Generics e Coleções Genéricas – Exercícios

1- Quais das seguintes classes estão presentes no nameaspace System.Collections.Generic :
Stack<T> 2-Tree<T> 3-SortedDictionary<T> 4-SortedArray<T>

a-) 1 e 2 somente
b-) 2 e 4 somente
c-) 1 e 3 somente
d-) Todas as opções
e-) Nenhuma das opções

2- Dado o trecho de código abaixo :

```
public class Generic<T>
{
   public T Campo;
   public void TesteSub()
   {
      T i = Campo + 1;
   }
}
class MeuPrograma
{
   static void Main(string[] args)
   {
      Generic<int> gen = new Generic<int>();
      gen.TesteSub();
   }
}
```

Qual das seguintes declarações são verdadeiras ?

a-) A adição produzirá o resultado 1.
b-) O resultado da adição depende do sistema.
c-) O programa gerará uma exceção em tempo de execução.
d-) O compilador vai relatar o erro: O operador '+' não está definido para os tipos T e int.
e-) Nenhuma das acima.

3- Quais das declarações a seguir são verdadeiras para o recurso Generics da plataforma .NET ?

1- Generics é um recurso de linguagem.
2- Podemos criar uma classe genérica, porém não podemos criar uma interface genérica em C#
3- Delegates genéricos não são permitidos em C#.
4- O recurso Generics são úteis em classes de coleção na plataforma .NET

a- 1 e 2 somente
b- 1, 2 e 3 somente
c- 1 e 4 somente
d- Todas as opções
e- Nenhuma das opções

Generics e Coleções Genéricas - Exercícios

4- Qual o resultado da execução do código abaixo:

```
Teste teste = new Teste();
teste.MetodoTeste<string>("Usando Generics -> ");
teste.MetodoTeste<float>(4.2f);

Console.ReadKey();

public class Teste
{
    public void MetodoTeste<T>(T arg)
    {
        Console.Write(arg);
    }
}
```

a-) O programa vai compilar e na execução imprimirá: Generics -> 4.2
b-) Uma classe não genérica Teste não pode ter um método genérico MetodoTeste<T>
c-) O compilador vai gerar um erro.
d-) O programa irá gerar uma exceção em tempo de execução
e-) Nenhuma das opções acima.

5- Qual o resultado da execução do código abaixo:

```
Generic<String> g = new Generic<String>();
g.Campo = "Exercício Generics";
Console.WriteLine(g.Campo);

Console.ReadKey();

public class Generic<T>
{
    public T? Campo;
}
```

a-) O nome Generic não pode ser usado como um nome de classe porque é uma palavra-chave.
b-) Não podemos criar uma classe genérica e definir um campo genérico
c-) Vai imprimir a string "**Exercício Generics**" no console.
d-) O campo de membro da classe Generic não é acessível diretamente.
e-) Nenhuma das acima.

6- Para o trecho de código fornecido abaixo, quais das seguintes declarações é válida ?

```
public class MeuContainer<T> where T: class, IComparable
{
    // Insira o código aqui
}
```

a-) A classe MeuContainer requer que seu tipo de argumento implemente a interface IComparable.
b-) O argumento do tipo da classe MeuContainer deve ser IComparable.
c-) O compilador reportará um erro para este bloco de código.

Generics e Coleções Genéricas - Exercícios

d-) A classe MeuContainer requer que seu argumento de tipo seja um tipo de referência e implemente a interface IComparable.

7- Qual das seguintes afirmações é válida sobre as vantagens dos genéricos?

a-) Generics transferem o ônus da segurança de tipo para o programador em vez do compilador.
b-) Generics requerem o uso de conversão de tipo explícito.
c-) Generics fornecem segurança de tipo sem a sobrecarga de várias implementações.
d-) Generics eliminam a possibilidade de erros de execução.
e-) Nenhuma das acima

8- Escreva um programa para adicionar dois números inteiros usando o conceito de Generics.

9- Escreva um programa que crie uma lista de objetos Aluno que contém as propriedades : Nome, Idade e Sexo. A seguir defina 5 objetos do tipo Aluno e exiba uma lista de objetos alunos no console.

10 - implemente um programa que verifica se uma expressão matemática contém parênteses balanceados seguindo os seguintes passos:

1. Crie uma variável do tipo Stack<char> para armazenar os parênteses abertos.
2. Percorra cada caractere da expressão matemática.
3. Se o caractere for um parêntese aberto ('(', '{', '['), adicione-o à pilha.
4. Se o caractere for um parêntese fechado (')', '}', ']'), verifique se a pilha não está vazia e se o último parêntese aberto adicionado na pilha corresponde ao parêntese fechado atual. Se sim, remova o último parêntese aberto da pilha. Caso contrário, a expressão matemática não contém parênteses balanceados.
5. Após percorrer todos os caracteres da expressão matemática, verifique se a pilha está vazia. Se estiver vazia, a expressão matemática contém parênteses balanceados. Caso contrário, a expressão não é balanceada.

11. Implementar um programa que simula uma fila de impressão seguindo o seguinte roteiro:

- Crie uma variável do tipo Queue<string> para representar a fila de impressão.
- Crie um loop que irá executar até que a fila de impressão esteja vazia.
- Dentro do loop, verifique se a fila de impressão não está vazia. Se não estiver vazia, remova o primeiro elemento da fila usando o método Dequeue() e imprima na tela que o arquivo "X" está sendo impresso.
- Simule o tempo de impressão com um Thread.Sleep() por um período aleatório de tempo entre 1 e 5 segundos.
- Após simular a impressão do arquivo, imprima na tela que o arquivo "X" foi impresso com sucesso.
- Repita os passos 3 a 5 até que a fila de impressão esteja vazia.

12- Escreva um programa seguindo as seguintes orientações:

Declare um método genérico chamado **ReverterEImprimir** em uma classe não genérica chamada Exemplo.
O método recebe como parâmetro um array de qualquer tipo.

Generics e Coleções Genéricas - Exercícios

A seguir declare três tipos diferentes de array : um array de int , um array de strings e um array de double
Invoque o método duas vezes com cada array.
Na primeira vez invoque o método com um determinado array, onde ele usa explicitamente o parâmetro de tipo.
Na segunda vez, invoque o método onde o tipo é inferido.