Lista de exercícios 2

Programação Orientada à Objetos

Todos exercícios deverão ser entregues no Moodle, em um único arquivo comprimido (.rar de preferência). Os códigos deverão seguir o estilo de código da linguagem Java usado pelo Google (ver link no Moodle) e deverão estar comentados.

1) Continue a desenvolver usando o exemplo base da aula. Inclua nas classes Retangulo e Triangulo métodos para calcular o perímetro dessas formas. Teste.

2) Crie um construtor para a classe Retangulo, similar ao construtor da classe Triangulo. Teste.

3) Crie uma classe Circulo, similar às outras formas (defina um construtor também, método de área e perímetro). Teste.

4) Na classe Comparador, crie um método comparaPerimetro, similar ao método existente que compara áreas. Além disso crie um comparaArea e comparaPerimetro para cada par de tipos (ex.: maior1(retangulo, triangulo), maior2(retangulo, circulo), maior3(triangulo, circulo)…).

5) Crie uma classe Somador seguindo a definição abaixo:

Somador
+area: double
+adicionarTriangulo(tri: Triangulo) +adicionarRetangulo(ret: Retangulo) +adicionarCirculo(circ: Circulo)

Essa classe deverá acumular as áreas das formas que forem adicionadas à ela. Para testar, adicione alguns objetos ao somador, e depois imprima a área e confira.

6) Garanta que todas as classes que não tem main estão em arquivos separados.

7) Crie uma classe Folha, que representa uma folha de papel aonde o usuário poderá “pintar” formas (como no MSPaint). A classe deverá ter os seguintes atributos e métodos (a figura abaixo usa o padrão UML):

Folha
+altura: double +largura: double +listaTriangulos: Triangulo[] +listaRetangulos: Retangulo[] +listaCirculos: Circulo[] +numeroTriangulos: int +numeroRetangulos: int +numeroCirculos: int

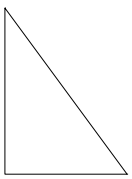
+adicionarTriangulo(tri: Triangulo) +adicionarRetangulo(ret: Retangulo) +adicionarCirculo(circ: Circulo) +listarFormas()

Os atributos listaTriangulos, listaRetangulos e listaCirculos são listas dos respectivos objetos. Devem ser declarados (ex.: Triangulo[] tris;) como atributos normais (dentro do corpo da classe), mas devem ser alocados (reserva de memória) no construtor da classe (ou seja, dentro do construtor devemos ter: tris = new Triangulo[10]). Aloque arrays de 10 elementos no máximo. Os métodos adicionar* deverão receber um triangulo criado no corpo principal (main) e deverão ser adicionados as listas da Folha. O método listarFormas deverá imprimir na tela quantas formas foram adicionadas no total, quantas foram adicionadas por tipo, e imprimir as propriedades de cada elemento de cada lista. Fique a vontade para adicionar outros atributos e métodos que acredite necessários.

8) Adicione um atributo “z” em todas as formas, que indica sua posição na folha. Z cresce a partir de 0, sendo 0 o primeiro elemento desenhado na folha, 1 o segundo e assim por diante. Esse atributo deverá ser sobreescrito pela classe folha, dentro dos métodos adicionar*. Ou seja, se o usuário desenhou um círculo, o método adicionarCirculo deverá mudar o atributo z desse objeto círculo para ser igual à 0 (além de adicionar o objeto na listaCirculo). Depois, se o usuário adicionou um triângulo, esse triângulo deverá ter seu atributo z sobreescrito com o valor 1, e assim por diante. Por fim, modifique o método listarFormas para mostrar o atributo z também. Fique a vontade para adicionar outros atributos e métodos que acredite necessários.

9) Adicione um método listarOrdenado() que lista as formas de acordo com o seu atributo z, de forma crescente.

10) Adicione a todas as formas atributos que indentificam sua posição na folha (através de x e y). Para os retângulos, esse valor indica aonde o canto inferior esquerdo estará na folha. Para o círculo, onde seu centro estará, e para o triângulo onde seu ângulo de 90 graus está. Suponha que triângulos sejam sempre desenhados na seguinte orientação:



Teste. Ao adicionar as formas na Folha, imprima uma mensagem de aviso informando o usuário caso a forma, ou alguma parte dela esteja saindo da folha.

11) Comente as classes e os métodos das classes criadas, seguindo o exemplo da classe Comparador. Depois vá em Run→ Generate Javadoc para verificar sua documentação.