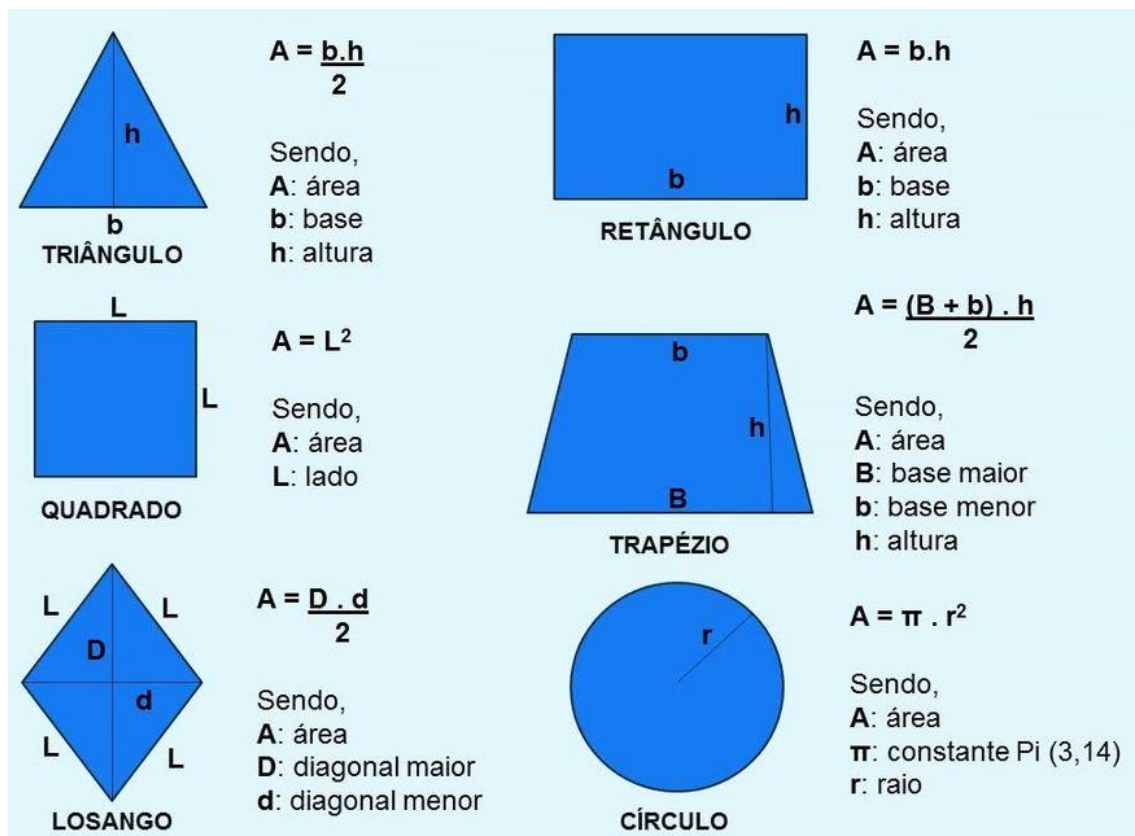


Instruções

Nessa atividade teremos a oportunidade de exercitar um pouco do que estamos estudando na Unidade 3.

Vamos aplicar, na prática, os conceitos de herança e polimorfismo, usando para isso figuras geométricas.

Observe a figura abaixo, que mostra como calcular a área de algumas dessas figuras.



Tendo como base essa figura, execute agora as tarefas a seguir.

Tarefa 1) Escreva o código de uma classe Java denominada "FiguraGeometrica". Essa classe:

- deve ser declarada como "abstract"
- deve ter um atributo "nome" para guardar o nome da figura geométrica
- deve um método "getArea", que retorna um valor do tipo "double" correspondente à área da figura geométrica. Esse método deverá ser declarado como "abstract", porque iremos

sobrescrevê-lo nas subclasses de "FiguraGeometrica".

- um método "getDescricao", que retorna um valor do tipo "String" correspondente à descrição da figura geométrica. Esse método deverá ser declarado como "abstract", porque iremos sobrescrevê-lo nas subclasses de "FiguraGeometrica".

Você deve postar, para essa tarefa, o código da sua classe "FiguraGeometrica". (0.3 pontos)

Importante: O atributo "nome" deve ser declarado como "private", e deve existir na classe métodos get/set declarados como "public" para acessar esse atributo.

Tarefa 2) Escreva o código de uma classe Java denominada "Retangulo". Essa classe:

- deve ser filha de "FiguraGeometrica"
- deve ter os atributos "base" e "altura" para guardar a base e a altura do retângulo
- deve ter um método "getArea", que sobrepõe o método definido na classe "FiguraGeometrica", e que retorna a área do retângulo.
- deve ter um método "getDescricao", que sobrepõe o método definido na classe "FiguraGeometrica", e que retorna um texto que mostra os valores da base e da altura do retângulo.

Você deve postar, para essa tarefa, o código da sua classe "Retangulo". (0.3 pontos)

Importante: Os atributos da classe devem ser declarados como "private", e deve existir na classe métodos get/set declarados como "public" para acessar os atributos.

Tarefa 3) Escreva o código de uma classe Java denominada "Triangulo". Essa classe:

- deve ser filha de "FiguraGeometrica"
- deve ter os atributos "base" e "altura" para guardar a base e a altura do triângulo
- deve ter um método "getArea", que sobrepõe o método definido na classe "FiguraGeometrica", e que retorna a área do

triângulo.

- deve ter um método "getDescricao", que sobrepõe o método definido na classe "FiguraGeometrica", e que retorna um texto que mostra os valores da base e da altura do triângulo.

Você deve postar, para essa tarefa, o código da sua classe "Triangulo". (0.3 pontos)

Importante: Os atributos da classe devem ser declarados como "private", e deve existir na classe métodos get/set declarados como "public" para acessar os atributos.

Tarefa 4) Escreva o código de uma classe Java denominada "Circulo". Essa classe:

- deve ser filha de "FiguraGeometrica"
- deve ter um atributo "raio" para guardar o raio do círculo
- deve ter um método "getArea", que sobrepõe o método definido na classe "FiguraGeometrica", e que retorna a área do círculo.
- deve ter um método "getDescricao", que sobrepõe o método definido na classe "FiguraGeometrica", e que retorna um texto que mostra o valor do raio do círculo.

Você deve postar, para essa tarefa, o código da sua classe "Circulo". (0.3 pontos)

Importante: Os atributos da classe devem ser declarados como "private", e deve existir na classe métodos get/set declarados como "public" para acessar os atributos.

Tarefa 5) Para essa tarefa, escreva uma classe contendo o método "main". Nesse método:

- adicione comandos para criar objetos do tipo "Retangulo", "Triangulo" e "Circulo" (um objeto de cada tipo).
- adicione comandos para definir valores para os atributos desses objetos, usando os métodos "set" correspondentes.
- adicione o trecho de código abaixo, para adicionar os objetos criados em um ArrayList:

```
// Adiciona as figuras geometricas em uma lista
List<FiguraGeometrica> listaFigurasGeometricas = new
ArrayList<>();
listaFigurasGeometricas.add( retangulo );
listaFigurasGeometricas.add( triangulo );
listaFigurasGeometricas.add( circulo );
```

- adicione o trecho de código abaixo, para mostrar a descrição e a área das figuras que você adicionou na lista:

```
// Mostra dados das figuras geometricas
for(FiguraGeometrica figuraGeometrica :
listaFigurasGeometricas) {
    System.out.println( figuraGeometrica.getDescricao() );
    System.out.printf( "Area da figura: %.2f \n",
figuraGeometrica.getArea() );
}
```

Você deve postar, para essa tarefa, o código do método "main" e também as mensagens que foram impressas no loop "for". (0.3 pontos)

Comentário. Você observou, no loop da Tarefa 5, que estamos tratando nossos objetos do tipo "Retangulo", "Triangulo" e "Circulo" como objetos do tipo "FiguraGeometrica"? Isso é o polimorfismo, na prática!

IMPORTANTE:

- Responda os itens de 1 a 5 postando UM ARQUIVO na plataforma.
- Deixe claro, na sua postagem, a resposta de cada item. Siga o seguinte modelo:

```
Tarefa 1) xxxxx
Tarefa 2) yyyyy
...
```

- Você não precisa repetir todo seu código na resposta de cada item. Responda apenas aquilo que foi solicitado. Por exemplo,

para o item 5 você não precisa postar novamente o código das classes "FiguraGeometrica", "Retangulo", "Triangulo" e "Circulo", porque você já fez isso no item 1. Poste, nesse item, somente o código do método "main".

- Tente ser breve e objetivo na sua resposta (por favor!)

- Capriche nas suas respostas, pois a postagem será avaliada **UMA ÚNICA VEZ!**

- E não se esqueça, esse trabalho é individual. Se sua resposta estiver idêntica (ou muito parecida) com a resposta de algum colega ela não receberá nota.

Então agora.... Mãos à obra!

E bom trabalho!

Ps. Gostou desse trabalho? Se sim, deixo aqui para você o desafio de criar as classes correspondentes às figuras geométricas que não foram implementadas na atividade.