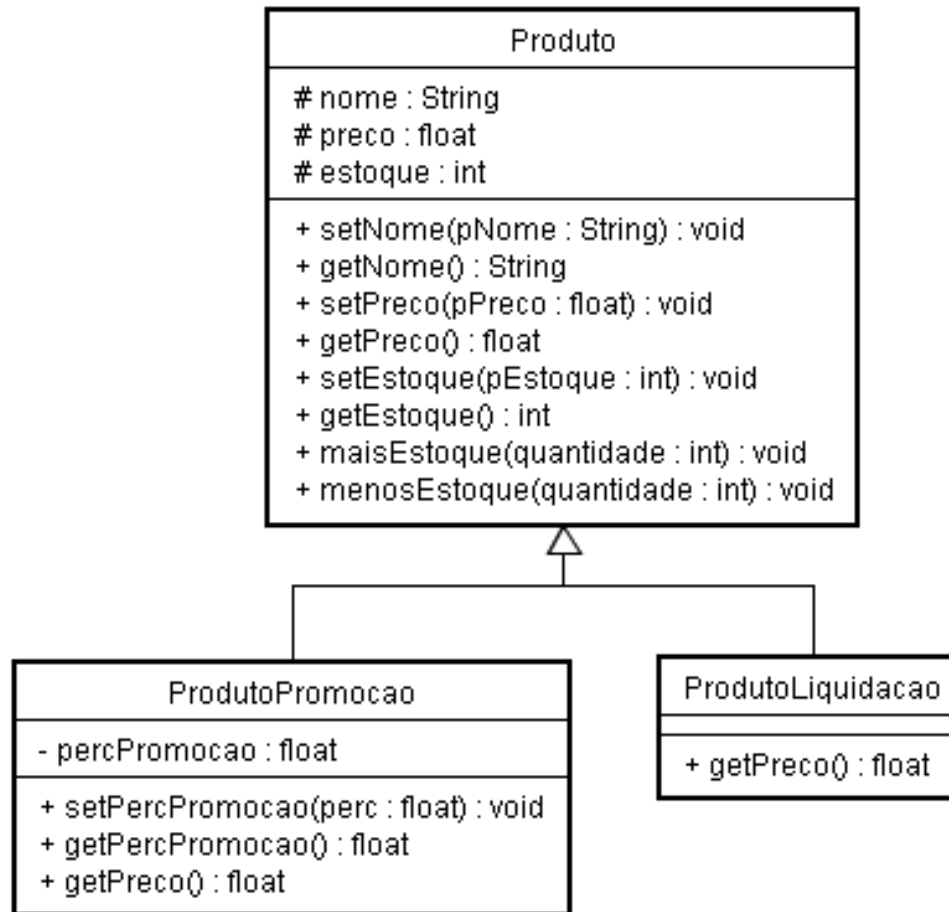
Desenvolvimento de Sistemas Orientados a Objeto

Programação Orientada a Objetos

Herança versus Composição

- ✓ Composição e herança são dois mecanismos para reutilizar funcionalidade
- ✓ A herança foi considerada a ferramenta básica de extensão e reuso de funcionalidade
- ✓ A composição estende uma classe pela delegação de trabalho para outro objeto
- ✓ A herança estende atributos e métodos de uma classe
- ✓ Hoje, considera-se que a composição é muito superior à herança na maioria dos casos

Herança versus Composição



Herança versus Composição

```
1 public class Produto {  
2     protected String nome;  
3     protected float preco;  
4     protected int estoque;  
5  
6     public void setNome(String pNome){  
7         nome = pNome;  
8     }  
9     public String getNome(){  
10        return nome;  
11    }  
12    public void setPreco(float pPreco){  
13        preco = pPreco;  
14    }  
15    public float getPreco(){  
16        return preco;  
17    }  
18    public void setEstoque(int pEstoque ){  
19        estoque = pEstoque;  
20    }  
21    public int getEstoque(){  
22        return estoque;  
23    }  
24 }
```

Herança versus Composição

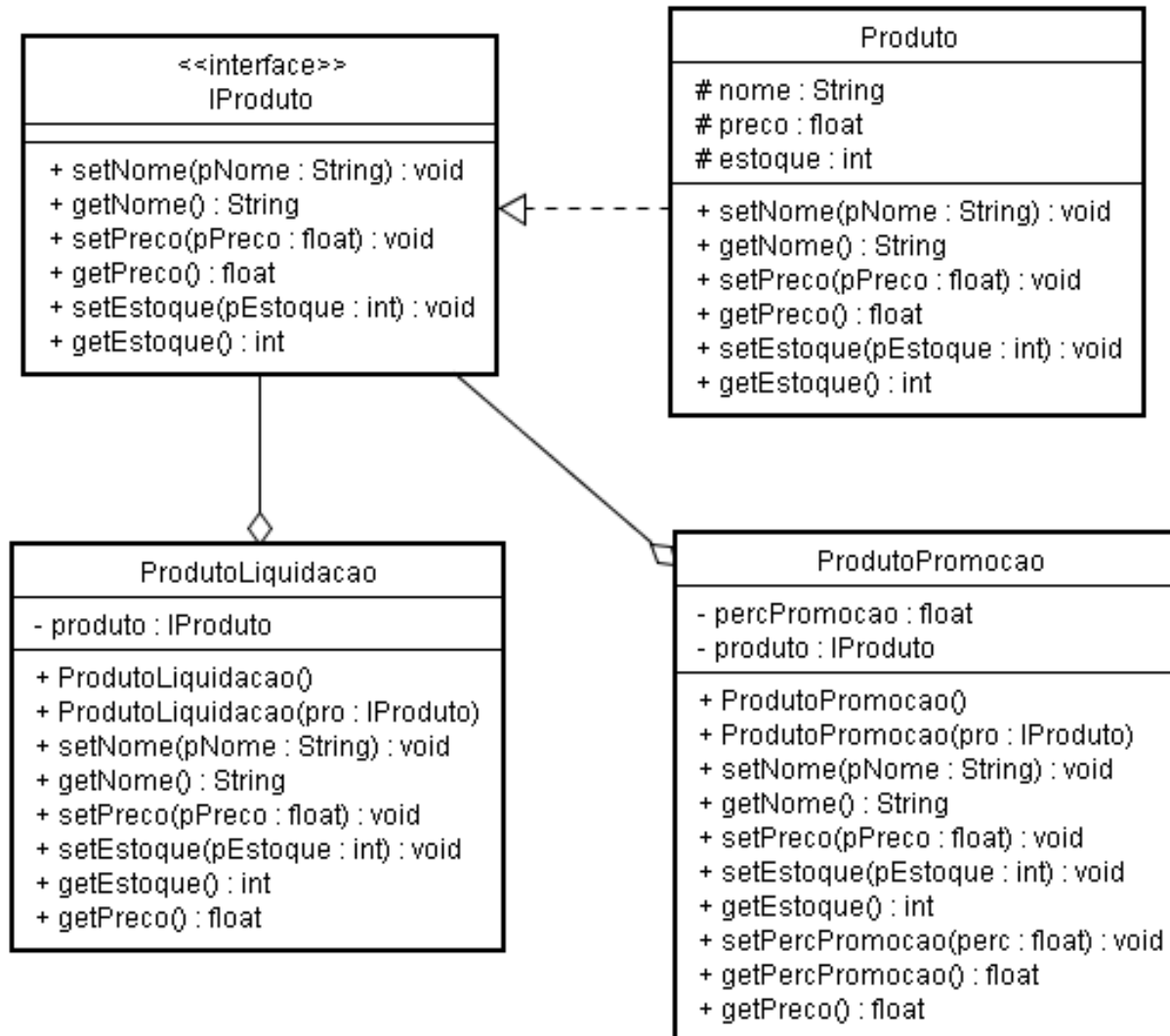
```
1 public class ProdutoPromocao extends Produto {
2
3     private float percPromocao;
4
5     public void setPercPromocao(float perc){
6         if(perc >=0 && perc < 50)
7             percPromocao = perc;
8     }
9     public float getPercPromocao(){
10         return percPromocao;
11     }
12     public float getPreco(){
13         return preco - (preco * percPromocao / 100);
14     }
15 }
```

```
1 public class ProdutoLiquidacao extends Produto {
2
3     public float getPreco(){
4         return preco / 2;
5     }
6 }
```

Herança versus Composição

- ✓ A herança estabelece uma relação estática, decidida em tempo de compilação
- ✓ As subclasses estão fortemente acopladas à superclasse
- ✓ Mudanças na superclasse afetam diretamente as subclasses
- ✓ ProdutoPromocao e ProdutoLiquidacao não são subtipos e sim papéis que o produto pode assumir por um período (temporal)

Herança versus Composição



Herança versus Composição

```
1 public interface IProduto {  
2     public void setNome(String pNome);  
3     public String getNome();  
4     public void setPreco(float pPreco);  
5     public float getPreco();  
6     public void setEstoque(int pEstoque );  
7     public int getEstoque();  
8 }
```


Herança versus Composição

```
1 public class Produto implements IProduto{
2     protected String nome;
3     protected float preco;
4     protected int estoque;
5
6     public void setNome(String pNome){
7         nome = pNome;
8     }
9     public String getNome(){
10         return nome;
11     }
12     public void setPreco(float pPreco){
13         preco = pPreco;
14     }
15     public float getPreco(){
16         return preco;
17     }
18     public void setEstoque(int pEstoque ){
19         estoque = pEstoque;
20     }
21     public int getEstoque(){
22         return estoque;
23     }
24 }
```

Herança versus Composição

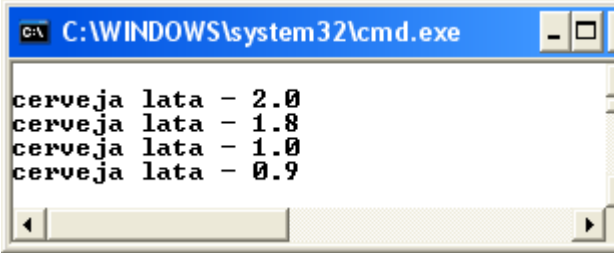
```
1 public class ProdutoLiquidacao implements IProduto {
2
3     private IProduto produto;
4
5     public ProdutoLiquidacao(){
6         produto = new Produto();
7     }
8     public ProdutoLiquidacao(IProduto pro){
9         produto = pro;
10    }
11    public void setNome(String pNome){
12        produto.setNome(pNome);
13    }
14    public String getNome(){
15        return produto.getNome();
16    }
17    public void setPreco(float pPreco){
18        produto.setPreco(pPreco);
19    }
20    public void setEstoque(int pEstoque ){
21        produto.setEstoque(pEstoque);
22    }
23    public int getEstoque(){
24        return produto.getEstoque();
25    }
26    public float getPreco(){
27        return produto.getPreco() / 2;
28    }
29 }
```

Herança versus Composição

```
1 public class ProdutoPromocao implements IProduto{
2
3     private float percPromocao;
4     private IProduto produto;
5
6     public ProdutoPromocao(){
7         produto = new Produto();
8     }
9     public ProdutoPromocao(IProduto pro){
10         produto = pro;
11     }
12     public void setNome(String pNome){
13         produto.setNome(pNome);
14     }
15     public String getNome(){
16         return produto.getNome();
17     }
18     public void setPreco(float pPreco){
19         produto.setPreco(pPreco);
20     }
21     public void setEstoque(int pEstoque ){
22         produto.setEstoque(pEstoque);
23     }
24     public int getEstoque(){
25         return produto.getEstoque();
26     }
27     public void setPercPromocao(float perc){
28         if(perc >=0 && perc < 50)
29             percPromocao = perc;
30     }
31     public float getPercPromocao(){
32         return percPromocao;
33     }
34     public float getPreco(){
35         return produto.getPreco() - (produto.getPreco() * percPromocao / 100);
36     }
37 }
```

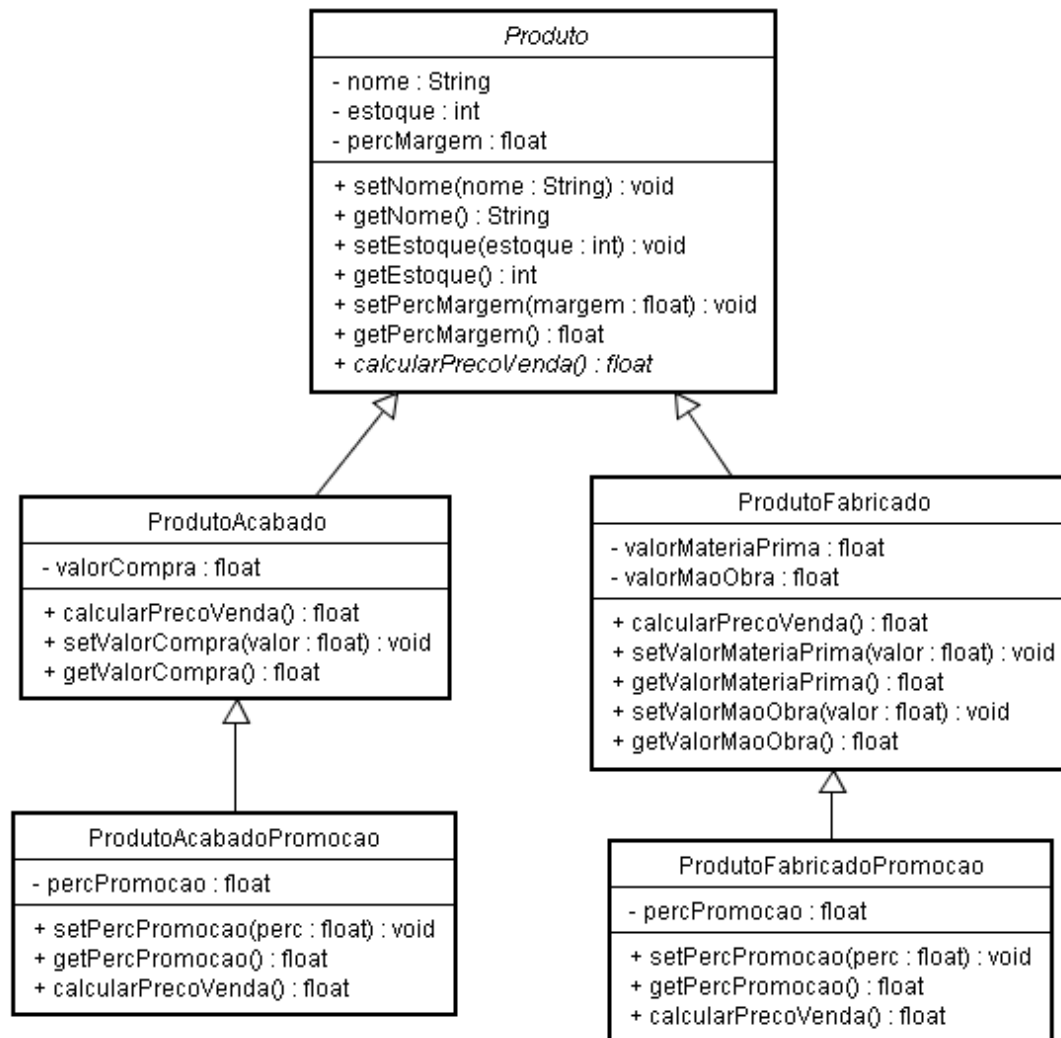
Herança versus Composição

```
1 public class TesteProdutos {
2     public static void main(String args[]){
3
4         IProduto produto;
5         ProdutoPromocao produtoProm;
6         ProdutoLiquidacao produtoLiq, produtoSuperLiq;
7
8         produto = new Produto();
9         produto.setNome("cerveja lata");
10        produto.setPreco(2.00f);
11        produto.setEstoque(3000);
12
13        System.out.println(produto.getNome()+" - "+produto.getPreco());
14
15        produtoProm = new ProdutoPromocao(produto);
16        produtoProm.setPercPromocao(10.0f);
17        System.out.println(produtoProm.getNome()+" - "+produtoProm.getPreco());
18
19        produtoLiq = new ProdutoLiquidacao(produto);
20        System.out.println(produtoLiq.getNome()+" - "+produtoLiq.getPreco());
21
22        produtoSuperLiq = new ProdutoLiquidacao(produtoProm);
23        System.out.println(produtoSuperLiq.getNome()+" - "+produtoSuperLiq.getPreco());
24
25    }
26 }
```

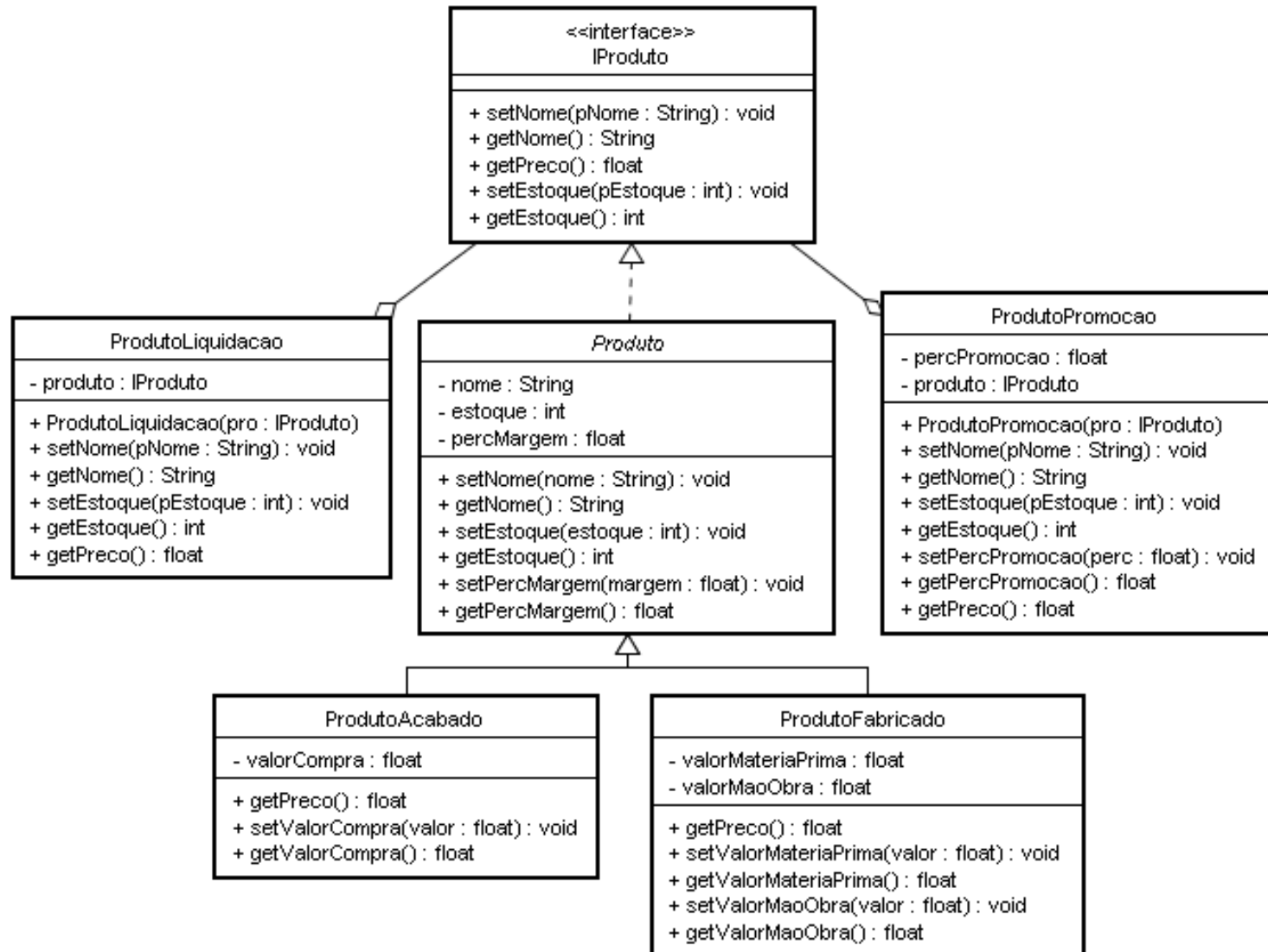


Com a composição, é possível trocar a classe do objeto em tempo de execução

Herança versus Composição



Herança versus Composição



Herança versus Composição

```
1 public interface IProduto {  
2     public void setNome(String pNome);  
3     public String getNome();  
4     public float getPreco();  
5     public void setEstoque(int pEstoque );  
6     public int getEstoque();  
7 }
```

Herança versus Composição

```
1 public abstract class Produto implements IProduto{
2
3     private String nome;
4     private int estoque;
5     private float percMargem;
6
7     public void setNome(String nome) {
8         this.nome = nome;
9     }
10    public String getNome() {
11        return nome;
12    }
13    public void setEstoque(int estoque) {
14        this.estoque = estoque;
15    }
16    public int getEstoque() {
17        return estoque;
18    }
19    public void setPercMargem(float margem) {
20        this.percMargem = margem;
21    }
22    public float getPercMargem() {
23        return percMargem;
24    }
25 }
```


Herança versus Composição

```
1 public class ProdutoAcabado extends Produto {
2
3     private float valorCompra;
4
5     public float getPreco(){
6         float valorMargem;
7         valorMargem = valorCompra * getPercMargem() / 100;
8         return valorCompra + valorMargem;
9     }
10    public void setValorCompra(float valor) {
11        valorCompra = valor;
12    }
13    public float getValorCompra() {
14        return valorCompra;
15    }
16 }
```

Herança versus Composição

```
1 public class ProdutoFabricado extends Produto {
2
3     private float valorMateriaPrima;
4     private float valorMaoObra;
5
6     public float getPreco(){
7         float valorBase, valorMargem;
8         valorBase = valorMateriaPrima + valorMaoObra;
9         valorMargem = valorBase * getPercMargem() / 100;
10        return valorBase + valorMargem;
11    }
12    public void setValorMateriaPrima(float valor) {
13        this.valorMateriaPrima = valor;
14    }
15    public float getValorMateriaPrima() {
16        return valorMateriaPrima;
17    }
18    public void setValorMaoObra(float valor) {
19        this.valorMaoObra = valor;
20    }
21    public float getValorMaoObra() {
22        return valorMaoObra;
23    }
24 }
```

Herança versus Composição

```
1 public class ProdutoLiquidacao implements IProduto {
2
3     private IProduto produto;
4
5     public ProdutoLiquidacao(IProduto pro){
6         produto = pro;
7     }
8     public void setNome(String pNome){
9         produto.setNome(pNome);
10    }
11    public String getNome(){
12        return produto.getNome();
13    }
14    public void setEstoque(int pEstoque ){
15        produto.setEstoque(pEstoque);
16    }
17    public int getEstoque(){
18        return produto.getEstoque();
19    }
20    public float getPreco(){
21        return produto.getPreco() / 2;
22    }
23 }
```

Herança versus Composição

```
1 public class ProdutoPromocao implements IProduto{
2
3     private float percPromocao;
4     private IProduto produto;
5
6     public ProdutoPromocao(IProduto pro){
7         produto = pro;
8     }
9     public void setNome(String pNome){
10         produto.setNome(pNome);
11     }
12     public String getNome(){
13         return produto.getNome();
14     }
15     public void setEstoque(int pEstoque ){
16         produto.setEstoque(pEstoque);
17     }
18     public int getEstoque(){
19         return produto.getEstoque();
20     }
21     public void setPercPromocao(float perc){
22         if(perc >=0 && perc < 50)
23             percPromocao = perc;
24     }
25     public float getPercPromocao(){
26         return percPromocao;
27     }
28     public float getPreco(){
29         return produto.getPreco() - (produto.getPreco() * percPromocao / 100);
30     }
31 }
```

Herança versus Composição

```
1 public class TesteProdutos {
2     public static void main(String args[]){
3
4         ProdutoAcabado produtoA;
5         ProdutoFabricado produtoF;
6         ProdutoPromocao produtoProm;
7         ProdutoLiquidacao produtoLiq, produtoSuperLiq;
8
9         produtoA = new ProdutoAcabado();
10        produtoA.setNome("cerveja lata");
11        produtoA.setPercMargem(50);
12        produtoA.setValorCompra(2.00f);
13        System.out.println(produtoA.getNome()+" - "+produtoA.getPreco());
14
15        produtoF = new ProdutoFabricado();
16        produtoF.setNome("sanduiche");
17        produtoF.setPercMargem(20.00f);
18        produtoF.setValorMateriaPrima(2.00f);
19        produtoF.setValorMaoObra(1.00f);
20        System.out.println(produtoF.getNome()+" - "+produtoF.getPreco());
21
22        produtoProm = new ProdutoPromocao(produtoA);
23        produtoProm.setPercPromocao(10.0f);
24        System.out.println(produtoProm.getNome()+" - "+produtoProm.getPreco());
25
26        produtoLiq = new ProdutoLiquidacao(produtoF);
27        System.out.println(produtoLiq.getNome()+" - "+produtoLiq.getPreco());
28
29        produtoSuperLiq = new ProdutoLiquidacao(produtoProm);
30        System.out.println(produtoSuperLiq.getNome()+" - "+produtoSuperLiq.getPreco());
31    }
32 }
```

