

**POO**  
**1º semestre**  
**Prova Antiga**  
**01/02/1234**

**Tempo limite: ≈75 Minutos**

**Nome:** \_\_\_\_\_  
**RA:** \_\_\_\_\_  
**Turma:** \_\_\_\_\_

Professores: Vinicius Pereira

---

Esta prova contém 8 páginas e 7 questões.

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

---

1. (1 ponto) Considere a seguinte classe

```
1 class Autor {  
2     private string nome;  
3     public Autor(string aNome) {  
4         this.nome = aNome;  
5     }  
6     public override string ToString() {  
7         return "Nome do Autor: " + this.nome + "\n";  
8     }  
9  
10 }
```

---

Escreva o resultado do seguinte código

```
1 class AutorListaTeste {  
2     public static void Main(string[] args) {  
3         List<Autor> autores = new List<Autor>();  
4         autores.Add(new Autor("Jane"));  
5         autores.Add(new Autor("Kurt"));  
6         autores.Add(new Autor("Wolf"));  
7  
8         foreach(Autor autor in autores) {  
9             Console.WriteLine(autor);  
10        }  
11    }  
12 }
```

---

---

Para as seguintes questões, considere as classes “Animal”, “Gato”, e “Cachorro”:

---

```
1 abstract class Animal {
2
3     private string nome;
4
5     public Animal(string aNome) {
6         this.nome = aNome;
7     }
8
9     public String GetNome() { return this.nome; }
10
11    virtual public void Dorme() {
12        Console.WriteLine("O animal " + this.nome + " dorme");
13    }
14
15    abstract public void FazBarulho();
16
17    public override string ToString() {
18        string res = "Animal\n";
19        res += "Nome: " + this.nome + "\n";
20        return res;
21    }
22
23 }
```

---

```
1 class Gato: Animal{
2     public Gato(string aNome): base(aNome) {
3     }
4     public override void FazBarulho() {
5         Console.WriteLine("O gato " + this.GetNome() + " mia");
6     }
7 }
```

---

```
1 class Cachorro: Animal {
2     public Cachorro(string aNome): base(aNome) {
3     }
4
5     public override void FazBarulho() {
6         Console.WriteLine("O cachorro " + this.GetNome() + " late");
7     }
8 }
```

---

2. (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 TesteAnimal {
2     static public void Main(string[] args) {
3         Animal animal = new Animal("Ani");
4         Console.WriteLine(animal);
5         animal.FazBarulho();
6     }
7 }
```

---

3. (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 TesteGato {
2     static public void Main(string[] args) {
3         Animal gato = new Gato("Chi");
4         Console.WriteLine(gato);
5         gato.FazBarulho();
6     }
7 }
```

---

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 TesteCachorro {
2     static public void Main(String[] args) {
3         Animal cachorro = new Cachorro("Brutus");
4         Console.WriteLine(cachorro);
5         cachorro.FazBarulho();
6     }
7 }
```

---

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 TesteLista {  
2     static public void Main(string[] args) {  
3         List<Animal> animais = new List<Animal>();  
4         animais.Add(new Gato("Mili"));  
5         animais.Add(new Cachorro("Max"));  
6  
7         foreach (Animal a in animais){  
8             System.Console.WriteLine(a);  
9             a.FazBarulho();  
10        }  
11    }  
12 }
```

6. (2 pontos) Escreva 3 classes:

Uma classe chamada `Funcionario` que tenha nome, e salário, e tenha o método `GetPagamento()` que retorne o salário.

Uma classe chamada `Gerente` que seja subclasse de `Funcionario`. Tenha o atributo `bonus` e sobrescreva o método `GetPagamento` fazendo com que retorne o salário mais o `bonus`.

Uma classe chamada `Horista` que seja subclasse de `Funcionario`. Tenha o atributo `horas` e sobrescreva o método `GetPagamento` fazendo com que retorne o salário multiplicado pelas horas.

7. (2 pontos) Considere a classe `Carro` e escreva o resultado do `TesteCarro`:

---

```
1
2 class Moto{
3     private string modelo;
4     private int gas;
5     public Moto(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    public override String ToString(){
23        return "Modelo: " + this.modelo + ", Gas: " + this.gas;
24    }
25
26 }
```

---

---

```
1 class TesteMoto{
2
3     public static void AbasteceGas(int gasMoto, int qtdGas){
4         gasMoto += qtdGas;
5     }
6     public static void AbasteceMoto(Moto moto, int qtdGas){
7         moto.Abastece(qtdGas);
8     }
9     public static void AbasteceLista(List<Moto> motos, int qtdGas){
10        foreach(Moto moto in motos){
11            moto.Abastece(qtdGas);
12        }
13    }
14
15    public static void Main(String[] args){
16        Moto moto01 = new Moto("Faz250");
17
18        Console.WriteLine("1");
19        Console.WriteLine(moto01);
20
21        Console.WriteLine("2");
22        AbasteceGas(moto01.GetGas(), 5);
23        Console.WriteLine(moto01);
24
25        Console.WriteLine("3");
26        AbasteceMoto(moto01, 3);
27        Console.WriteLine(moto01);
28
29        Console.WriteLine("4");
30        List<Moto> motos = new List<Moto>();
31        motos.Add(moto01);
32        motos.Add(new Moto("Mtx"));
33        AbasteceLista(motos, 2);
34        foreach(Moto moto in motos){
35            Console.WriteLine(moto);
36        }
37    }
38
39 }
```

---

