

MÓDULO 04 – PRÁTICA DE LABORATÓRIO I

1 INTRODUÇÃO

Nos tópicos a seguir **estaremos praticando os conceitos visto no módulo 4 de Programação I**. Revise estes conceitos nos seguintes capítulos do livro de apoio de Programação I:

- Capítulo 04 – CLASSES E OBJETOS;
- Capítulo 05 – MÉTODOS DE CONFIGURAÇÃO E ACESSO AOS ATRIBUTOS.

Na sequência, leia atentamente o resumo dos conceitos e os exemplos apresentados neste documento e tire suas dúvidas com o tutor no fórum de dúvidas.

Atenção: Para que você possa comparar as suas classes, com as classes desenvolvidas neste material, faça o download deste projeto que está disponível para download no módulo 04.

2 MÉTODOS DE CONFIGURAÇÃO E ACESSO AOS ATRIBUTOS

No capítulo 05 do livro de apoio de Programação I, vimos que podemos utilizar métodos para atribuir valores e recuperar os valores armazenados nos atributos dos objetos criados em memória. Os métodos de configuração e acesso aos atributos nos permitirão: alterar os valores dos atributos dos objetos criados e visualizar os valores dos atributos dos objetos, respectivamente.

Como vimos no módulo anterior, o método construtor, pode receber valores por meio dos parâmetros para: configurar e/ou atribuir valores para os atributos do objeto a ser criado. Porém, **o método construtor, somente poderá ser executado durante a criação do objeto**, por meio do operador: **new**. O que fazemos então se precisarmos alterar o valor de um atributo, depois da criação do objeto em memória? Para isso vamos então utilizar os métodos de configuração dos atributos.

Um método de configuração de um atributo é um método que é acionado, para enviar um valor para ser armazenado no atributo. Costuma-se usar a palavra **set** seguida do nome do atributo para compor o identificador (nome) do método de configuração. No corpo do método devemos ter um comando de atribuição, responsável pelo armazenamento do valor no atributo. Veja o exemplo deste tipo de método para o atributo: **modeloProcessador** da classe: **Computador**:

```
//método de configuração de valor para o atributo: modeloProcessador  
public void setModeloProcessador(String mod){  
    modeloProcessador = mod;  
}
```

Principais características do método de configuração dos atributos:

- Não possui retorno, portanto utiliza a palavra: void após o identificador: public;
- Possui o prefixo: **set** sempre antes do nome do atributo;
- Recebe parâmetro do tipo de dado do atributo que estará recebendo o valor;
- Atribui o valor do parâmetro para o atributo referenciado no nome do método.

No módulo anterior, também vimos que para visualizar (acessar) os valores dos atributos de um objeto criado em memória, precisamos utilizar a opção: Inspect. Porém, será esta a única forma que podemos utilizar para acessar os valores de um atributo? E será que esta é a melhor forma de realizar este tipo de acesso? A respostas para as duas perguntas é: Não. Então, para acessarmos os valores dos atributos dos objetos criados em memória, estaremos utilizando os métodos de acesso aos atributos.

Um método de acesso a um atributo é um método que quando acionado retorna o valor armazenado no atributo. Costuma-se usar a palavra **get** seguida do nome do atributo para compor o identificador (nome) do método de acesso. No corpo do método devemos ter o comando `return`, responsável pelo retorno do valor. Veja o exemplo deste tipo de método para o atributo: **modeloProcessador** da classe: **Computador**:

```
//método de acesso ao valor do atributo: modeloProcessador  
public String getModeloProcessador(){  
    return modeloProcessador;  
}
```

Principais características do método de acesso dos atributos:

- Possui retorno do mesmo tipo de dado do atributo, este tipo deve ser informado após o identificador: `public`;
- Possui o prefixo: **get** sempre antes do nome do atributo;
- Não recebe nenhum parâmetro;
- Utiliza o comando: **return** para retornar o valor do atributo do objeto.

Importante: revise o capítulo 05 – Métodos de Configuração e Acesso aos Atributos do livro de apoio de Programação I para verificar os detalhes sobre estes métodos.

2.1 EXEMPLO MÉTODOS DE CONFIGURAÇÃO E ACESSO AOS ATRIBUTOS

Vamos então praticar a criação dos métodos de configuração e acesso aos atributos da classe: **Computador**. Para isso, será necessário criar um método de configuração e um método de acesso para cada atributo da classe. A classe **Computador** ficará então com a seguinte estrutura:

```
//método de configuração de valor para o atributo: modeloProcessador
public void setModeloProcessador(String mod){
    modeloProcessador = mod;
}

//método de acesso ao valor do atributo: modeloProcessador
public String getModeloProcessador(){
    return modeloProcessador;
}

//método de configuração de valor para o atributo: capacidadeProcessador
public void setCapacidadeProcessador(double cap){
    capacidadeProcessador = cap;
}

//método de acesso ao valor do atributo: capacidadeProcessador
public double getCapacidadeProcessador(){
    return capacidadeProcessador;
}

//método de configuração de valor para o atributo: quantidadeMemoriaRAM
public void setQuantidadeMemoriaRAM(int qua){
    quantidadeMemoriaRAM = qua;
}

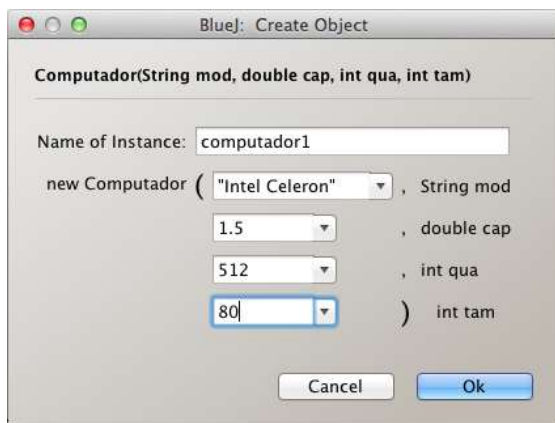
//método de acesso ao valor do atributo: quantidadeMemoriaRAM
public int getQuantidadeMemoriaRAM(){
    return quantidadeMemoriaRAM;
}

//método de configuração de valor para o atributo: tamanhoHD
public void setTamanhoHD(int tam){
    tamanhoHD = tam;
}

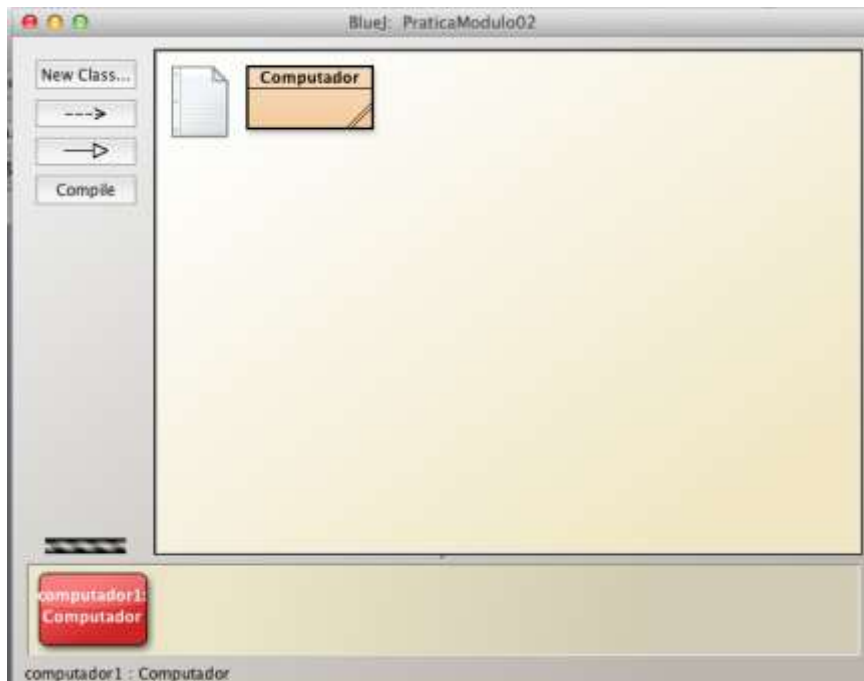
//método de acesso ao valor do atributo: tamanhoHD
public int getTamanhoHD(){
    return tamanhoHD;
}
```

Depois de editarmos a classe: Computador e criarmos os métodos de configuração e acesso aos atributos, devemos compilar a classe e fechar a edição.

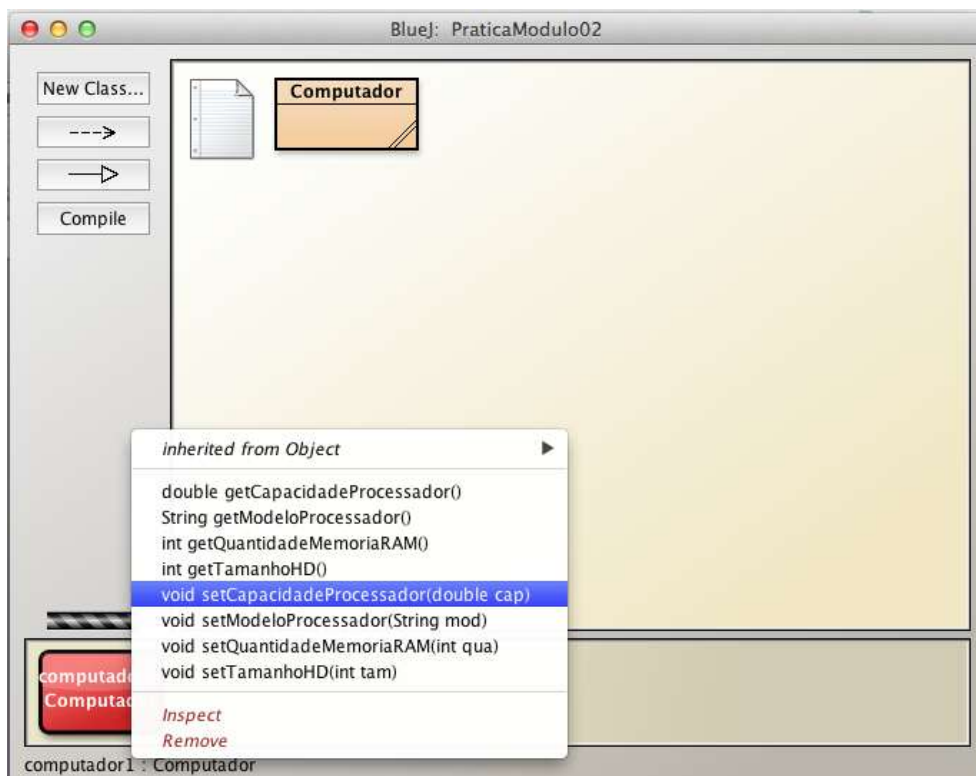
Na sequencia, estamos aptos para testar os métodos, para tanto, deveremos criar novos objetos por meio do construtor, opção: new, conforme figura abaixo:



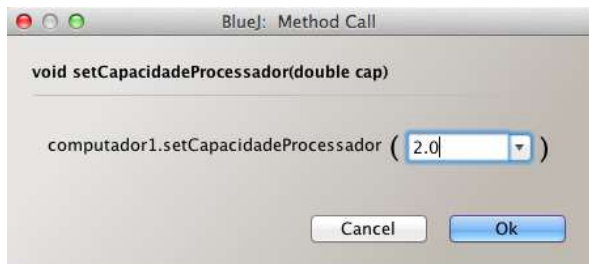
Após informar os valores das propriedades do objeto, vamos clicar no botão: [OK] para que o objeto seja criado.



Com o objeto criado em memória, podemos então clicar com o botão direito sobre o objeto e executar qualquer um dos métodos que são apresentados. Os métodos apresentados são os métodos que criamos neste exemplo.

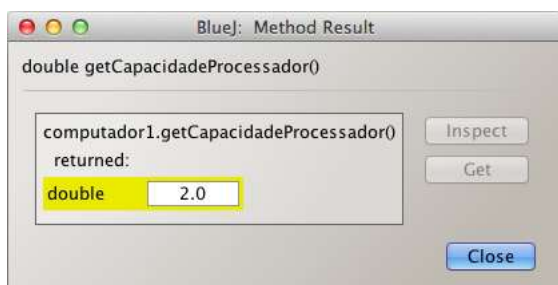


Vamos então selecionar o método que permite a alteração do valor do atributo: capacidadeProcessador. Quando criamos o nosso objeto, informamos o valor: 1.5 para a capacidadeProcessador. Vamos alterar este valor para: 2.0, conforme ilustra a figura abaixo:



Após informarmos o novo valor e clicarmos no botão: [OK], este novo valor será atribuído para o atributo: capacidadeProcessador do objeto criado. Para confirmarmos esta alteração, podemos utilizar o método de acesso ao atributo: capacidadeProcessador, ou mesmo a opção: Inspect como já utilizamos no primeiro exemplo.

Validando a alteração com o método de acesso ao atributo:



Validando a alteração com a opção: Inspect do BlueJ:

computador1 : Computador

private String modeloProcessador	Intel Celeron	Inspect
private double capacidadeProcessador	2.0	Get
private int quantidadeMemoriaRAM	512	
private int tamanhoHD	80	

Show static fields Close