

```

20 String toString () { return "Circulo_de_raio_" + raio; }
21 };

```

(a) (1v) Explícite a relação entre as 3 classes mencionadas, i.e. herança, características das classes, etc. Para isso, desenhe um diagrama com estas classes.

(b) (2v) Defina uma classe Retangulo com o posicionamento nesta hierarquia e comportamento esperados.

O construtor a definir deverá aceitar dois parâmetros do tipo double: um para a largura e outro para a altura. Inclua definições para os métodos `area()`, `perimetro()` e `toString()` apropriados para um retângulo.

(c) (1v) Se quisesse definir o Quadrado em termos do Retangulo, o que é que faria?

(d) (1v) Considere que `f` é uma Coleção de formas, e que pretendemos saber qual delas a que tem a maior área. Para isso definimos um método `maiorForma()`. Este método poderá recorrer a todas mensagens que se podem enviar a objetos da classe `Forma`.

Sabe-se que a classe de `f` permite aceder a todas as formas que integram a coleção, recorrendo ao método `nextForma()`, que retorna cada uma das formas, em chamadas sucessivas.

Pretendemos que `maiorForma()` retorne a `Forma` do seu conjunto que tiver **maior área**, por exemplo, se tivermos:

```

1  Coleção f;
2  Forma maior;
3  f.acrescenta (new Quadrado (3.0)); // quadrado de lado 3
4  f.acrescenta (new Circulo (2.0)); // circulo de raio 2
5  f.acrescenta (new Retangulo (1.5, 2.5)); // retangulo de 1.5 x 2.5
6  maior = f.maiorForma ();
7  System.out.println ("A_maior_forma_é_" + maior);

```

Diga qual será o output deste troço de código. Assuma que as declarações de classe estão todas feitas e que o método `toString()` foi definido para as subclasses de `Forma`.

(e) (2v) Complete a definição da função `maiorForma`.

```

1  class Coleção {
2      ...;
3      Forma nextForma ();
4
5      Forma maiorForma () {
6          Forma aForma;
7          ...; // RESPONDA AQUI
8          return aForma;
9      }
10 }

```