

Universidade Paulista
Curso Superior de Tecnologia

Exercícios da 1ª semana de quarentena

Nome: Daniel Pessoto Medeiros Guimarães RA: N4286J1

ADS 3º Semestre

Campus: Pinheiros

Matéria: Programação orientada a objeto I

Professor: Angel

São Paulo

Maio/2020

Todos os exercícios que possuem código, foram resolvidos em código:
https://github.com/dpessoto/Exercicios_Primeira_Semana_Quarentena

M9 - 1)

Dado o código abaixo:

```
class Aluno
{
    public String nome;
}
0 references
class Program
{
    0 references
    static void Main(string[] args)
    {
        new Aluno();
        new Aluno();
    }
}
```

Considere as afirmações abaixo:

- I. O programa não compila pois não apresenta instanciação.
- II. O programa cria duas instâncias
- III. Os objetos caso compilasse não cria objetos.

Solução:

Alternativa-Correta: B

Justificativa Cria 2 estâncias, sem referência.

M9 - 2)

```
1      public class Classe1
2      {
3          double a1, a2, a3;
4
5          public Classe1(double p)
6              : this(p, p)
7          {
8              Console.WriteLine("construtor com 1 parametro: ");
9          }
10
11         public Classe1(double p12, double p3)
12             : this(p12, p12, p3)
13         {
14
15             Console.WriteLine("construtor com 2 parametros");
16         }
17
18         public Classe1(double p1, double p2, double p3)
19         {
20             this.a1 = p1;
21             this.a2 = p2;
22             this.a3 = p3;
23             Console.WriteLine("construtor com 3 parametros: ");
24         }
25     }
26     class Program
27     {
28         static void Main(string[] args)
29         {
30             Classe1 c1 = new Classe1(1,1);
31         }
32     }
```

Qual a saída? Justifique transcrevendo a sequência de linhas executadas pelo código.

Solução:

Alternativa-Correta: D

Justificativa: Vai executar o construtor com 2 parâmetros, porém nesse método ele : this (p12, p12, p3), portanto vai chamar o construtor com 3 parâmetros e imprimir "construtor com 3 parametros:" primeiro, depois vai imprimir "construtor com 2 parametros", pois volta para o método com 2 parâmetros .

M9 - 4)

```
1      public class Classe1
2      {
3          public float d1;
4
5          public Classe1(float p)
6          {
7              this.d1=p;
8          }
9          public void metodo1(float m, float n)
10         {
11             d1 *= (m + n);
12         }
13         public void metodo1(float m)
14         {
15             this.metodo1(m, m);
16             d1 *= m;
17         }
18         public void metodo1()
19         {
20             this.metodo1(this.d1);
21         }
22     }
23     class Program
24     {
25         static void Main(string[] args)
26         {
27             Classe1 c1 = new Classe1(3);
28             c1.metodo1();
29             Console.WriteLine(c1.d1);
30         }
31     }
```

Quala saída? Justifique escrevendo a sequência de execução .

Solução:

Alternativa-Correta: D

Justificativa: Quando iniciado é chamado o construtor com 1 parâmetro, neste construtor chamado é feito o seguinte cálculo, $d1 *= (m + n)$, que equivale a $3 * (3+3)$, o resultado disto é 18, após isso ele retorna para o método de 1 parâmetro, onde executa $d1 *= m$, que equivale a $18 * 3$, que resulta em 54, e depois é impresso o valor de d1 que equivale a 54.

M9 - 5)

A classe Classe1 abaixo necessita ser totalmente encapsulada, sendo que o atributo total não pode ter acesso externo para modificação.

```
class Classe1
{
    public String nome;
    public float total=0;
    public void incrementa(float valor)
    {
        this.total += valor;
    }
}
class Program
{
    static void Main(string[] args)
    {
        Classe1 c1 = new Classe1();
        c1.nome = "Galileu";
        c1.incrementa(100);
        Console.WriteLine("{0} {1}\n",c1.nome,c1.total);
    }
}
```

Qual das alternativas apresenta o encapsulamento correto?

Quala saída? Justifique escrevendo a sequência de execução .

Solução:

Alternativa-Correta: D

Justificativa: Pois é o único que incrementa da forma correto o get que sempre retorna o atributo, e o set que atribui um valor ao atributo em questão.

M9 - 6)

```
class Motor
{
    public int Cilindros;
    public double Capacidade;
    public Motor(int cil, double cc)
    {
        this.Cilindros = cil;
        this.Capacidade = cc;
    }
}                                     classe motor

class Carro
{
    public string cor;
    public Motor motor;
    public Carro(String cor, Motor motor)
    {
        this.cor = cor;
        this.motor = motor;
    }
}                                     Classe carro.
```

Considere as seguintes operações:

1. Montar os carros com os seus motores

Carro	Cor	Motor
c1	Preto	m1
c2	Prata	m2

Motor	Cilindros	Capacidade
m1	4	1.0
m2	4	1.6
m3	4	1.8

2 - Trocar o motor do carro c2 para m3

3 - Aumentar a capacidade do motor do carro c1 para 1.2

Qual dos programas executa estas operações/ Justifique mostrando os erros das alternativas incorretas.

Solução:

Alternativa-Correta: B

Justificativa: Alternativa A está errada, pois quando está instanciando o objeto c1 ou o c2, como parâmetro ele passa, m1 ou m2, porém esses objetos não foram criados ainda.

Alternativa c, não tem instancia do objeto motor, na alternativa A tem, porém é criada depois das instâncias do objeto Carro.

Na D, ele passa m3 para m2, porém tinha que ser c2.motor= m3, pois estando m2=m3, não vai mudar o motor do c2.

Na E, as instancias c2 e c1, já foram declaradas o tipo dela, portanto não precisa repetir.

M9 - 7)

```
1      class Classe1 : Classe2
2      {
3          public Classe1()
4          {
5              Console.WriteLine("Lugar1");
6          }
7      }
8      class Classe2
9      {
10         public Classe2()
11         {
12             Console.WriteLine("Lugar2");
13         }
14     }
15     class Program
16     {
17         static void Main(string[] args)
18         {
19             Classe1 c1 = new Classe1();
20             Classe2 c2 = new Classe2();
21         }
```

Qual o resultado mostrado na tela ? Justifique escrevendo a sequência de execução

Solução:

Alternativa-Correta: D

Justificativa: O primeiro construtor a ser chamado é da classe 1 que extend a classe 2, assim ele escreve "Lugar 2", porque chama a classe 2, ai depois imprime "Lugar1", e depois "Lugar2" pois é estanciado a Classe2().