

## Aula prática #3

**Revisões de Java:**

1. Defina a classe **Shape**. Uma "shape" representa um figura geométrica qualquer. A única coisa que sabemos sobre estas figuras é têm uma área.
2. Defina a classe **Circle**. Esta classe implementa as figuras geométricas, círculo.
3. Defina a classe **Square**. Esta classe implementa a forma geométrica quadrado.
4. As formas geométricas são objectos comparáveis. Proceda às alterações necessárias na classe **Shape** para que tal se verifique.
5. Defina na classe **Shape** o método `toString()`, este método deve retornar uma mensagem do tipo "I Am a ??? my area is: ???". Por exemplo de `c1` for um círculo de raio `.21`, a mensagem do `toString()` deverá ser `I Am a class Circle my area is: 0.13854423602330987`
6. Defina uma classe **Teste**. Nesta classe crie Círculos, Quadrados, arrays de Shapes, arrays de Círculos, e percorra os arrays listando a representação em String dos objectos do array. Em concreto defina: `Shape[] arr1`, `Circle[] arr2`, e `Square[] arr3`, adicione a estes array objectos de tipo apropriado.
7. Defina o método estático `totalShape` que recebe um array de Shape e calcula a área total das figuras do array.
  - 7.1. Use o método `totalShape` para calcular a área total das figuras que constam nos arrays definidos em 6.
  - 7.2. faça a afectação de `arr2` a `arr1`, p.e. e veja os erros de compilação/e ou de execução.
  - 7.3. CONVENÇA-SE da **COVARIÂNCIA** DOS ARRAYS.
8. Defina o método estático `totalShape2` que recebe um colecção de Shape e calcula a área total das figuras da colecção.
  - 8.1. Defina 3 colecções: `Collection<Shape> col1`, `Collection<Circle> col2`, `Collection<Square> col3`
  - 8.2. Aplique o método `totalShape2` às colecções da alínea anterior.
  - 8.3. CONVENÇA-SE da **INVARIÂNCIA** DAS COLECÇÕES.
  - 8.4. Usando wildcards adapte o método `totalShape2`, para aceitar colecções de Shape ou colecções de qualquer de qualquer subclasse de Shape

Aula prática **#3**

9. Na classe de *Teste* defina o método *maior* que recebe um array de Comparables e calcula o maior elemento do array

- 9.1. Teste o método