



PROGRAMAÇÃO O.O.

(C#)



Aula 04: Exercícios de concursos públicos!
Professor: João Luiz Lagôas



Questão de Concurso



Ano: 2009 **Banca:** FCC **Órgão:** TRT – (MA) **Prova:** Tecnologia da Informação

A diretiva public é utilizada em C# para aplicar a encapsulação pública

- a) aos métodos e classes, apenas.
- b) aos atributos e métodos.
- c) às classes, apenas.
- d) aos atributos, apenas.
- e) aos atributos e classes, apenas.

Questão de Concurso



Ano: 2009 **Banca:** FCC **Órgão:** TRT – (MA) **Prova:** Tecnologia da Informação

A diretiva public é utilizada em C# para aplicar a encapsulação pública

- a) aos métodos e classes, apenas.
- ☒ b) aos atributos e métodos.
- c) às classes, apenas.
- d) aos atributos, apenas.
- e) aos atributos e classes, apenas.

Questão de Concurso



Ano: 2008 **Banca:** CESPE **Órgão:** TRT (BA)

Prova: [Analista Judiciário – Tecnologia da Informação](#)

Em C#, os métodos public de uma classe são utilizados pelos clientes da classe para manipular dados armazenados em objetos dessa classe.

(Certo) (Errado)

Ano: 2012 **Banca:** CESPE **Órgão:** Banco da Amazônia

Prova: [Técnico Científico \(Análise de Sistemas\)](#)

Os métodos de uma classe C# criada com controle de acesso protect podem ser acessados apenas por classes do mesmo namespace.

(Certo) (Errado)

Questão de Concurso



Ano: 2008 **Banca:** CESPE **Órgão:** TRT (BA)

Prova: [Analista Judiciário – Tecnologia da Informação](#)

Em C#, os métodos public de uma classe são utilizados pelos clientes da classe para manipular dados armazenados em objetos dessa classe.



(Errado)

Ano: 2012 **Banca:** CESPE **Órgão:** Banco da Amazônia

Prova: [Técnico Científico \(Análise de Sistemas\)](#)

Os métodos de uma classe C# criada com controle de acesso protect podem ser acessados apenas por classes do mesmo namespace.

(Certo) (Errado)

Questão de Concurso



Ano: 2008 **Banca:** CESPE **Órgão:** TRT (BA)

Prova: [Analista Judiciário – Tecnologia da Informação](#)


Em C#, os métodos public de uma classe são utilizados pelos clientes da classe para manipular dados armazenados em objetos dessa classe.

(Certo)  (Errado)

Ano: 2012 **Banca:** CESPE **Órgão:** Banco da Amazônia

Prova: [Técnico Científico \(Análise de Sistemas\)](#)

Os métodos de uma classe C# criada com controle de acesso protect podem ser acessados apenas por classes do mesmo namespace.

(Certo)  (Errado)

Questão de Concurso



Ano: 2014 **Banca:** FCC **Órgão:** TRT **Prova:** Analista Judiciário (T.I.)

O valor que será impresso na execução do método main é

- a) 6.0
- b) 7
- c) 8.0
- d) 5.0
- e) 12.5

```
class Program
{
    static void Main(string[] args)
    {
        Program t = new Program();
        Console.WriteLine(t.Multi(2.5, 2));
        Console.ReadLine();
    }

    public float Multi(float flt, int n)
    {
        return flt * n + 1;
    }

    public int Multi(float dbl, double n)
    {
        return Convert.ToInt32(dbl * n + 2);
    }

    public double Multi(double i, int n)
    {
        return i * n + 3;
    }
}
```

Questão de Concurso



Ano: 2014 **Banca:** FCC **Órgão:** TRT **Prova:** Analista Judiciário (T.I.)

O valor que será impresso na execução do método main é

- a) 6.0
- b) 7
- ☒ c) 8.0
- d) 5.0
- e) 12.5

```
class Program
{
    static void Main(string[] args)
    {
        Program t = new Program();
        Console.WriteLine(t.Multi(2.5, 2));
        Console.ReadLine();
    }

    public float Multi(float flt, int n)
    {
        return flt * n + 1;
    }

    public int Multi(float dbl, double n)
    {
        return Convert.ToInt32(dbl * n + 2);
    }

    public double Multi(double i, int n)
    {
        return i * n + 3;
    }
}
```


Desafio de Certificação



What is the expected output?

- a) cougar c c
- b) cougar c f
- c) feline cougar c c
- d) feline cougar c f
- e) Compilation fails
- f) An exception is thrown at run time

```
class Feline
{
    public string type = "f ";
    public Feline()
    {
        Console.WriteLine("feline");
    }
}

class Cougar : Feline
{
    public Cougar()
    {
        Console.WriteLine("cougar");
    }

    public static void Main(string[] args)
    {
        new Cougar().go();
        Console.ReadLine();
    }

    void go()
    {
        type = "c ";
        Console.WriteLine(type + base.type);
    }
}
```

Desafio de Certificação



What is the expected output?

- a) cougar c c
- b) cougar c f
- ☒ c) feline cougar c c
- d) feline cougar c f
- e) Compilation fails
- f) An exception is thrown at run time

```
class Feline
{
    public string type = "f ";
    public Feline()
    {
        Console.WriteLine("feline");
    }
}

class Cougar : Feline
{
    public Cougar()
    {
        Console.WriteLine("cougar");
    }

    public static void Main(string[] args)
    {
        new Cougar().go();
        Console.ReadLine();
    }

    void go()
    {
        type = "c ";
        Console.WriteLine(type + base.type);
    }
}
```

Questão de Concurso



Ano: 2009 **Banca:** CESPE **Órgão:** ANAC]

Prova: Analista Administrativo - Tecnologia da Informação

Com relação aos conceitos de programação orientada a objetos e sua implementação na linguagem C#, julgue o próximo item.

Na programação orientada a objetos, o conceito de herança pode ser utilizado mediante a criação de subclasses a partir de classes anteriormente criadas. Em C#, as subclasses herdam as variáveis de instância e os métodos de instância da superclasse, podendo ter acesso a todos os membros referidos na superclasse.

(Certo) (Errado)

Questão de Concurso



Ano: 2009 **Banca:** CESPE **Órgão:** ANAC]

Prova: Analista Administrativo - Tecnologia da Informação

Com relação aos conceitos de programação orientada a objetos e sua implementação na linguagem C#, julgue o próximo item.

Na programação orientada a objetos, o conceito de herança pode ser utilizado mediante a criação de subclasses a partir de classes anteriormente criadas. Em C#, as subclasses herdam as variáveis de instância e os métodos de instância da superclasse, podendo ter acesso a todos os membros referidos na superclasse.

(Certo)



(Errado)

Questão de Concurso



Ano: 2012 **Banca:** CESPE **Órgão:** Banco da Amazônia

Prova: Técnico Científico (Análise de Sistemas)

Em C#, para todo método e atributo que se declara há um controle de acesso, independentemente de o controle ser explicitamente indicado.

(Certo) (Errado)

Ano: 2013 **Banca:** CESPE **Órgão:** INPI

Prova: Técnico Científico (Análise de Sistemas)

Uma classe pode acessar atributos de uma superclasse mesmo quando eles são declarados como private.

(Certo) (Errado)

Questão de Concurso



Ano: 2012 **Banca:** CESPE **Órgão:** Banco da Amazônia

Prova: Técnico Científico (Análise de Sistemas)

Em C#, para todo método e atributo que se declara há um controle de acesso, independentemente de o controle ser explicitamente indicado.



(☒) (Errado)

Ano: 2013 **Banca:** CESPE **Órgão:** INPI

Prova: Técnico Científico (Análise de Sistemas)

Uma classe pode acessar atributos de uma superclasse mesmo quando eles são declarados como private.

(Certo) (Errado)


Questão de Concurso



Ano: 2012 **Banca:** CESPE **Órgão:** Banco da Amazônia

Prova: Técnico Científico (Análise de Sistemas)


Em C#, para todo método e atributo que se declara há um controle de acesso, independentemente de o controle ser explicitamente indicado.

() (Errado)

Ano: 2013 **Banca:** CESPE **Órgão:** INPI

Prova: Técnico Científico (Análise de Sistemas)

Uma classe pode acessar atributos de uma superclasse mesmo quando eles são declarados como private.

(Certo) ()

Questão de Concurso



Ano: 2007 **Banca:** Consulplan **Órgão:** Chesf

Prova: Analista de Sistema

A partir das afirmativas abaixo relativas aos comandos da linguagem C#, pode-se afirmar que:

- I. A palavra chave class é usada para definir tipos derivados em C#. De fato, tais tipos constituem as principais estruturas dos programas orientados a objetos.
- II. O modificador private, quando aplicado a um atributo de classe, impede que qualquer outra classe tenha acesso a tal atributo.
- III. O símbolo : é usado para permitir a herança múltipla em C#.
- IV. O modificador protected, quando aplicado a um atributo de classe, permite que classes filhas desta classe, tenham acesso a este atributo.

Estão corretas apenas as afirmativas:

- a) II e III b) II e IV c) II, III e IV d) I, II e IV e) I e IV

Questão de Concurso




Ano: 2007 **Banca:** Consulplan **Órgão:** Chesf

Prova: Analista de Sistema

A partir das afirmativas abaixo relativas aos comandos da linguagem C#, pode-se afirmar que:

- I. A palavra chave class é usada para definir tipos derivados em C#. De fato, tais tipos constituem as principais estruturas dos programas orientados a objetos.
- II. O modificador private, quando aplicado a um atributo de classe, impede que qualquer outra classe tenha acesso a tal atributo.
- III. O símbolo : é usado para permitir a herança múltipla em C#.
- IV. O modificador protected, quando aplicado a um atributo de classe, permite que classes filhas desta classe, tenham acesso a este atributo.

Estão corretas apenas as afirmativas:

- a) II e III  b) II e IV c) II, III e IV d) I, II e IV e) I e IV

Questão de Concurso



Ano: 2010 **Banca:** CESPE **Órgão:** MPU

Prova: Analista de Desenvolvimento de Sistemas

No código C# mostrado ao lado, as classes Conta e Poupanca implementam o polimorfismo dinâmico.

(Certo) (Errado)

```
class Conta
{
    protected float saldo;
    public float getSaldo(int i)
    {
        float saldo = 0.0f;
        if (i == 1) saldo = this.saldo * 1.03f;
        return saldo;
    }
    public void setSaldo(float saldo)
    {
        this.saldo = saldo + 20f;
    }
}

class Poupanca : Conta
{
    public float getSaldo()
    {
        return saldo;
    }
}
```

Questão de Concurso



Ano: 2010 **Banca:** CESPE **Órgão:** MPU

Prova: Analista de Desenvolvimento de Sistemas

No código C# mostrado ao lado, as classes Conta e Poupanca implementam o polimorfismo dinâmico.

(Certo) (Errado)

```
class Conta
{
    protected float saldo;
    public float getSaldo(int i)
    {
        float saldo = 0.0f;
        if (i == 1) saldo = this.saldo * 1.03f;
        return saldo;
    }
    public void setSaldo(float saldo)
    {
        this.saldo = saldo + 20f;
    }
}

class Poupanca : Conta
{
    public float getSaldo()
    {
        return saldo;
    }
}
```

Questão de Concurso



Ano: 2008 **Banca:** ESAF **Órgão:** CGU

Prova: Tecnologia da Informação – Desenvolvimento de Sistemas

A programação orientada a objetos é baseada em diversos conceitos, tais como encapsulamento, herança, polimorfismo e abstração. Com relação a esses conceitos, é correto afirmar que:

- a) O conceito de encapsulamento é alcançado por meio da definição da visibilidade pública aos atributos e métodos.
- b) Herança é um mecanismo que permite que uma classe herde todo o comportamento e os atributos de outra classe. Em C#, pode-se implementar tanto a herança única quanto a múltipla naturalmente.
- c) Interface pode ser considerada como a forma com que um objeto se apresenta para outros, no que diz respeito aos seus atributos e métodos. Em C#, uma mesma classe não pode implementar mais de uma interface.
- d) Polimorfismo é o uso de um mesmo nome para identificar diferentes implementações de métodos. Seu uso é comum na definição de construtores, em que os mesmos podem ser implementados em diferentes versões para as diferentes formas de se instanciar classe.
- e) Para um classe ser considerada abstrata, todos os seus métodos devem ser abstratos. Em C#, para se definir uma classe abstrata deve-se utilizar a palavra chave “abstract” no início de sua declaração.

Questão de Concurso



Ano: 2008 **Banca:** ESAF **Órgão:** CGU

Prova: Tecnologia da Informação – Desenvolvimento de Sistemas

A programação orientada a objetos é baseada em diversos conceitos, tais como encapsulamento, herança, polimorfismo e abstração. Com relação a esses conceitos, é correto afirmar que:

- a) O conceito de encapsulamento é alcançado por meio da definição da visibilidade pública aos atributos e métodos.
- b) Herança é um mecanismo que permite que uma classe herde todo o comportamento e os atributos de outra classe. Em C#, pode-se implementar tanto a herança única quanto a múltipla naturalmente.
- c) Interface pode ser considerada como a forma com que um objeto se apresenta para outros, no que diz respeito aos seus atributos e métodos. Em C#, uma mesma classe não pode implementar mais de uma interface.



- d) Polimorfismo é o uso de um mesmo nome para identificar diferentes implementações de métodos. Seu uso é comum na definição de construtores, em que os mesmos podem ser implementados em diferentes versões para as diferentes formas de se instanciar classe.
- e) Para um classe ser considerada abstrata, todos os seus métodos devem ser abstratos. Em C#, para se definir uma classe abstrata deve-se utilizar a palavra chave “abstract” no início de sua declaração.

Questão de Concurso



Ano: 2010 **Banca:** CESGRANRIO **Órgão:** PETROBRAS

Prova: Analista Júnior – Eng. de Software

Ao tentar compilar e executar o código ao lado, o resultado será:

```
class Program
{
    static void Main(string[] args)
    {
        A obj = new B();
        obj.Metodo();
        obj = new C();
        obj.Metodo();
    }
}
```

```
abstract class A
{
    public A()
    {
        Console.WriteLine("A");
    }

    public abstract void Metodo();
}

class B : A
{
    public override void Metodo()
    {
        Console.WriteLine("B");
    }
}

class C : B
{
    public override void Metodo()
    {
        Console.WriteLine("C");
    }
}
```

Questão de Concurso



Ano: 2010 **Banca:** CESGRANRIO **Órgão:** PETROBRAS

Prova: Analista Júnior – Eng. de Software


- a) A correta compilação e execução do código, com a exibição na saída padrão da sequência BC.
- b) A correta compilação e execução do código, com a exibição na saída padrão da sequência ABAC.
- c) Um erro de compilação, pois A é uma classe abstrata e não pode ter instâncias, como obj.
- d) Um erro de compilação, pois obj é da classe A e tentou-se criar uma instância dela com o construtor da classe B.
- e) Um erro de execução, pois uma vez feito o binding de obj com a classe B, não se pode mudar a classe do mesmo.

Questão de Concurso



Ano: 2010 **Banca:** CESGRANRIO **Órgão:** PETROBRAS

Prova: Analista Júnior – Eng. de Software

- a) A correta compilação e execução do código, com a exibição na saída padrão da sequência BC.
-  b) A correta compilação e execução do código, com a exibição na saída padrão da sequência ABAC.
- c) Um erro de compilação, pois A é uma classe abstrata e não pode ter instâncias, como obj.
- d) Um erro de compilação, pois obj é da classe A e tentou-se criar uma instância dela com o construtor da classe B.
- e) Um erro de execução, pois uma vez feito o binding de obj com a classe B, não se pode mudar a classe do mesmo.

Questão de Concurso



Ano: 2014 **Banca:** NCE-UFRJ **Órgão:** BNDES

Prova: Analista de Desenvolvimento Júnior

Observe o trecho de código C# ao lado, definido em um único arquivo, C1.cs.

As variáveis que podem ser referenciadas no ponto assinalado, no interior do método m1, são:

- a) m, n
- b) j, k n
- c) j, k, m, n
- d) j, k, l, m, n
- e) i, j, k, l, m, n

```
class C1
{
    private int i;
    public int j;
    protected int k;
    int l;
}

class C2 : C1
{
    private int m;

    public void m1()
    {
        int n = 0;

        // que variáveis podem ser
        // acessadas aqui??
    }
}
```

Questão de Concurso



Ano: 2014 **Banca:** NCE-UFRJ **Órgão:** BNDES

Prova: Analista de Desenvolvimento Júnior

Observe o trecho de código C# ao lado, definido em um único arquivo, C1.cs.

As variáveis que podem ser referenciadas no ponto assinalado, no interior do método m1, são:

- a) m, n
- b) j, k n
- c) j, k, m, n
- d) j, k, l, m, n
- e) i, j, k, l, m, n



```
class C1
{
    private int i;
    public int j;
    protected int k;
    int l;
}

class C2 : C1
{
    private int m;

    public void m1()
    {
        int n = 0;

        // que variáveis podem ser
        // acessadas aqui??
    }
}
```

Desafio de Certificação



What is the expected output?

- a) alpha beta
- b) beta beta
- c) gamma gamma
- d) alpha alpha
- e) Compilation fails

```
class Program
{
    static void Main(string[] args)
    {
        Gamma g1 = new Alpha();
        Gamma g2 = new Beta();
        Console.WriteLine(g1.GetType() + " " +
g2.GetType());
    }
}

class Alpha
{
    string GetType() { return "alpha"; }
}

class Beta : Alpha
{
    string GetType() { return "beta"; }
}


class Gamma : Beta
{
    string GetType() { return "beta"; }
}
```

Desafio de Certificação



What is the expected output?

- a) alpha beta
- b) beta beta
- c) gamma gamma
- d) alpha alpha

 Compilation fails

```
class Program
{
    static void Main(string[] args)
    {
        Gamma g1 = new Alpha();
        Gamma g2 = new Beta();
        Console.WriteLine(g1.GetType() + " " +
g2.GetType());
    }
}

class Alpha
{
    string GetType() { return "alpha"; }
}

class Beta : Alpha
{
    string GetType() { return "beta"; }
}

class Gamma : Beta
{
    string GetType() { return "beta"; }
}
```