

LPOO
1º semestre de 2022
Prova Surpresa
14/04/2023
Tempo limite: ≈75 Minutos

Nome: _____
RA: _____

Professores: Vinicius Pereira

Esta prova contém 7 páginas e 7 questões.

Para exibir algo na saída padrão você pode usar o termo `sout()`.

1. (1 ponto) Escreva o resultado do seguinte código

```
1 class Autor{
2     String nome;
3
4     public Autor(String aNome){
5         this.nome = aNome;
6     }
7
8     @Override
9     public String toString(){
10        String res = "Autor\n";
11        res += "Nome: " + this.nome + "\n";
12        return res;
13    }
14
15    public static void main(String[] args){
16        Autor autor = new Autor("GRRM");
17        System.out.println(autor);
18    }
19
20 }
```

2. (2 pontos) Escreva as seguintes classes

a) Uma classe chamada “Autor”, que contenha

- O atributo privado: “nome”, que seja uma *String*.
- Um constructor que receba uma *String* e inicialize o atributo “nome”.
- O método de acesso *get* e *set* para o atributo
- Override do método *toString*, que retorna uma string com os valores de todos os atributos.

b) Uma classe chamada “Livro”, que contenha

- os atributos privados: “nome”, que seja uma *String* e “autor”, que seja um *Autor*.
- Um constructor que receba o *nome do livro*, e o *nome do autor*, e inicialize **todos** os atributos
- Os métodos de acesso *public String getNomeDoLivro()* que retorna o nome do livro e *public String getNomeDoAutor()* que retorna o nome do autor. (não precisa ter os *setters*)
- Override do método *toString*, que retorna uma string com os valores de todos os atributos.

3. (2 pontos) Escreva uma classe chamada “Jogador”, que contenha
- os atributos privados: “nome”, que seja uma *String* e “pontuacao”, que seja um *double*.
 - Um constructor *public Jogador(String aNome, double aPontuacao)* que inicialize **todos** os atributos
 - Os métodos de acesso *getters* e *setters* para todos os atributos
 - Um método publico para aumentar a pontuação. Este método recebe um valor como parâmetro e aumenta a pontuação com este valor recebido. O método não retorna nada.
 - Override do método *toString*, que retorna uma string com os valores de todos os atributos.

Para as seguintes questões, considere as classes “Pato”, “PatoDonald”, e “PatoDeBorracha”:

```
1 class Pato {
2
3     public void fala(){
4         sout("quack");
5     }
6     public void nada(){
7         sout("nada no rio");
8     }
9     public void voa(){
10        sout("voa para o sul");
11    }
12 }
```

```
1 class PatoDonald extends Pato {
2
3     @Override
4     public void voa(){
5         sout("voa de aviao");
6     }
7 }
```

```
1 class PatoDeBorracha extends Pato {
2
3     @Override
4     public void nada(){
5         sout("boia na banheira");
6     }
7     @Override
8     public void voa(){
9         sout("cai");
10    }
11 }
```

4. (1 ponto) Verifique se o código a seguir será compilado. Se for, escreva o resultado do código, caso não seja, explique porque não irá compilar.
-

```
1 class TestePato {
2     public static void main(String[] args) {
3         Pato pato = new Pato();
4         pato.fala();
5         pato.nada();
6         pato.voa();
7     }
8 }
```

5. (1 ponto) Verifique se o código a seguir será compilado. Se for, escreva o resultado do código, caso não seja, explique porque não irá compilar.
-

```
1 class TestePatoDonald {
2     public static void main(String[] args) {
3         Pato pato = new PatoDonald();
4         pato.fala();
5         pato.nada();
6         pato.voa();
7     }
8 }
```

6. (1 ponto) Verifique se o código a seguir será compilado. Se for, escreva o resultado do código, caso não seja, explique porque não irá compilar.
-

```
1 class TestePatoDeBorracha {
2     public static void main(String[] args) {
3         PatoDeBorracha pato = new PatoDeBorracha();
4         pato.fala();
5         pato.nada();
6         pato.voa();
7     }
8 }
```

7. (2 pontos) Escreva o resultado do seguinte código:

```
1
2 class Carro{
3     private String modelo;
4     private int gas;
5     public Carro(String aModelo){
6         this.modelo = aModelo;
7         this.gas = 0;
8     }
9
10    public String getModelo(){
11        return this.modelo;
12    }
13
14    public int getGas(){
15        return this.gas;
16    }
17
18    public void abastece(int qtdGas){
19        this.gas += qtdGas;
20    }
21
22    @Override
23    public String toString(){
24        String res = "Carro:\n";
25        res += "Mod: " + this.modelo + "\n";
26        res += "Gas: " + this.gas + "\n";
27        return res;
28    }
29
30 }
```

```
1 import java.util.ArrayList;
2
3 class TesteCarro{
4
5     public static void abasteceGas(int gasDoCarro, int qtdGas){
6         gasDoCarro += qtdGas;
7     }
8     public static void abasteceCarro(Carro carro, int qtdGas){
9         carro.abastece(qtdGas);
10    }
11    public static void abasteceLista(ArrayList<Carro> carros, int qtdGas){
12        for(Carro carro: carros){
13            carro.abastece(qtdGas);
14        }
15    }
16
17    public static void main(String[] args){
18        Carro carro01 = new Carro("def");
19
20        System.out.println("1");
21        System.out.println(carro01);
22
23        System.out.println("2");
24        abasteceGas(carro01.getGas(), 2);
25        System.out.println(carro01);
26
27        System.out.println("3");
28        abasteceCarro(carro01, 2);
29        System.out.println(carro01);
30
31        System.out.println("4");
32        ArrayList<Carro> carros = new ArrayList<Carro>();
33        carros.add(carro01);
34        carros.add(new Carro("sec"));
35        abasteceLista(carros, 4);
36        for(Carro carro: carros){
37            System.out.println(carro);
38        }
39    }
40
41 }
```
