



**Universidade Federal do Ceará**  
**Campus de Sobral**  
**Curso de Engenharia da Computação**  
**Técnicas de Programação – TD03 15/04/2016**  
**Prof. Iális Cavalcante (ialis@sobral.ufc.br)**

**Prazo de Entrega: 17/04/2016**

Inicie um novo projeto chamado de **TD03**. Crie o pacote de nome **td03** (se não surgir automaticamente) e neste mesmo adicione duas classes nomeadas como: **TestaFormas** (que será a classe principal) e **Ponto**.

Na classe **Ponto**, defina duas variáveis *x* e *y* correspondendo às coordenadas de um ponto real. Crie dois métodos construtores:

- um de comportamento padrão (sem argumentos), no qual os atributos recebem o valor zero;
- outro que tenha dois argumentos, e que estes mesmos atribuam os novos valores de *x* e *y*;

ainda em **Ponto**, defina um chamado de **descreveForma** o qual retorna um valor String. Este valor será atribuído por uma variável interna a este método que recebe o seguinte valor: "Coordenadas do Ponto: (" +String.valueOf(x)+" , "+String.valueOf(y)+" )";

Partindo agora para a classe principal **TestaFormas**, insira dentro do método principal os seguintes comandos:

- crie um vetor de nome **formas** e de tipo **Ponto**, com três elementos na sua definição;
- ao primeiro elemento do vetor, instancie um objeto **Ponto** a partir do construtor padrão;
- ao segundo elemento do vetor, instancie um objeto **Ponto** a partir do construtor padrão;
- ainda no segundo elemento do vetor, modifique o atributo *x* para assumir o valor 5;
- neste mesmo elemento do vetor, modifique o atributo *y* para assumir o valor 15;
- ao terceiro elemento do vetor, instancie um objeto **Ponto** a partir do outro construtor e inserindo os valores 20 e 10 nos argumentos do mesmo.

Nesta mesma classe, crie um *loop* para percorrer todo o vetor **formas** afim de que em cada elemento, descreva a forma do mesmo (a partir do método **descreveForma**).

Defina uma nova classe chamada **Circulo**, que vai herdar as características de **Ponto** (utilize a palavra-chave **extends**). Defina uma variável *raio* correspondendo a um raio real. Crie dois métodos construtores:

- um de comportamento padrão (sem argumentos), no qual o atributo *raio* recebe o valor zero e invoca o construtor padrão de **Ponto**: *super(0,0)*;
- outro que tenha três argumentos, e que estes mesmos atribuam o novo valor de *raio* e invoca o outro construtor de **Ponto**: *super(argumento1,argumento2)*;

ainda em **Circulo**, defina um chamado de **descreveForma** o qual retorna um valor String. Este valor será atribuído por uma variável interna a este método que recebe o seguinte valor: "Raio do Circulo: " +String.valueOf(raio)+" , e " +super.descreveForma();

Voltando para a classe principal **TestaFormas**, insira dentro do método principal (logo após as instruções definidas anteriormente) os seguintes comandos: crie um vetor de nome **formas2** e de tipo **Circulo**, com três elementos na sua definição;

- ao primeiro elemento do vetor, instancie um objeto **Circulo** a partir do construtor padrão;
- ao segundo elemento do vetor, instancie um objeto **Circulo** a partir do construtor padrão;
- ainda no segundo elemento do vetor, modifique o atributo *x* para assumir o valor 5;
- neste mesmo elemento do vetor, modifique o atributo *y* para assumir o valor 15;
- neste mesmo elemento do vetor, modifique o atributo *raio* para assumir o valor 15;
- ao terceiro elemento do vetor, instancie um objeto **Circulo** a partir do outro construtor e inserindo os valores 30, 20 e 10 nos argumentos do mesmo.

Nesta mesma classe, crie um *loop* para percorrer todo o vetor **formas2** afim de que em cada elemento, descreva a forma do mesmo (a partir do método **descreveForma**).

## Questão 1

### Aplicar Encapsulamento nas Classes

Aplique a técnica de encapsulamento nas classes **Ponto** e **Circulo**: determine que os atributos destas sejam privados e os demais métodos sejam públicos (inclusive os novos métodos de acesso a esses atributos encapsulados). Verifique os erros que surgem na classe **TestaFormas** e corrija-os de acordo com o encapsulamento das classes anteriores.

## Questão 2

### Exercitar relacionamento de Herança entre as Classes

Seguindo o raciocínio das classes **Ponto** e **Circulo**, defina uma nova classe chamada **Cilindro** que vai herdar as características de **Circulo**. Defina um atributo *altura* encapsulado correspondendo a uma altura real. Crie dois métodos construtores nesta mesma classe:

- um de comportamento padrão (sem argumentos), no qual o atributo *altura* recebe o valor zero e invoca o construtor padrão de **Circulo**: `super(0,0,0);`
- outro que tenha quatro argumentos, e que estes mesmos atribuam o novo valor de *altura* e invoca o outro construtor de **Circulo**: `super(argumento1,argumento2,argumento3);`

ainda em **Cilindro**, defina um chamado de **descreveForma** o qual retorna um valor String. Este valor será atribuído por uma variável interna a este método que recebe o seguinte valor: "Altura do Cilindro: "+String.valueOf(getAltura())+", e "+super.descreveForma();

Volte para a classe principal **TestaFormas**, insira dentro do método principal (logo após as instruções definidas anteriormente) os seguintes comandos:

- crie um vetor de nome **formas3** e de tipo **Ponto**, com cinco elementos na sua definição;
- ao primeiro elemento do vetor, instancie um objeto **Ponto** a partir do construtor de dois argumentos: 10 e 10;
- ao segundo elemento do vetor, instancie um objeto **Circulo** a partir do construtor de três argumentos: 5, 20 e 20;
- ao terceiro elemento do vetor, instancie um objeto **Cilindro** a partir do construtor padrão;

- ao quarto elemento do vetor, instancie um objeto **Cilindro** a partir do construtor de quatro argumentos: 6, 9, 10 e 12;
- ao quinto elemento do vetor, instancie um objeto **Ponto** a partir do construtor padrão.

Nesta mesma classe, crie um *loop* para percorrer todo o vetor **formas3** afim de que em cada elemento, descreva a forma do mesmo (a partir do método **descreveForma**). Verifique o resultado final e tente compreender a influência da técnica de herança neste resultado.

---

**Nota:** Esta lista deve ser entregue via SIGAA com todo o projeto criado compactado identificando os nomes dos alunos.

Bom Trabalho!

---