



INSTITUTO FEDERAL DE  
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA  
RIO GRANDE DO NORTE

# AULA 14

## MODIFICADORES DE ACESSO

**Disciplina: Programação Orientada a Objetos**

**Professora: Alba Lopes**

**[alba.lopes@ifrn.edu.br](mailto:alba.lopes@ifrn.edu.br)**

# MODIFICADORES DE ACESSO

- Os modificadores de acesso são empregados para restringir o acesso aos atributos e métodos de uma classe.

```
public class ContaCorrente {  
    float saldo;
```

- Exemplo:

```
//..
```

```
boolean sacar(float valor){  
    if (saldo >= valor){  
        saldo = saldo - valor;  
        return true;  
    }else{  
        return false;  
    }  
}
```

```
}
```



# MODIFICADORES DE ACESSO

- Da forma como nossa classe foi construída, por mais que modifiquemos o atributo saldo através do método **sacar**, é possível atribuir qualquer valor ao atributo saldo.

```
public static void main(String[] args) {  
  
    ContaCorrente conta = new ContaCorrente();  
    conta.definirSaldoInicial(1000);  
    conta.sacar(300);  
    conta.saldo = -100;  
  
}
```



# MODIFICADORES DE ACESSO

- A melhor forma de resolver isso seria forçar quem usa a classe a chamar sempre o método sacar e não permitir que seja atribuído um valor diretamente ao atributo saldo.
- Para fazer isso em Java, basta declarar que os atributos não podem ser acessados fora da classe usando a palavra chave **private**.

```
public class ContaCorrente {  
    private float saldo;  
}
```



# MODIFICADORES DE ACESSO

- Isso não é conveniente, pois pode ser que um valor inválido seja atribuído ao valor do saldo.
- A melhor forma de resolver isso seria forçar quem usa a classe ContaCorrente a chamar sempre o método sacar e não permitir que seja atribuído um valor diretamente ao atributo saldo.
- Para fazer isso em Java, basta declarar que os atributos não podem ser acessados fora da classe usando a palavra chave **private**.



# MODIFICADORES DE ACESSO

```
public class ContaCorrente {  
    private float saldo;  
}
```

- **private** é um modificador de acesso (também chamado de modificador de visibilidade)
- Marcando um atributo como privado, fechamos o acesso a ele a partir de outras classes.
- É uma prática quase que **obrigatória** proteger os atributos de suas classes como **private**
- Quem chama o método **sacar** não precisa saber que o saldo está sendo checado. Quem for usar essa classe, basta saber *o que o método faz*, e não *como ele faz*.



# MODIFICADORES DE ACESSO

- A palavra chave **private** também pode ser utilizada para modificar o acesso a um método.
  - Isso é usado quando existe um método apenas auxiliar à própria classe, e não queremos que outros o enxerguem.
- Há também o modificador **public**, que permite a todos acessarem um determinado atributo ou método
- É muito comum que atributos sejam **private** e quase todos os métodos sejam **públic** (não é uma regra)
  - Assim, toda conversa de um objeto com outro é feita através de troca de mensagem (acessando seus métodos)



# MODIFICADORES DE ACESSO

- Exemplo:

```
public class ContaCorrente {  
    private float saldo;  
    private Cliente clienteConta;  
  
    public void definirSaldoInicial(float saldoInicial){  
        saldo = saldoInicial;  
    }  
  
    public void depositar (float valor){  
        saldo = saldo + valor;  
    }  
  
    //..
```





# MODIFICADORES DE ACESSO

- Java possui os seguintes modificadores de acesso

<b>public</b>	É visível em qualquer lugar.
<b>protected</b>	Só é visível na mesma classe e em suas subclasses.
<b>package</b>	Default. Só é visível em classes do mesmo pacote. Não existe modificador com este nome. A ausência de modificador o torna package.
<b>private</b>	Só é visível dentro da mesma classe.



# ENCAPSULAMENTO

- Serve para ocultar os dados
- Evita que o usuário acesse membros que ele não precisa manipular
  - Proteção do código
- Permite a modificação interna de uma classe sem alterar a sua funcionalidade e o modo como é acessada
- Utilizado em Java através dos modificadores de acesso.



# MÉTODOS GETTERS E SETTERS

- Para permitir o acesso aos atributos (já que eles são **private**) de uma maneira controlada, a prática mais comum é criar dois métodos
  - Um que retorna o valor
  - E outro que muda o valor
- O padrão para esses métodos é colocar a palavra **get** ou **set** antes do nome do atributo
- O padrão do método **get** não vale para variáveis do tipo **boolean**
  - Esses atributos são acessados via **is** e **set**
  - Exemplo: para verificar se uma lâmpada está acesa, seriam criados os métodos **isLigado** e **setLigado**



# MÉTODOS GETTERS E SETTERS

## ○ Exemplo:

- Incluindo o modificador **private** nos atributos da nossa Cliente, ela ficará como:

```
public class Cliente {  
  
    private String nome;  
    private String endereco;  
    private float renda;  
    private String profissao;  
  
}
```



# MÉTODOS GETTERS E SETTERS

## ○ Exemplo:

- E então, incluindo os métodos **get** para cada um dos atributos:

```
//..
```

```
public String getNome() {  
    return nome;  
}
```

```
public String getEndereco() {  
    return endereco;  
}
```

```
public float getRenda() {  
    return renda;  
}
```

```
public String getProfissao() {  
    return profissao;  
}
```



# MÉTODOS GETTERS E SETTERS

## ○ Exemplo:

- E também o método set:

```
//..  
public void setNome(String novoNome) {  
    nome = novoNome;  
}  
  
public void setEndereco(String novoEndereco) {  
    endereco = novoEndereco;  
}  
  
public void setRenda(float novaRenda) {  
    renda = novaRenda;  
}  
  
public void setProfissao(String novaProfissao) {  
    profissao = novaProfissao;  
}
```



# MÉTODOS GETTERS E SETTERS

- Exemplo:

```
public class ContaCorrente {
    private float saldo;
    private Cliente clienteConta;

    public float getSaldo() {
        return saldo;
    }

    public void setSaldo(float novoSaldo) {
        saldo = novoSaldo;
    }

    public Cliente getClienteConta() {
        return clienteConta;
    }

    public void setClienteConta(Cliente novoCliente) {
        clienteConta = novoCliente;
    }
}
```



# EXERCÍCIO

- Crie um atributo **numero** na classe ContaCorrente. Defina o atributo numero como **private**. Crie seus métodos **get** e **set**.





# REFERÊNCIAS

- <http://www.hardware.com.br/artigos/programacao-orientada-objetos/>
- <http://www.fontes.pro.br/educacional/materialpaginas/java/arquivos/jdbc/jdbc.php>
- <http://www.dm.ufscar.br/~waldeck/curso/java>
- PORTAL EDUCAÇÃO - Cursos Online : Mais de 900 cursos online com certificado  
<http://www.portaleducacao.com.br/informatica/artigos/7852/moderadores-de-acesso#ixzz2AAmxO3JD>
- <http://www.slideshare.net/regispires/java-08-modificadores-acesso-e-membros-de-classe-presentation>

