10. Funkcijas (metodes)

Jebkuru samēra neatkarīgu programmas fragmentu var noformēt kā apakšprogrammu un izsaukt no galvenās programmas. Priekš kam? Šāda programmas strukturēšana ievērojami atvieglo lielu programmu izstrādāšanu un lāgošanu.

- sadalot lielu un sarežģītu uzdevumu vairākos mazākos un vienkāršākos, būs vieglāk atrisināt to;
- vienas programmas izstrādāšanā var piedalīties vairāki programmētāji, sadalot darbu savu starpā,
- ja kāds fragments sastopas programmā vairākas reizes, apakšprogrammu izmantošana samazina programmas kodu.

Daudzās valodās izmanto divus apakšprogrammu veidus: funkcijas un procedūras. Gan funkcijas, gan procedūras var izpildīt kādas darbības. Bet funkcijas obligāti aprēķina un atgriež kādu vērtību. Valodā C# tādas apakšprogrammu sadalīšanas funkcijās un procedūrās nav - formāli eksistē tikai funkcijas. Bet ja apakšprogrammai nevajag neko aprēķināt un atgriezt, to noformē kā nenoteiktā tipa funkciju (**void**).
Tā kā **C#** ir stingri objektorientētā valoda, tajā nevar būt funkciju, kas eksistē pati par sevi,- tās visas obligāti ir atbilstošās **klases metodes**. Tas atspoguļojas arī funkciju (pareizāk - metožu) izsaukšanas sintaksē. Programmētājam ir iespēja ne tikai izmantot gatavas (standarta, bibliotēkas) metodes, bet arī pašam veidot savējās. Tieši tam ir veltīta šī nodaļa.
Piemēram, realizēsim metodi, kura parāda, cik vienā jūras jūdzē ir metru:

```
namespace Metodes
{
  public partial class Form1 : Form
  {
    public Form1()
    {
      InitializeComponent();
    }
    //metodes definēšana
    static double JudzesMetros(double z)
    {
      return z * 1852;
    }

    private void button1_Click(object sender, EventArgs e)
    {
      double y = double.Parse(textBox2.Text);
      text3.Text = JudzesMetros(y).ToString();
    }
  }
}
```

Atslēgas vārds **static** nozīmē, ka metodi var izsaukt, neveidojot tā saturošo klases eksemplāru. Mainīgo x funkcijas definēšanā sauc par parametru. Funkcijai var būt vairāki vai neviena parametra.

Piemēram, realizēsim metodi, kura neatgriež nekādu vērtību, tikai izvada paziņojumu uz ekrāna:

```
namespace Metodes
{
   public partial class Form1 : Form
   {
      public Form1()
      {
         InitializeComponent();
      }
      static void Pazinot()
      {
         MessageBox.Show("Viss ir kārtībā!");
      }

      private void button1_Click(object sender, EventArgs e)
      {
         Pazinot();
      }
   }
}
```

Piemēram: Standarta metode **Math.Sin** aprēķina leņķa sinusu, pie tam leņķim ir jābūt radiānos. Izveidojiet metodi **SinG** sinusa aprēķināšanai leņķim, izteiktam grādos.

```
namespace Metodes
{
   public partial class Form1 : Form
   {
      public Form1()
      {
         InitializeComponent();
      }

      //metodes definēšana
      static double SinG(double x)
      {
         return Math.Sin(x * Math.PI / 180);
      }

      private void button1_Click(object sender, EventArgs e)
      {
         double y = double.Parse(textBox1.Text);
         //metodes izsaukšana, izmantošana
         label1.Text = SinG(y).ToString();
      }
   }
}
```

C# ļauj izmantot masīvus gan kā metodes parametrus, gan kā atgriežamas vērtības. Pie tam masīvi-parametri vienmēr ir parametri-norādes, t.i. to vērtības var mainīties metodes izpildes gaitā. Nākošais piemērs demonstrē šīs iespējas:

Metode **DoRndArray**(N) veido masīvu, kurā ir **N** nejaušie veseli skaitļi ar vērtībām diapazonā [0; 100).

```
        ...
static int[] DoRndArray(int N)
{
   Random r = new Random();
   int[] A = new int[N];
   int k;
   for (k = 0; k < N; k++)
      A[k] = r.Next(100);
   return A;
}

private void button1_Click(object sender, EventArgs e)
{
   textBox1.Text = "";
   int N = 10;
   int[] A = DoRndArray(N);
   for (int i = 0; i < A.Length; i++)
   {
      textBox1.Text = textBox1.Text + A[i].ToString()+Environment.NewLine;
   }
}
        ...
```

UZDEVUMI:

1. Izstrādāt programmu, kura ļauj ievadīt EUR/USD kursu un tad ļauj veikt konvertāciju abos virzienos. Konvertācijas aprēķinus realizēt kā metodes.
2. Izveidot metodi kvadrātvienādojuma sakņu aprēķinam un nodemonstrējiet programmā to pielietojumu.