



Disciplina:	POO	Prova:	1	Turma:	
Aluno(a):				Data:	
<ul style="list-style-type: none"><li>❖ Esta avaliação deve ser respondida usando caneta com tinta <b>azul</b> ou <b>preta</b>.</li><li>❖ Respostas a lápis são permitidas, mas implicam na impossibilidade do pedido de revisão de prova.</li><li>❖ SIGA SEMPRE OS PADRÕES APRESENTADOS EM AULA.</li></ul>					

1. O resultado da execução do programa Java a seguir é (0,3):

```
public class LengthArrayTest {  
    public static void main(String[] args) {  
        int arr[] = new int[2];  
        System.out.println(arr[0]);  
    }  
}
```

- (A) o programa não compila porque o valor de arr[0] foi utilizado antes de ter sido inicializado;  
(B) o programa gera uma exceção em tempo de execução porque o valor arr[0] foi utilizado antes de ter sido inicializado;  
(C) o programa compila e exibe 0 quando executado;  
(D) o programa compila e imprime 1 quando executado;  
(E) o programa compila e executa, mas os resultados são imprevisíveis porque arr[0] não foi inicializado.
2. Após a execução do trecho abaixo, o que será impresso no console? (0,5)

```
public class SomaMisteriosa {  
    public static void main(String[] args) {  
        int x=0;  
        int y[] = {0};  
        somaDois(x);  
        somaTres(y);  
        somaDois(y[0]);  
        System.out.println(x + " " + y[0]);  
    }  
    public static void somaTres( int x [] ){  
        x[0] +=3;  
    }  
    private static void somaDois(int x) {  
        x +=2;  
    }  
}
```

Resposta: \_\_\_\_\_

3. Resolva a questão abaixo (1,2):

Crie uma interface chamada Navegante com um método chamado navegar. O método navegar não retorna nada e possui um argumento do tipo float chamado velocidade.

Crie uma classe chamada Navio que implementa Navegante. A classe Navio possui um atributo de instância privado chamado velocidadeMaxima do tipo float. A classe Navio possui um método chamado ligar que não retorna nada, não tem argumentos e imprime no console "Navio navegando...". Crie um construtor para atribuir a velocidadeMaxima quando um novo construtor for instanciado, visto que **não** haverá getters e setters. Implemente todo o código de forma que não gere erro de compilação.

4. Observe os códigos abaixo e responda se as classes abaixo compilam. Caso sim, diga o que sai no console. Caso não, explique a razão. (1,5)

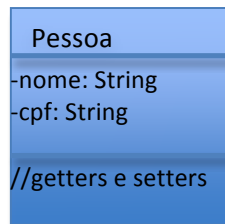
```
package doceria;
public class Rocambole extends Doce{
    private String nome;
    public String getNome() {
        return nome;
    }
    public void setNome(String nome) {
        this.nome = nome;
    }
}
```

```
package doceria;
public class Doce {
    public static int numeroDoces = 0;
    public Doce() {
        numeroDoces++;
    }
    public int getNumero() {
        return numeroDoces;
    }
}
```

```
package doceria;
public class Principal {
    public static void main(String[] args) {
        Rocambole d = new Rocambole();
        Rocambole d2 = new Rocambole();
        d.setNome("caramelo");
        d2.setNome("chocolate");
        teste(d, d2);
        int k = d.getNumero();
        System.out.println(k);
        k++;
        ++k;
        System.out.println(k);
        System.out.println(d.getNome());
        System.out.println(d2.getNome());
    }
    public static void teste(Rocambole d, Rocambole d2) {
        Rocambole d3 = d;
        d = d2;
        d2 = d3;
        d3.setNome("morango");
        d3 = new Rocambole();
        d2.setNome("pessego");
        d3.setNome("Abacate");
    }
}
```

Resposta: \_\_\_\_\_

5. Implemente o que for pedido.(3,0)  
Crie uma classe chamada Pessoa:



Crie uma classe principal. Defina um ArrayList e insira nesse arrayList dois objetos do tipo Pessoa. Preencha os atributos nome e CPF dos dois objetos. Sabendo que o método `indexOf(Object o)` de ArrayList retorna o índice de um objeto no ArrayList, implemente o método que fará com que ele possa ser utilizado da seguinte forma: se um objeto Pessoa tiver o cpf igual ao cpf de uma Pessoa dentro da lista, esse índice deve ser retornado.

Além disso, implemente o método que fará com que, ao imprimir a referência de um objeto do tipo Pessoa, seja impresso o nome e o cpf da pessoa, ao invés de `Pessoa@1432` ou algo semelhante.

Crie um novo objeto Pessoa com os mesmos valores de um objeto na lista. Imprima o índice desse objeto no console. Ex: "O objeto verificado possui o índice [x]". Por fim, imprima a referencia desse novo objeto.

6. Resolva a questão abaixo (3,5):

Crie uma classe abstrata chamada Figura que contenha um atributo privado do tipo String chamado "cor". Crie, nessa mesma classe, os getters e setters desse atributo. Em seguida, crie um método abstrato chamado `desenhar()` que não retorne nada e seja público. Crie um construtor com um argumento (cor) do tipo String que seja responsável por atribuir o valor recebido nesse argumento à variável de instância. Você não pode incluir o construtor sem argumentos nessa classe.

5.1 – Crie uma classe chamada Quadrado que seja filha de Figura. Implemente um construtor com um argumento que será a cor da figura. Este argumento será responsável por atribuir o valor recebido à variável de instância da classe pai. Implemente o construtor sem argumentos. Implemente o método `desenhar()` imprimindo no console: "Desenhando um quadrado...". Faça com que essa classe compile sem problemas.

5.2- Siga o mesmo molde do item 4.1 para criar uma classe chamada Pentagono. Além disso, crie o método `recortar()` (APENAS NA CLASSE PENTAGONO) que não retorne nada, seja público e que imprima no console: "Recortando um pentágono...". Não esqueça de implementar o método `desenhar()`.

5.3 – Crie uma classe chamada Principal que seja executável.

Pressuponha que sua classe Principal possui o seguinte método:

```
public static Figura retornaFiguraAleatoria() {
    int i = (int) (Math.random()*100);
    if (i % 2 == 0) {
        return new Quadrado ("azul");
    } else {
        return new Pentagono ("vermelho");
    }
}
```

Sua classe Principal deverá:

- Criar um array (dentro do método main) do tipo Figura com 1000 elementos.
- Utilizar o método `retornaFiguraAleatoria()` para popular as 1000 posições do array.
- Crie um método chamado `contarFiguras` que não retorne nada e irá receber como argumento um array de Figuras. Esse método irá contar quantos objetos existem do tipo Pentagono e quantos existem do tipo Quadrado e imprimir no console o total de Pentagonos e Quadrados. Faça a chamada a esse método. Ex: "Existem 300 pentágonos e 700 quadrados."
- Crie um método chamado `recortarSePossivel` que não retorne nada e receba como argumento um array de Figuras. Esse método irá percorrer os 1000 elementos do array e chamará o método `recortar()` caso o objeto seja uma instância de Pentagono. Faça a chamada ao método.
- Crie um método chamado `desenharTudo` que receba como argumento um array de Figura, não retorne nada. Esse método deverá chamar o método `desenhar` de cada Figura.
- Depois de todas as chamadas, a última instrução do seu método main, deverá ser imprimir no console o seguinte: "Fim do programa..."