

GET e SET

Nessa aula falaremos sobre os get e sets:

A melhor forma que acessarmos os atributos de uma classe é utilizando métodos. Os métodos GET e SET são técnicas padronizadas para gerenciamento sobre o acesso dos atributos. Nesses métodos determinamos quando será alterados um atributo e o acesso ao mesmo, tornando o controle e modificações mais práticas e limpas, sem contudo precisar alterar assinatura do método usado para acesso ao atributo.

Como criar métodos GET e SET

Na criação dos métodos para acesso a esses atributos privados devemos colocar GET ou SET antes do nome do atributo. Porém, existem diferenças entre os métodos, pois modularizamos um procedimento para método SET e uma função para método GET.

Método Get

Quando formos acessar, “pegar” alguns atributos da classe, devemos utilizar os métodos GET. Esse método sempre retornará um valor, seja ele String, int, double etc. Então devemos criar uma função. Vejamos:

```
1 public String getNomeProduto() {  
2     return nomeproduto;  
3 }  
4  
5 public int getQuantidade() {  
6     return quantidade;  
7 }  
8  
9 public String getValorUnitario() {  
10    return valorunitario;  
11 }
```

Método set

Para alterarmos, modificarmos os valores de um atributo da classe de maneira protegida, utilizamos os métodos SET. Esse método não terá um retorno, pois o atributo será somente modificado, criando um método de tipo VOID, sem retorno. Porém ele deve receber algum argumento para que possa ocorrer a devida alteração. Vejamos:

```
1 public void setNomeProduto(String nomeproduto) {
2     this.nomeproduto = nomeproduto;
3 }
4
5 public void setQuantidade(int quantidade) {
6     this.quantidade = quantidade;
7 }
8
9 public void setValorUnitario(String valorunitario) {
10    this.valorunitario = valorunitario;
11 }
```

Como usar os métodos get e set

Usamos get para obter informações. Esse tipo de método sempre retorna um valor. Usamos set para definir valores. Esse tipo de método geralmente não retorna valores.

getSet.java

```
public class getSet{

    public static void main(String[] args){
        String nome = "Neil Peart";
        int ID=2112;
        double salario = 1000000;

        Funcionario chefe = new Funcionario();

        chefe.setNome(nome);
        chefe.setID(ID);
        chefe.setSalario(salario);

        chefe.exibir();
    }
}
```

Dia 13 - Gets e Sets

Exemplo 2:

```
public class Funcionario {
    private String nome;
    private int ID;
    private double salario;

    public void exibir(){
        System.out.printf("O funcionário %s, de número %d recebe %.2f por mês", getNome(),getID(),getSalario());
    }

    public void setNome( String nome ){
        this.nome = nome;
    }

    public void setID( int ID ){
        this.ID = ID;
    }

    public void setSalario( double salario ){
        this.salario = salario;
    }

    public String getNome(){
        return this.nome;
    }

    public int getID(){
        return this.ID;
    }

    public double getSalario(){
        return this.salario;
    }
}
```

Como invocar métodos de dentro do construtor

Outra maneira de inicializarmos as variáveis de um Objeto, é usando os métodos 'set', direto do construtor:

setGet.java

```
public class setGet{

    public static void main(String[] args){
        String nome = "Neil Peart";
        int ID=2112;
        double salario = 1000000;

        Funcionario chefe = new Funcionario(nome, ID, salario);

        chefe.exibir();
    }
}
```

Dia 13 - Gets e Sets

```
public class Funcionario {
    private String nome;
    private int ID;
    private double salario;

    public Funcionario( String nome, int ID, double salario){
        setNome(nome);
        setID(ID);
        setSalario(salario);
    }

    public void exibir(){
        System.out.printf("O funcionário %, de número %d recebe %.2f por mês", getNome(),getID(),getSalario());
    }

    public void setNome( String nome ){
        this.nome = nome;
    }

    public void setID( int ID ){
        this.ID = ID;
    }

    public void setSalario( double salario ){
        this.salario = salario;
    }

    public String getNome(){
        return nome;
    }

    public int getID(){
        return ID;
    }

    public double getSalario(){
        return salario;
    }
}
```

Uso errado do get e set

Também vemos muitas utilizações erradas do get e set mas Qual a diferença entre alterar e pegar o valor de uma variável diretamente (caso ela seja pública), e alterar e pegar o valor dela através de set e get (se ela for privada)? Não é segurança, já que podemos alterar seu valor como bem quisermos, através dos métodos set.

Definir get e set para todas as variáveis, de forma direta, é um exemplo de péssimo hábito de programação. Infelizmente é uma técnica bem comum e muitos programadores, até os mais experientes, cometem esse ato inútil.

Para que servem então? Quando usar?

Só use set e get quando você for fazer algo além de simplesmente mudar o valor de uma variável 'private'.

Lá no começo de nosso tutorial, usamos o exemplo de contador, que limita o número de saques de um cliente a um caixa eletrônico.

Outro exemplo disso, seria em um aplicativo de vendas.

Ao comprar, o método set ou get automaticamente define o valor de desconto ou juros.

Dia 13 - Gets e Sets

Assim esse método faz algo mais além do que mudar a variável. Assim, sua aplicação se torna robusta e segura, pois se torna impossível mudar o valor da variável sem que seja alterada o valor dos juros ou desconto.

Sempre que seu método set do tipo for:

```
retorno metodo(tipo valor){  
  
this.nome_da_variavel = valor;  
  
}
```

Muito provavelmente ele é inútil. Para saber quando usar é simples: use quando for fazer algo além de fornecer valor.

Um exemplo util:

```
retorno metodo(tipo valor){  
  
this.nome_da_variavel = valor * juros;  
  
};
```