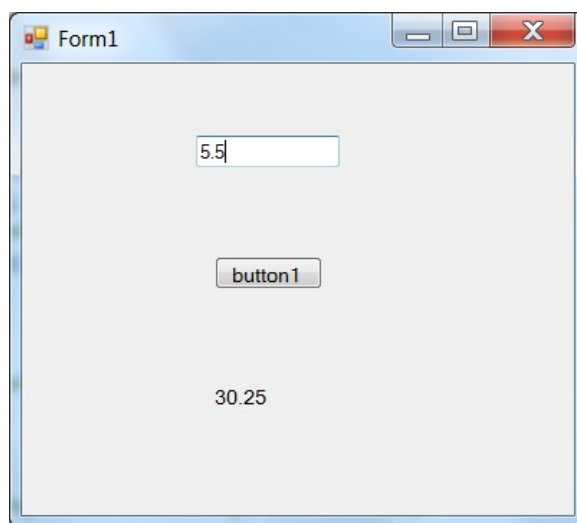


## VJEŽBA 6: Soketi (3. dio, programski jezik C#)

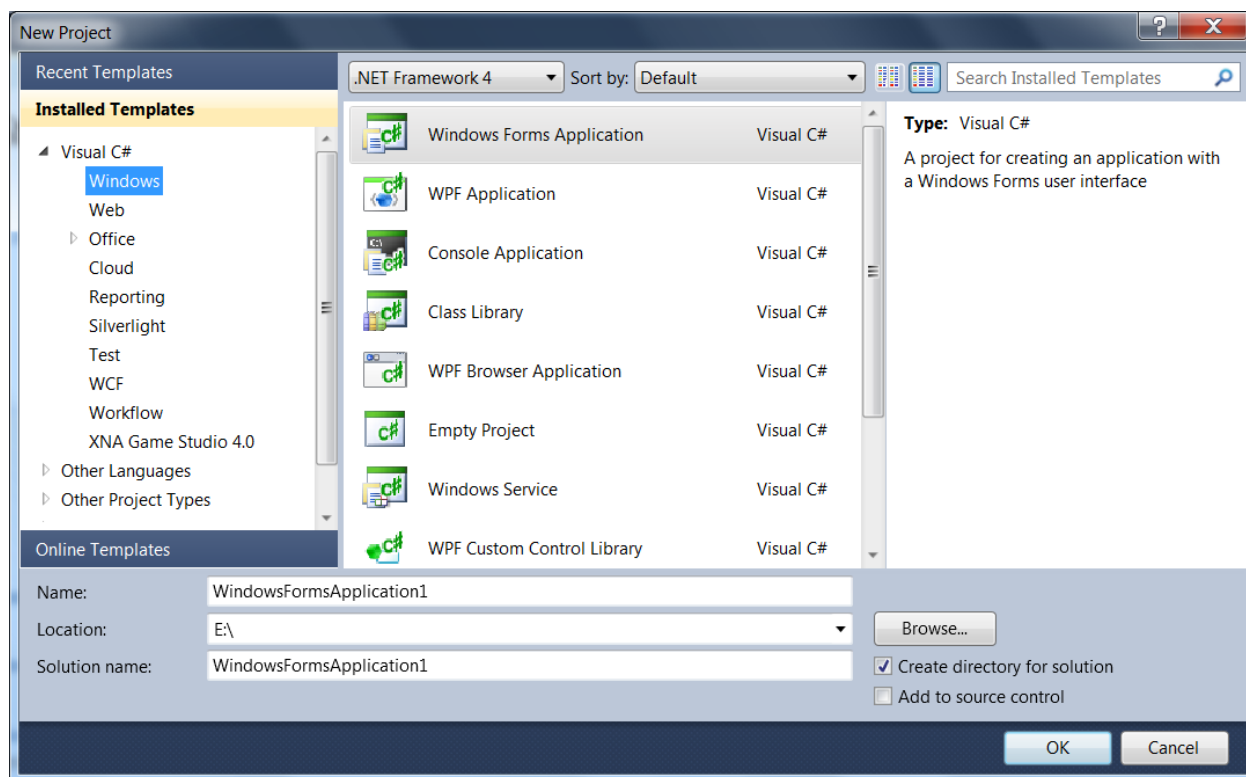
Zadatak ove vježbe je napraviti soket klijenta u C# koji se izvršava na lokalnom računalu na kojem radite vježbu, a spaja se na soket servera iz 4. vježbe (isprogramiranog u C-u). Server se izvršava na udaljenom Linux serveru (adria.fesb.hr ili neki drugi server koji možete koristiti). Klijent se ponaša isto kao i klijent iz 4. vježbe, tj. serveru šalje neki broj (tipa float), a server klijentu vraća kvadrirani broj.

Proširiti ćemo funkcionalnost klijenta tako da ćemo klijentu ugraditi grafičko korisničko sučelje. Izgled sučelja je prikazan na slici 1. Korisnik upiše broj u polje za unos teksta (TextBox kontrola), i kada pritisne botun, klijent se spoji na server, pošalje broj, sačeka povratnu vrijednost, ispiše je u labeli i zatvori konekciju prema serveru.



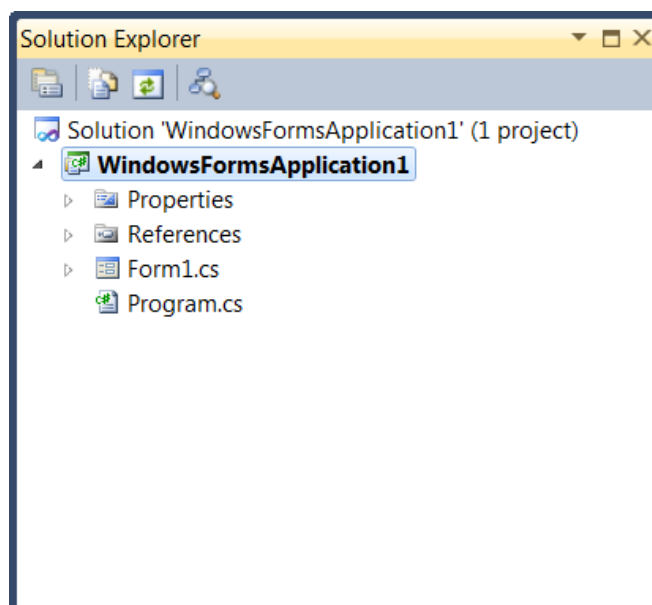
*Slika 1. Izgled korisničkog sučelja*

Za razvoj klijenta koristite Visual Studio aplikaciju. Vaša aplikacija se kreira korištenjem gotovog predloška za aplikaciju tipa Windows Forms Application kojem se pristupa na sljedeći način (slika 2): **Izbornik File -> New -> Project-> Windows Forms Application**. Možete promijeniti Location i Name opcije prema želji.

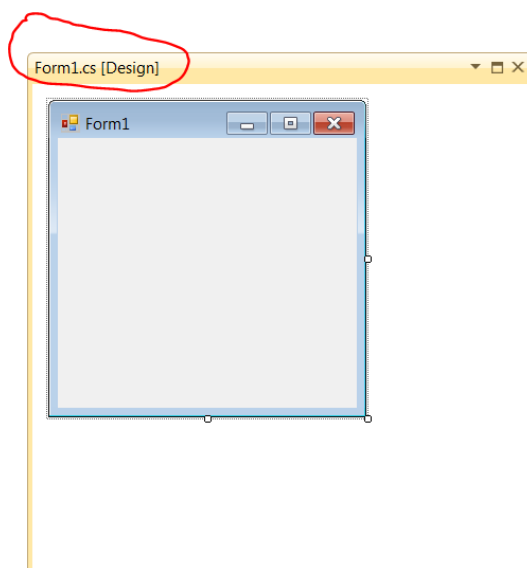


*Slika 2. Korištenje gotovog predloška za izradu Windows Forms aplikacije*

U prozoru Solution Explorer (slika 3) (ako vam ovaj prozor nije otvoren možete ga otvoriti na slijedeći način: **Izbornik View-> Solution Explorer**) kliknite dva puta na Form1.cs i dobiti ćete Design prikaz forme (slika 4).

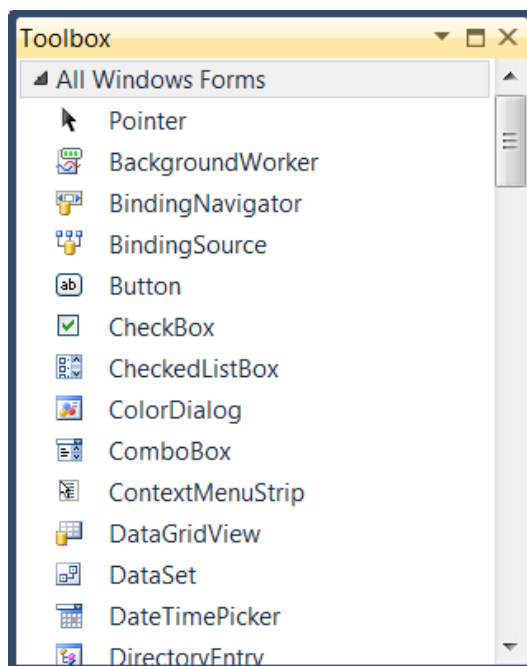


*Slika 3. Solution Explorer*



*Slika 4. Design prikaz forme*

U Design prikazu dodate element na formu preko Toolbox prozora (slika 5) (ako vam ovaj prozor nije otvoren možete ga otvoriti na slijedeći način: **Izbornik View -> Toolbox**) tako da željeni element selektirate u toolboxu i držeći lijevi botun miša vučete do forme. Elemente koje trebate dodati na formu su Button, Label i TextBox. Rasporedite ih po formi tako da svi budu vidljivi.



*Slika 5. Toolbox prozor*

Da biste isprogramirali C# soket klijenta, trebate napraviti funkciju koja će se pozivati svaki put kada korisnik mišem klikne na botun na formi. Kostur te metode možete dobiti tako da dva puta klijevim mišem kliknete na botun na formi (u Visual Studiju u Design prikazu). Automatski će se kreirati funkcija koja će se pozivati svaki put kada korisnik klikne na miša. Tu ubacite C# kod za soket klijenta.

Dohvaćanje podataka i pisanje podataka u i iz elemenata sučelja radi se preko imena objekta i odgovarajućeg svojstva (TextBox i Label).

Vaš kod treba izgledati ovako:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.Net.Sockets; // za rad sa soketima (ovo morate sami uključiti)

namespace WindowsFormsApplication1
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            // povežite se soketom na server
            TcpClient cl = new TcpClient();
            cl.Connect("adria.fesb.hr", 50576);
            NetworkStream str = cl.GetStream();

            // ovdje dodajte kod za slanje i primanje podataka sa servera

            str.Close();
        }
    }
}
```

Koristite funkcije Write (str.Write()) i Read (str.Read()) za pisanje i primanje podataka sa servera.

Funkcija **Write** prima četiri argumenta:

- niz tipa Byte[] u kojemu su zapisani podaci koji se trebaju poslati serveru;
- offset koji kaže sa kojega bajta u nizu se započinju slati podaci serveru;
- broj bajtova iz niza koji se žele poslati serveru (ako želite poslati sve podatke u nizu, ovdje upišite veličinu niza).

Funkcija **Read** prima četiri argumenta:

- niz tipa Byte[] u kojega će se zapisati podaci koji se prime sa servera;
- offset koji kaže na kojemu će se bajtu u nizu početi spremati podaci sa servera;
- broj bajtova koji se žele primiti sa servera.

Da biste mogli podatke poslati serveru, najprije ih trebate konvertirati u niz bajtova. To možete napraviti na slijedeći način:

```
Byte[] array = System.Text.Encoding.ASCII.GetBytes(podaci koje šaljete serveru);
```

Da biste primljene podatke mogli ispisati u labeli na formi, trebate ih iz niza bajtova konvertirati u string. To možete napraviti na slijedeći način:

```
label1.Text = System.Text.Encoding.ASCII.GetString(podaci koje ste primili od servera);
```

Vodite računa da na početku trebate uključiti imenski prostor naredbom **using System.Net.Sockets**.

## Dodatak: server u C-u

Programski kod za servera u C-u dan je u nastavku teksta (u slučaju da ste svoj kod sa 4. vježbe izbrisali sa adria servera):

```
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <string.h>

int main(int argc, char *argv[])
{
    int ld, sd;
    struct sockaddr_in skaddr;
    struct sockaddr_in from;
    socklen_t addrlen, length;

    ld = socket(PF_INET, SOCK_STREAM, 0);

    if (ld < 0)
    {
        printf("Problem sa kreiranjem soketa!\n");
        exit(-1);
    }

    skaddr.sin_family = AF_INET;
    skaddr.sin_addr.s_addr = htonl(INADDR_ANY);
    skaddr.sin_port = htons(50576); // promijenite broj porta u slučaju da je ovaj zauzet

    int result = 0;
    result = bind(ld, (struct sockaddr *) &skaddr, sizeof(skaddr));

    if (result < 0)
    {
        printf("Problem sa bindingom!\n");
        printf("Port mozda nije dostupan!\n");
        exit(-1);
    }

    length = sizeof(skaddr);

    result = 0;
    result = getsockname(ld, (struct sockaddr *) &skaddr, &length);
    if (result < 0)
    {
        printf("Doslo je do greske prilikom dohvacanja imena soketa!\n");
        exit(-1);
    }
}
```

```

}

printf("Server je na portu %d\n",ntohs(skaddr.sin_port)); // network to host short

if (listen(ld,5) < 0)
{
    printf("Greska prilikom osluskivanja konekcije!\n");
    exit(-1);
}

while (1)
{
    printf("Cekam na vezu ...\n");
    addrlen = sizeof(skaddr);

    sd = accept(ld, (struct sockaddr*) &from, &addrlen);

    if (sd < 0)
    {
        printf("Problem sa prihvacanjem veze!\n");
        exit(-1);
    }

    printf("Soketi su spojeni. Racunam ...\n");

    char buffer[20]; // max 20 znakova
    int n = 0; // broj bajtova kopiranih u buffer, 1 bajt po znaku
    n = read(sd, buffer, 20);
    printf("Broj kopiranih bajtova: %d\n", n);

    buffer[n] = '\0';
    float primljeni_broj = 0;
    primljeni_broj = atof(buffer);

    printf("Primljeni broj: %f\n", primljeni_broj);

    float kvadrirani_broj = 0;
    kvadrirani_broj = primljeni_broj * primljeni_broj;

    char rezultat[50];
    sprintf(rezultat, "%f", kvadrirani_broj);
    printf("Saljem rezultat klijentu... Rezultat je %s.\n", rezultat);
    write(sd, rezultat, strlen(rezultat));

    printf("Kraj racunanja!\n\n\n");
    close(sd);
}

return 0;
}

```

Servera pokrećete korištenjem slijedećih naredbi:

```

gcc -Wall -o server server.c
./server

```