

MÓDULO 06 – PRÁTICA DE LABORATÓRIO I

1 INTRODUÇÃO

Nos tópicos a seguir **estaremos praticando os conceitos visto no módulo 6 de Programação I**. Revise estes conceitos nos seguintes capítulos do livro de apoio de Programação I:

- Capítulo 06 – CLASSE DE TESTE.

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 06.

1.1 CLASSE DE TESTE

No capítulo 06 do livro de apoio de Programação I, vimos que podemos utilizar classes de teste para criar nossos objetos e também para executar outros tipos de comandos Java. Inclusive, nos módulos anteriores de Laboratório I, todos os exercícios e exemplos que utilizamos foi com base neste tipo de classe.

A classe de teste, por sua vez, é utilizada principalmente para testar as classes que criam objetos. Uma classe de teste não costuma ter atributos e pode ser constituída com um único método, de nome **main**, que tem a seguinte assinatura:

```
public static void main (String args[ ] )
```

O corpo do método **main** deve conter comandos para instanciar objetos da classe que estamos testando e comandos para chamar os métodos da classe sendo testada para verificar se eles funcionam. Além disso, podemos também utilizar a classe de teste para executar comandos Java, como os comandos vistos nos módulos anteriores de Programação e Laboratório I.

1.2 EXEMPLO UTILIZANDO CLASSE DE TESTE PARA CRIAR OBJETOS

No módulos anteriores de Laboratório I, criamos os objetos por meio das opções do BlueJ. Neste novo exemplo, vamos praticar a criação dos objetos por meio de uma classe de teste. Para isso, vamos criar a classe de teste abaixo e utilizar a nossa classe Computador para a criação dos objetos:

```
//Classe de Teste
public class CriaComputador
{
    //método main
    public static void main(String args[])
    {
        //criando o primeiro objeto: computador 1
        Computador computador1 = new Computador("Intel Celeron",1.5,520,80);

        //criando o segundo objeto: computador 2
        Computador computador2 = new Computador("Intel Pentium 4",1.5,256,40);

        //criando o terceiro objeto: computador 3
        Computador computador3 = new Computador("Intel Core Duo",1.6,2,120);

        //acessando o valor do atributo: capacidadeProcessador por meio do método de acesso do atributo
        System.out.println("Capacidade do Computador 1: " + computador1.getCapacidadeProcessador());
        System.out.println("Capacidade do Computador 2: " + computador2.getCapacidadeProcessador());
        System.out.println("Capacidade do Computador 3: " + computador3.getCapacidadeProcessador());

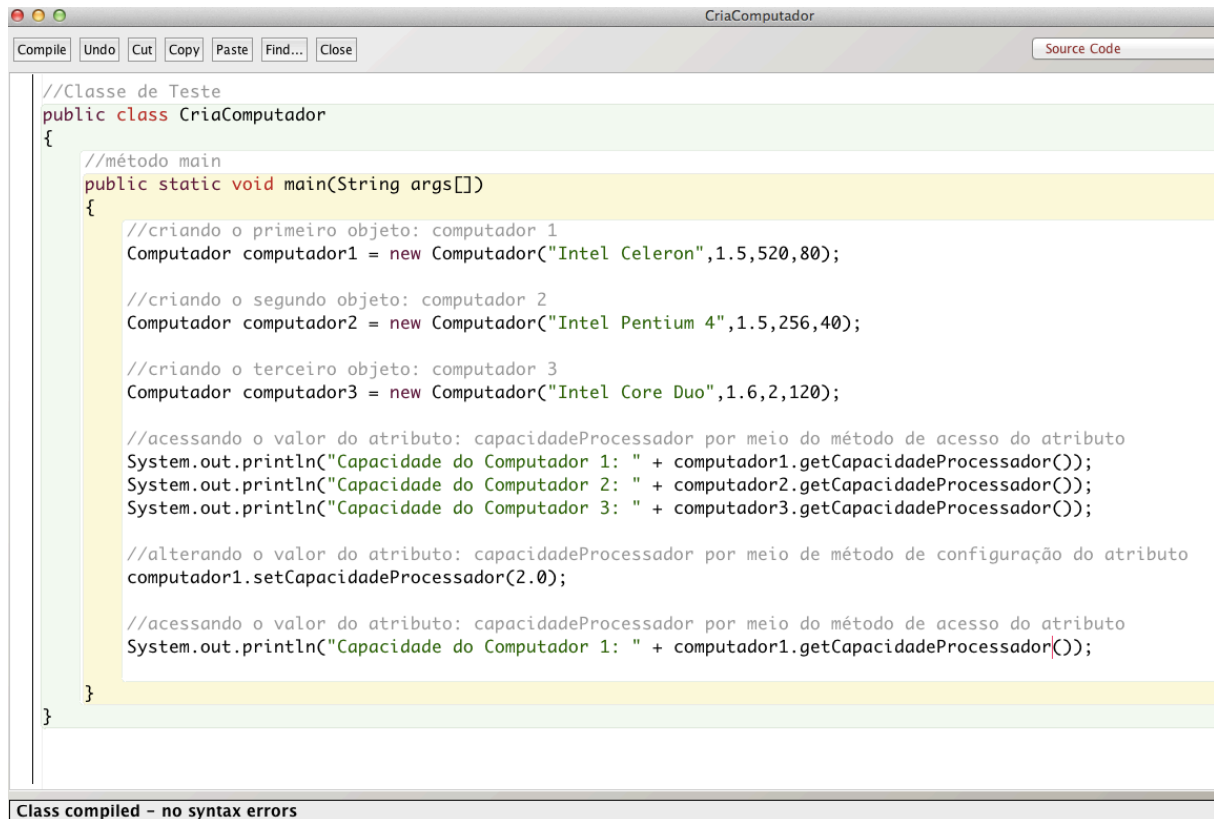
        //alterando o valor do atributo: capacidadeProcessador por meio de método de configuração do atributo
        computador1.setCapacidadeProcessador(2.0);

        //acessando o valor do atributo: capacidadeProcessador por meio do método de acesso do atributo
        System.out.println("Capacidade do Computador 1: " + computador1.getCapacidadeProcessador());
    }
}
```

Na classe de teste apresentada acima estamos: 1) criando os três objetos Computador, 2) acessando os valores do atributo: capacidadeProcessador por meio dos métodos de acesso e 3) utilizando um comando de saída para apresentar na tela.

Por fim, utilizamos um método de configuração para alterar o valor do atributo: capacidadeProcessador do objeto: **computador1** e, depois da alteração, apresentamos novamente o valor na tela por meio do comando de saída de dados na tela.

Para executarmos a classe de teste, basta: 1) compilar a classe; 2) clicar com o botão direito em cima da classe e 3) selecionar a opção para a execução do método main, como já fizemos na prática de Laboratório do módulo 01 e como ilustra nas figuras abaixo:



```
//Classe de Teste
public class CriaComputador
{
    //método main
    public static void main(String args[])
    {
        //criando o primeiro objeto: computador 1
        Computador computador1 = new Computador("Intel Celeron",1.5,520,80);

        //criando o segundo objeto: computador 2
        Computador computador2 = new Computador("Intel Pentium 4",1.5,256,40);

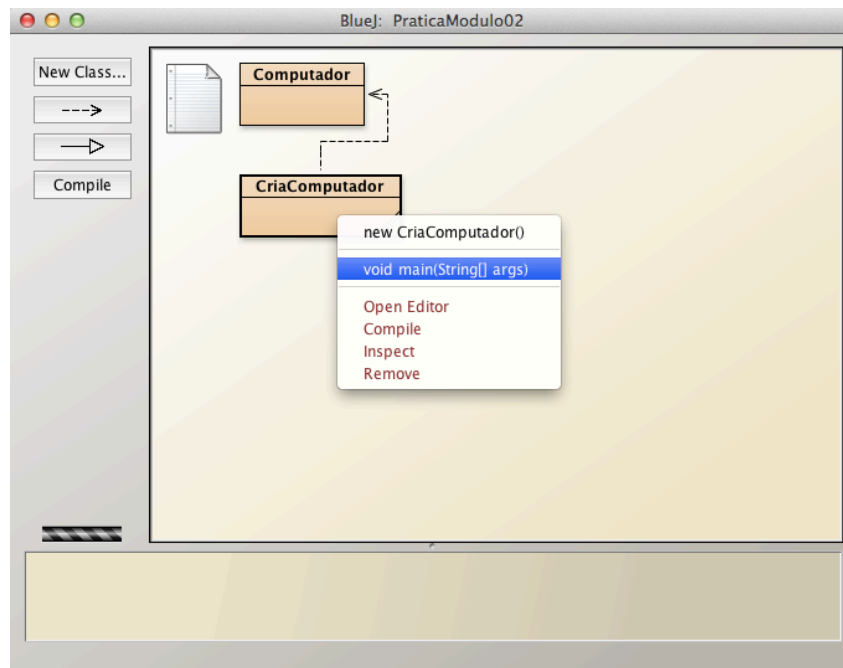
        //criando o terceiro objeto: computador 3
        Computador computador3 = new Computador("Intel Core Duo",1.6,2,120);

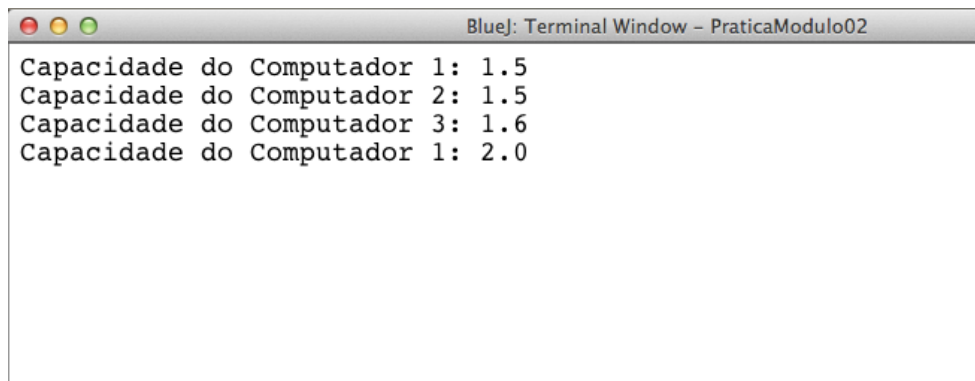
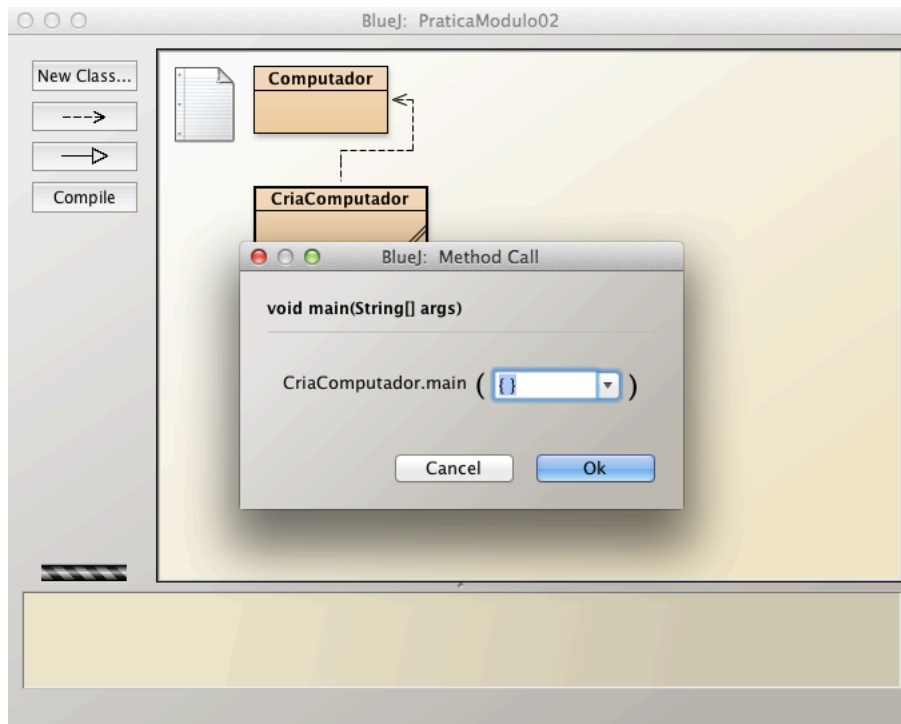
        //acessando o valor do atributo: capacidadeProcessador por meio do método de acesso do atributo
        System.out.println("Capacidade do Computador 1: " + computador1.getCapacidadeProcessador());
        System.out.println("Capacidade do Computador 2: " + computador2.getCapacidadeProcessador());
        System.out.println("Capacidade do Computador 3: " + computador3.getCapacidadeProcessador());

        //alterando o valor do atributo: capacidadeProcessador por meio de método de configuração do atributo
        computador1.setCapacidadeProcessador(2.0);

        //acessando o valor do atributo: capacidadeProcessador por meio do método de acesso do atributo
        System.out.println("Capacidade do Computador 1: " + computador1.getCapacidadeProcessador());
    }
}
```

Class compiled - no syntax errors





The screenshot shows the "BlueJ: Terminal Window - PraticaModulo02". The terminal displays the following output:

```
Capacidade do Computador 1: 1.5  
Capacidade do Computador 2: 1.5  
Capacidade do Computador 3: 1.6  
Capacidade do Computador 1: 2.0
```