

12. Kolekcijas (2.daļa)

12.2. Hashtable

HashTable ir līdzīgs `ArrayList`, taču attēlo vienumus kā atslēgas un vērtības kombināciju. Tā izmanto atslēgu, lai piekļūtu kolekcijas elementiem. “Jaukšanas tabula” tiek izmantota, ja jums ir nepieciešams piekļūt elementiem, izmantojot atslēgu, un jūs varat identificēt noderīgu atslēgas vērtību.

Piemērs 1:

```
Hashtable numberNames = new Hashtable();
numberNames.Add(1, "One");
numberNames.Add(2, "Two");
numberNames.Add(3, "Three");

foreach (DictionaryEntry de in numberNames)
    textBox1.Text= textBox1.Text + de.Key + de.Value.ToString();

//izveido Hashtable ar collection-initializer sintaksi
var cities = new Hashtable(){
    {"UK", "London, Manchester, Birmingham"},
    {"USA", "Chicago, New York, Washington"},
    {"India", "Mumbai, New Delhi, Pune"}
};

foreach (DictionaryEntry de in cities)
    textBox1.Text = textBox1.Text + de.Key + de.Value.ToString();
```

Piemērs 2:

```
Hashtable ht = new Hashtable();

ht.Add("001", "John");
ht.Add("002", "Sam");
ht.Add("003", "Phill");
ht.Add("004", "Mike");

if (ht.ContainsValue("Jessica"))
{
    MessageBox.Show("This student name is already in the list");
}
else
{
    ht.Add("005", "Jessica");
}

// Get a collection of the keys.
ICollection key = ht.Keys;

foreach (string k in key)
{
    textBox1.Text=textBox1.Text+k + ": " + ht[k];
}
```

Lai atrast vienību: `textBox1.Text = ht["003"].ToString();`

Izvadā redzēsīm tikai vārdu, bez atslēgas. Dzēšanai izmanto metodi **Remove**:

```
Ht.Remove("003");
```

12.3. SortedList

Klase **SortedList** ir atslēgu (key) un vērtību pāru kolekcija, kas ir sakārtoti pēc atslēgām un ir pieejama pēc atslēgas un indeksa. SortedList ir masīva un hash tabulas kombinācija. Tajā ir to vienumu saraksts, kuriem var piekļūt, izmantojot atslēgu vai indeksu. Ja vienumiem piekļūstat, izmantojot indeksu, tas ir ArrayList, un, ja vienumiem piekļūstat, izmantojot atslēgu, tas ir Hashtable. Vienumu kolekcija vienmēr tiek sakārtota pēc atslēgas vērtības.

```
SortedList numberNames = new SortedList();
numberNames.Add(3, "Three");
numberNames.Add(1, "One");
numberNames.Add(2, "Two");
numberNames.Add(4, null);
numberNames.Add(10, "Ten");
numberNames.Add(5, "Five");

//rezultāti tiks sakārtoti pēc atslēgas vērtības
foreach (DictionaryEntry d in numberNames)
{
    textBox1.Text = textBox1.Text + d.Key + ": " + d.Value+Environment.NewLine;
}
```

12.4. Stack

Stack ir LIFO (pēdējais iekšā pirmais ārā) datu struktūra. Uztveriet to kā priekšmetu kaudzīti, kurā viss, ko ievietojat kaudzē, tiks novietots augšpusē un, ja jums kaut kas ir jānoņem, tas tiks noņemts no augšas. Plāksņu kaudze vai grāmatu kaudze ir divi izplatīti kaudzes piemēri.

```
Stack stk = new Stack();
//pievieno vienības
stk.Push("Pirmais");
stk.Push("Otrais");
stk.Push("Trešais");
stk.Push("Ceturtais");

foreach (object o in stk)
{
    textBox1.Text=textBox1.Text+o + " "+Environment.NewLine;
}

textBox1.Text = textBox1.Text + Environment.NewLine;

//nodzēš vienību
stk.Pop();

foreach (object o in stk)
{
    textBox1.Text = textBox1.Text + o + " " + Environment.NewLine;
}
```

12.5. Queue

Queue C# apzīmē FIFO (pirmais iekšā, pirmais ārā) objektu kolekciju. Klasiskais Queue piemērs ir cilvēku rinda, kas gaida.

```
Queue rinda = new Queue();
//pievieno vienības
rinda.Enqueue(1);
rinda.Enqueue(2);
rinda.Enqueue(3);
rinda.Enqueue(4);

foreach (object o in rinda)
{
    textBox1.Text=textBox1.Text+o + " "+Environment.NewLine;
}

textBox1.Text = textBox1.Text + Environment.NewLine;

//nodzēš vienību
rinda.Dequeue();

foreach (object o in rinda)
{
    textBox1.Text = textBox1.Text + o + " " + Environment.NewLine;
}
```

UZDEVUMI:

1. Izveidot programmu ar **HashTable** kolekciju lietotāju paroli uzglabāšanai pēc dota parauga. Programmā var apskatīt lietotāja paroli ievadot tā vārdu:

John	8~SP_ZK4nR*(5e\~
Jack	2?T6C[5neAZb6\NE
Mike	~%qG:gv+AQ@t2K4N
Sam	W9<]EZ5`W9YB/gN?
Tom	pV!n#Fpf{B&7<;.n

2. Izveidot programmu, kura izmanto **Queue** kolekciju apmeklētāju rindas uzskaitēi pēc šāda parauga:

Form3

Vārds, Uzvārds

Pievienot rindai

Izsaukt nākošo!

Rindas aktuālais sastāvs: