

2401 - PROGRAMAÇÃO ORIENTADA A OBJETOS - Resultados

1 Considere o programa abaixo escrito na linguagem C#

public class veiculo{ }

public class carro : veiculo{ }

public class aviao : veiculo{ }

Qual a afirmativa CORRETA?



- **a** A classe veiculo é superclasse das classes carro e aviao.
- **b** A classe aviao é subclasse da classe carro.
- **c** As classes carro e aviao são superclasses da classe veiculo.
 - **d** A classe veiculo é subclasse da classe aviao.
 - **e** As classes veículo e carro são subclasses da classe maquinas.

Pontuação: 1

2 Na linguagem C#, o polimorfismo refere-se à ligação tardia de uma chamada a uma ou várias implementações diferentes de um método em uma hierarquia de herança. Neste contexto, considere as seguintes classes descritas na Linguagem C#.

```
public class P001
{
    public virtual int Calcula()
    {
        return 1;}
}

public class P002 : P001
{
    public override int Calcula()
        {
            return base.Calcula() + 1;
        }
}

public class P003 : P002
{
    public override int Calcula()
        {
            return base.Calcula() + 1;
        }
}
```

Se estas classes forem utilizadas a partir do programa a seguir

```
public class TestaP00
{
    public static void Main(string[] args)
    {
        int result = 0;
        P001[] objs = new P001[3];
        objs[0] = new P001();
        objs[1] = new P002();
        objs[2] = new P003();

        for (int i = 0; i < 3; i++)
        {
            result += objs[i].Calcula();
        }

        Console.WriteLine(result);
    }
}</pre>
```

a saída desse programa será:

- **a** 3
- **b** 5
- **c** 0



e 9

Pontuação: 1

- **3** Assinale a alternativa CORRETA acerca dos conceitos de classes em linguagem C#:
 - **a** Uma classe pode implementar somente uma interface.
 - **b** Uma interface é uma classe abstrata para a qual todos os métodos são privados e todos os atributos são publicos.



- c Métodos abstratos de classes abstratas precisam do modificador abstract.
- **d** A forma básica de herança em C# é a extensão simples entre uma superclasse e sua classe derivada. Para tanto, utiliza-se na definição da classe derivada a palavra extends seguida pelo nome da superclasse.
- ✓ e Uma classe abstrata pode ser instanciada, ou seja, existem objetos que podem ser construídos diretamente de sua definição.

Pontuação: 1

4 Qual das seguintes afirmações descreve corretamente a relação entre classe e objeto na programação orientada a objetos?

- **a** Objetos são templates que definem estados e comportamentos, e classes são instâncias desses templates usadas para criar múltiplos objetos com o mesmo estado.
- **b** Classes e objetos são termos intercambiáveis na programação orientada a objetos e, portanto, têm o mesmo significado e uso.
- **c** Um objeto é uma definição abstrata que inclui métodos e propriedades, enquanto uma classe é a implementação concreta dessa definição.
- **d** Uma classe é uma instância de um objeto, servindo como um molde concreto a partir do qual os objetos são criados.
- Uma classe é uma blueprint ou template que define as características e comportamentos (métodos) que os objetos criados a partir dela terão.

Pontuação: 1

5 Qual é o modificador de acesso, utilizado na linguagem C#, para definir que manteremos somente uma cópia de determinados atributos na memória, independentemente da quantidade de objetos que forem instanciados?

Ou seja, o programa deve conter apenas uma cópia de cada variável definida com esse modificador em memória, mesmo se tivermos dez objetos instanciados.

Analise as alternativas e marque a que apresente esse modificador de acesso:

- **a** protected
- **b** private



- **d** default
- e public

Pontuação: 1

6 Considere o código em C# abaixo:

```
public class Calculadora{
  public int quadrado(int x){
    return x*x;
  }
  public double quadrado(int y){
    return y*y;
  }
}
```

Assinale a alternativa CORRETA acerca dos conceitos envolvidos na sobrecarga de métodos na linguagem C#:

> a A sobrecarga de métodos acontece na herança, quando a subclasse sobrepõe o método original.



- As chamadas de método podem ser distinguidas tendo o mesmo nome e tipos de retorno diferentes.
 - **c** Os métodos com o mesmo nome não podem ser declarados na mesma classe.
 - **d** Não podem existir métodos implementados com o mesmo nome.
 - **e** As assinaturas dos métodos quadrado() são exemplos de implementações de métodos estáticos.

Pontuação: 1

7 Considere a classe abaixo (na linguagem C#), que representa parte da definição de um curso de uma universidade:

```
public class Curso
{
    private string nome;

    public Curso(string nome)
    {
        this.nome = nome;
    }

    public string GetNome()
    {
        return this.nome;
    }
}
```

Modifique a classe **Curso** para que ela consiga armazenar os nomes das disciplinas que compõem o curso, lembre-se que a quantidade de disciplina de um curso pode variar de um curso para outro curso.

```
public class TestaCurso
{
    public static void Main(string[] args)
    {
        Curso c1 = new Curso("Ciência da Computação");
        c1.AddDisciplina("AlgProgII");
        c1.AddDisciplina("SO");
        c1.AddDisciplina("Grafos");

        Console.WriteLine(c1);
    }
}
```

Além disso a sua modificação deve atender o teste acima, e na execução do testes temos a seguinte saída:

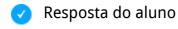
Curso:Ciencia da Computacao

Disciplinas:[AlgProgII, SO, Grafos]

```
Conceito: Errado - Pontuação: 0
Explicação:
using System;
using System.Collections.Generic;
class Curso
{
  private string nome;
  private List listaDisciplina;
  public Curso(string nome)
  {
    this.nome = nome;
    this.listaDisciplina = new List();
  }
  public string GetNome()
  {
    return this.nome;
  }
  public void AddDisciplina(string disciplina)
  {
    listaDisciplina.Add(disciplina);
  }
  public override string ToString()
  {
    return $"Curso: {this.nome}\nDisciplinas: {string.Join(", ", listaDisciplina)}";
  }
}
```

Legenda:

🧪 Alternativa correta



Pontuação total: 4