PT.UA.DETI.POO.A8

NMEC ..... Nome .....

1. Para cada uma das declarações seguintes, assinale com um círculo se é Verdadeira (V) ou Falsa (F):

   V / F Numa classe abstrata podemos definir métodos sem corpo (sem código).
   Por exemplo: `public abstract double area();`
   em vez de: `public double area() { /* corpo */ }`

   V / F Se *Forma* for uma classe abstrata podemos criar referências do tipo *Forma*.
   Por exemplo: `Forma f1;`

   V / F Se *Forma* for uma classe abstrata podemos criar objetos do tipo *Forma*.
   Por exemplo: `Forma f1 = new Forma();`

   V / F Se a classe *Triângulo* estender a classe abstrata *Forma*, *Triângulo* tem de implementar os métodos abstratos de *Forma*, ou, em alternativa, ser abstrata.

   V / F Uma classe pode herdar de várias classes e implementar uma ou mais interfaces.

   V / F Se *Forma* for uma interface podemos criar referências do tipo *Forma*.
   Por exemplo: `Forma f1;`

   V / F Se *Forma* for uma interface podemos criar objetos do tipo *Forma*.
   Por exemplo: `Forma f1 = new Forma();`

   V / F Para além da declaração de assinaturas, podemos escrever métodos dentro de uma interface (definidas com o atributo *default*)

2. Considerando o código abaixo, indique se as condições seguintes dão um resultado *true* (V) ou *false* (F):

```
class Bull extends Animal {...}
class Dog extends Animal implements Pet {...}
Animal fido = new Dog();
Animal black = new Bull();
```

   V / F `if (black instanceof Animal) ..`

   V / F `if (black instanceof Pet) ..`

   V / F `if (fido instanceof Bull) ..`

   V / F `if (fido instanceof Animal) ..`

   V / F `if (fido instanceof Pet) ..`

3. Escreva a assinatura da função `paintSomeCreature` com um argumento único, de modo a que possa aceitar tanto um *Bull* como um *Dog* (entidades do problema 2).