**Zadaci – kolekcije**

**Predikati**

s => uslov nad s

**Primer:** Pronalaženje poslednjeg člana liste koji sadrži datu reč i svih članova liste koji sadrže datu reč.

 List<string> myList = new List<string>();

 myList.Add("Sumarice");

 myList.Add("Beograd");

 myList.Add("Kragujevac");

 myList.Add("Belosevac");

 string result = myList.FindLast(s => s.Contains("Be"));

 Console.WriteLine(result);

 Console.WriteLine("------------");

 List<string> resultL = myList.FindAll(s => s.Contains("Be"));

 foreach(string t in resultL) Console.WriteLine(t);

**Zadatak1. List<>**

Napisati program koji za unetiniz od n elemenata:

1. Ispituje da li u nizu postoji bar jedan par jednakih elemenata. Upotrebiti **List<>**, **Find()**, **Count()**

 static void Main(string[] args)

 {

 List<int> lista = new List<int>();

 int n = Convert.ToInt32(Console.ReadLine());

 int i;

 for (i = 0; i < n; i++)

 {

 lista.Add(Convert.ToInt32(Console.ReadLine()));

 }

 for (i = 0; i < n; i++)

 {

 if (lista.FindAll(s => s == lista[i]).Count() > 1) break;

 }

 if (i < n) Console.WriteLine("Ima duplikata");

 else Console.WriteLine("Nema duplikata");

 Console.ReadKey();

 }

1. Ispisuje broj različitih elemenata u nizu. **List<>, Distinct (), Count()**

 static void Main(string[] args)

 {

 List<int> lista = new List<int>();

 int n = Convert.ToInt32(Console.ReadLine());

 int i;

 for (i = 0; i < n; i++)

 {

 lista.Add(Convert.ToInt32(Console.ReadLine()));

 }

 Console.WriteLine(lista.Distinct().Count());

 Console.ReadKey();

 }

1. Formira niz različitih elemenata niza. **List<>, Distinct()**

 static void Main(string[] args)

 {

 List<int> lista = new List<int>();

 int n = Convert.ToInt32(Console.ReadLine());

 int i;

 for (i = 0; i < n; i++)

 {

 lista.Add(Convert.ToInt32(Console.ReadLine()));

 }

 List<int> pom = lista.Distinct().ToList();

 foreach (int x in pom)

 Console.Write(x);

 Console.ReadKey();

 }

**Zadatak 2. Dictionary<>**

Unosi se niz prirodnih brojeva. Dužina niza nije poznata, a obeležje kraja niza je nula. Napisati program kojim se za svaki broj koji se pojavljuje u nizu ispisuje broj njegovog pojavljivanja u tom nizu.

 static void Main(string[] args)

 {

 Dictionary<int,int> kolekcija = new Dictionary<int,int>();

 int br = Convert.ToInt32(Console.ReadLine());

 while (br!=0) {

 if (kolekcija.ContainsKey(br)) kolekcija[br]++;

 else kolekcija[br]=1;

 br = Convert.ToInt32(Console.ReadLine());

 }

 foreach(KeyValuePair<int,int> p in kolekcija)

 {

 Console.WriteLine("par {0} : {1}",p.Key,p.Value);

 }

 Console.ReadKey();

 }

**Zadatak 2.List<>**

Unosi se niz dvocifrenih prirodnih brojeva. Dužina niza nije poznata, a obeležje kraja niza je nula. Napisati program kojim se za svaki dvocifreni broj koji se pojavljuje u nizu ispisuje broj njegovog pojavljivanja u tom nizu.

Napomena: Dato rešenje nije optimalno, već samo demonstrira upotrebu pojedinih metoda kolekcije List<>

 static void Main(string[] args)

 {

 List<int> kolekcija = new List<int>();

 int br = Convert.ToInt32(Console.ReadLine());

 while (br!=0)

 {

 kolekcija.Add(br);

 br = Convert.ToInt32(Console.ReadLine());

 }

 foreach(int p in kolekcija.Distinct().ToList())

 {

 Console.WriteLine("par {0} : {1}",p,kolekcija.Count(x => x == p));

 }

 Console.ReadKey();

 }

**Zadatak 2. Queue<>**

Napraviti konzolnu igru namenjenu jednom do 4 igrača. Igrači prate kretanje robota na tabli šahovskoj tabli (beskonačnih dimenzija) i počevši od prvog pogađajaju njegovo kretanje. Ako igrač koji je na redu pogodi smer, dobija jedan poen i pogađa dalje. Igra se završava kada robot obiđe bar 64 različita polja ili jedan igrač sakupi više od 21 poena.

Kretanje robota se unapred generiše slučajnim izborom.

Primer:

Neka je kretanje robota opisano na sledeći način 1,1,1,2,2,3,4,4,4,3,1,1, pri čemu se smerovi kretanja opisuju na sledeći način gore -1, desno – 2, dole – 3, levo 4.

Neka u igri učestvuju 2 igrača, jedan od mogućih scenarija je:

Prvi igrač unosi 1 Trenutni skore: Prvi igrač – 1, Drugi igrač – 0

Prvi igrač unosi 2 Trenutni skore: Prvi igrač – 1, Drugi igrač – 0

Drugi igrač unosi 1 Trenutni skore: Prvi igrač – 1, Drugi igrač – 1

Drugi igrač unosi 1 Trenutni skore: Prvi igrač – 1, Drugi igrač – 1

Prvi igrač unosi 3 Trenutni skore: Prvi igrač – 1, Drugi igrač – 1

Drugi igrač unosi 1 Trenutni skore: Prvi igrač – 1, Drugi igrač – 1

…

BONUS:

Ako važi da robot ne izlazi iz okvira table dimenzija 20x20, napiši program kojim bi se iscrtala putanja kojom je robot prošao.