//1.Массивы(Arrays): Создайте массив целых чисел и наполните его

//несколькими значениями от -10 до 10.

int[] mas = new int[4] { -8, 0, 4, 10 };

foreach (int i in mas)

Console.Write(i + " ");

Console.WriteLine();

for (int i = 0; i < mas.Length; i++)

Console.Write(mas[i] + " ");

//2. Списки (List): Создайте список строк и добавьте в него несколько

//элементов произвольных имен.

List<string> list = new List<string>() { "Michail", "Igor", "Masha", "Ira" };

list.Add("Ivan");

list.Add("Bob");

list.Add("Petr");

Console.WriteLine();

foreach (string i in list)

Console.WriteLine(i);

Console.WriteLine($"Количество элементов в коллекции List: {list.Count}");

list.Remove("Petr");

foreach (string i in list)

Console.WriteLine(i);

//3. Стек (Stack): Инициализируйте стек целых чисел и добавьте несколько

//элементов.

Stack<int> stack = new Stack<int>();

stack.Push(5);

stack.Push(3);

stack.Push(7);

stack.Push(9);

stack.Push(2);

stack.Push(4);

stack.Push(3);

//4. Продемонстрируйте операции добавления и извлечения элементов из

//стека.

Console.WriteLine(stack.Peek());

Console.WriteLine(stack.Pop());

Console.WriteLine(stack.Pop());

Console.WriteLine(stack.Pop());

Console.WriteLine(stack.Peek());

Console.WriteLine($"Количество элементов в стеке: {stack.Count}");

if (stack.Contains(7)) Console.WriteLine("Стек содержит цифру 7");

stack.Clear();

Console.WriteLine($"Количество элементов в стеке: {stack.Count}");

//5. Очередь (Queue): Создайте очередь строк и добавьте в нее элементы.

var queue = new Queue<string>(list);

Console.WriteLine($"Количество записей в очерди: {queue.Count}");

if (queue.Contains("Masha")) Console.WriteLine("Очередь содержит Masha");

Console.WriteLine(queue.Peek());

Console.WriteLine(queue.Dequeue());

Console.WriteLine(queue.Peek());

Console.WriteLine($"Количество записей в очерди: {queue.Count}");

queue.Enqueue("Mishail");

Console.WriteLine($"Количество записей в очерди: {queue.Count}");

Console.WriteLine(queue.Peek());

queue.Clear();

Console.WriteLine($"Количество записей в очерди: {queue.Count}");

//6. Словари (Dictionary): Создайте словарь, связывая строки с целыми

//числами (например, слова и их длины).

var diction = new Dictionary<int, string>();

for (int i = 0; i < 5; i++)

{

Console.WriteLine($"Введите {i} строку:");

string str = Console.ReadLine()!;

diction.Add(str.Length, str);

}

Console.WriteLine($"Количество элементов в словаре: {diction.Count}");

var dictString = $"Count: {diction.Count}\n";

int j = 0;

foreach (var val in diction)

dictString += $"[{j++}]: \"{val.Key}\" \"{val.Value}\"\n";

Console.WriteLine(dictString);

var diction2 = new Dictionary<int, string>();

diction2 = diction;

Console.WriteLine(diction2.Count);

diction2.Clear();

Console.WriteLine(diction2.Count);

//7. Множества (HashSet, SortedSet): Создайте множество целых чисел,

//добавьте в него несколько элементов.

//Класс HashSet<T> содержит неупорядоченный список различающихся элементов

//Rkfcc SortedSet<T> элементы упорядочены

HashSet<int> hash = new HashSet<int>();

Random random = new Random();

for (int i = 0; i < 25; i++)

{

hash.Add(random.Next(10, 100));

}

ShowCollHash(hash, "Hash множество");

SortedSet<int> ss = new SortedSet<int>();

SortedSet<int> ss1 = new SortedSet<int>();

for (int i = 0; i < 25; i++)

{

ss.Add(random.Next(10, 100));

}

ShowColl(ss, "Первая коллекция: ");

for (int i = 0; i < 25; i++)

{

ss1.Add(random.Next(10, 100));

}

ShowColl(ss1, "Вторая коллекция");

ss.SymmetricExceptWith(ss1);

ShowColl(ss, "Исключили разноименность (одинаковые элементы) двух множеств: ");

ss.UnionWith(ss1);

ShowColl(ss, "Объединение множеств: ");

ss.ExceptWith(ss1);

ShowColl(ss, "Вычитание множеств");

Console.ReadLine();

void ShowColl(SortedSet<int> ss, string s)

{

Console.WriteLine(s);

foreach (int ch in ss)

Console.Write(ch + " ");

Console.WriteLine("\n");

}

void ShowCollHash(HashSet<int> ss, string s)

{

Console.WriteLine(s);

foreach (int ch in ss)

Console.Write(ch + " ");

Console.WriteLine("\n");

}

//8.Коллекции для работы с ключами и значениями (KeyValuePair,

//KeyValuePair<TKey, TValue>): Создайте коллекцию KeyValuePair, связывая

//имена и возраст. Добавьте несколько элементов и покажите, как

//работать с этими парами.

try

{

var mike1 = new KeyValuePair<int, string>(56, "Mike1");

var mike2 = new KeyValuePair<int, string>(46, "Mike2");

var mike3 = new KeyValuePair<int, string>(39, "Mike3");

var mike4 = new KeyValuePair<int, string>(39, "Mike4");

var mike5 = new KeyValuePair<int, string>(27, "Mike5");

var employees = new List<KeyValuePair<int, string>>() { mike1, mike2, mike3, mike4, mike5 };

var people = new Dictionary<int, string>(employees);

dictString = $"Count: {people.Count}\n";

int j = 0;

foreach (var val in people)

dictString += $"[{j++}]: \"{val.Key}\" \"{val.Value}\"\n";

Console.WriteLine(dictString);

}

catch (ArgumentException)

{

Console.WriteLine("Неверный ввод");

}