

Linguagem de Programação

Aula 14
Modificadores de classes - Get e Set

Modificadores de Acesso

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

- Exemplo:

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

- Java possui os seguintes modificadores de acesso

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

```
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;  
    }  
}
```

Obrigado!

