Segunda Prova de Programação Orientada a Objetos

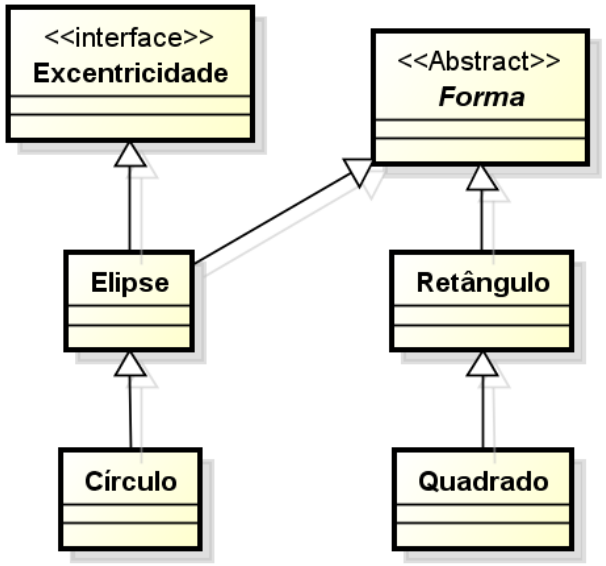
Data: 16/09/2022

Nome:_____ Nota: _____

Observações: Leiam a prova TODA com calma e atenção!

Lembrem dos conceitos de Orientação à Objetos

Boa Prova!



Você é o responsável por criar a modelagem de figuras geométricas e seus respectivos testes para o sistema.

MODELAGEM DO SISTEMA

Considerando o diagrama de classes acima como base da implementação, implemente as classes presentes. Os detalhes de cada uma serão apresentado a seguir.

Como podemos ver todas as figuras são filhas de Forma, e não irão existir instâncias de Forma. Toda Forma sabe calcular sua área, seu perímetro e ter um método para imprirmir esses valores.

Um Retângulo tem duas características, as quais são a **largura** e **comprimento**. Desta forma, sua área se dá pela multiplicação dos dois e, seu perímetro, pela soma destes multiplicado por 2. Um Quadrado é um caso particular de Retângulo onde a largura e comprimento são iguais.

Uma Elipse possui duas características, um eixo maior chamado ***a*** e um eixo menor chamado ***b***, como visto na Figura 1 abaixo.

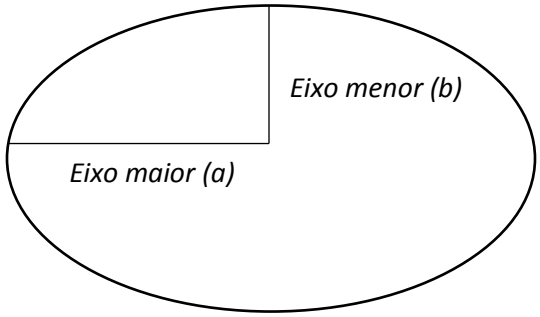


Figure 1: Elipse

No caso de uma Elipse o eixo ***a > b***, e caso ***a = b* este se torna um Círculo**! A medida do quão redonda é uma Elipse é dada pela sua Excentricidade *E*, onde os valores estão sempre entre 0 e 1. Por exemplo, para um Círculo *E = 0*.

Vamos às fórmulas da Elipse:

Perímetro: $P = \pi\sqrt{2(a^2 + b^2) - (a - b)^2/2}$ (Note, que se a = b = r, então $P = 2\pi r$)

Área: $A = \pi AB$

Excentricidade: $E = \sqrt{1 - b^2/a^2}$

Lembrando, **todas as formas que possuem Excentricidade devem saber calculá-la**!

TESTES

Após criar as classes acima demonstradas pelo diagrama de classe e suas definições, crie uma classe chama **TestaFormas**, a qual possui um **método estático** chamado **criaFormas** que retorna um vetor de formas (ou ArrayList, o que você achar mais fácil). Dentro deste método você irá preencher o vetor (ou ArrayList) usando uma estrutura de **switch** (como podem ver na modelagem acima temos 4 formas que podem ser criadas). Use o *Random* para definir quais formas serão adicionadas ao vetor e também para preencher os atributos de cada forma! (Acredite é mais fácil assim).

Crie uma nova classe chamada **TesteDeDownCasting** para testar a criação de instâncias. Nesta classe, **onde você colocará seu método main**, você deve criar um vetor (ou ArrayList) de Formas e utilizar o método estático definido anteriormente para preenchê-lo. Com o vetor(ou ArrayList) preenchido, você deverá percorrer e imprimir os **atributos** de cada uma das Formas do vetor(ou ArrayList) – Dica os **gets** são indispensáveis para que isso seja realizado!

Lembrete: Random rand = new Random();

rand.nextDouble(valorMaximo);