



Universidade Estadual da Paraíba
Campus VII – Gov. Antônio Mariz
CCEA – Centro de Ciências Exatas e Sociais Aplicadas
Licenciatura em Computação
Java - Noturno

Exercício Prático 2

Nome: _____

Dada a classe Pessoa:

```
public class Pessoa{
    private int idade;
    private String nome;

    //construtor
    public Pessoa(){

    }

    /**
     * Altera o nome da Pessoa.
     * @param nome O nome da pessoa
     */
    public void setNome(String nome){
        this.nome = nome;
    }

    /**
     * Altera a idade da Pessoa.
     * @param nome A idade da pessoa em anos.
     */
    public void setIdade(int idade){
        this.idade = idade;
    }

    /**
     * Resgata a idade da Pessoa.
     * @return idade O valor da idade em anos.
     */
    public int getIdade(){
        return this.idade;
    }

    /**
     * Resgata o nome da Pessoa.
     * @return nome O nome da pessoa no formato alfanumerico.
     */
    public String getNome(){
        return this.nome;
    }
}
```

Sabemos que o paradigma de orientação à objetos tende a representar coisas reais através da linguagem, onde os atributos `idade` e `nome` são propriedades do Objeto Pessoa, já os métodos representam comportamentos dos atributos.. Com base nisso, modifique o código adicionando atributos e que por padrão os atributos são privados, pois existem métodos que se encarregam de alterar e resgatar os valores dos atributos que são os `set's` e `get's` respectivamente.

Com base nisso, adicione 2(dois) ou mais atributos ao objeto pessoa e seus métodos `get's` e `set's` respectivamente.

Ao finalizar essa modificação, crie uma classe que conterà o método `main` e faça testes com os métodos e os atributos criados. Exemplo:

```
...
public static void main(String args[]){
    Scanner teclado = new Scanner(System.in);
    //instanciando a classe pessoa
    Pessoa fulano = new Pessoa();

    System.out.println("Digite o nome: ");
    //utilizando o método set para alterar o valor do atributo da classe pessoa
    que até então está vazio.
    fulano.setNome(teclado.nextLine());
    //forma de acessar o método através da instância da classe Pessoa
    System.out.println("O nome é: "+ fulano.getNome());
}
...
```

Avançando um pouco mais o nosso exercício, vamos modificar o **construtor** da classe. O construtor é um método padrão que é chamado assim que a classe é instanciada com a palavra **new**. Ex.: **new Pessoa()**; O construtor permite que os atributos de uma classe iniciem com valores passados por parâmetros no construtor e cada classe pode conter vários construtores. Como exemplo abaixo demonstra, a necessidade da classe poder iniciar já com um “nome” associado à Pessoa, poderíamos ter um construtor como segue abaixo:

```
...
public Pessoa(String nome){
    this.nome = nome;
}
...
```

Agora que temos a possibilidade de criar instâncias da classe Pessoa com inicialização de seus atributos, crie mais de um construtor de acordo com a necessidade dos atributos que você criou anteriormente e logo após faça um teste usando uma classe que contenha um método principal. Vale salientar que todos os métodos devem ter comentários do tipo javadoc e ao finalizar este exercício prático deve-se gerar um javadoc da sua classe pessoa, que poderá ser reutilizada futuramente.

Comandos importantes:

<code>ls</code>	lista os arquivos de um determinado diretório;
<code>pwd</code>	mostra o diretório atual;
<code>mkdir <nome></code>	cria um diretório com o <nome> passado como parâmetros;
<code>cd <nome></code>	direciona o terminal para o diretório com o <nome>;
<code>javac <nome>.java</code>	compila um arquivo com o nome passado e gera o bytecode;
<code>java <nome></code>	executa o bytecode gerado pelo processo de compilação;
<code>javadoc <nome>.java</code>	gera o javadoc do arquivo <nome>.java;

É importante saber que as “...” reticências indicam a existência de um código implícito.