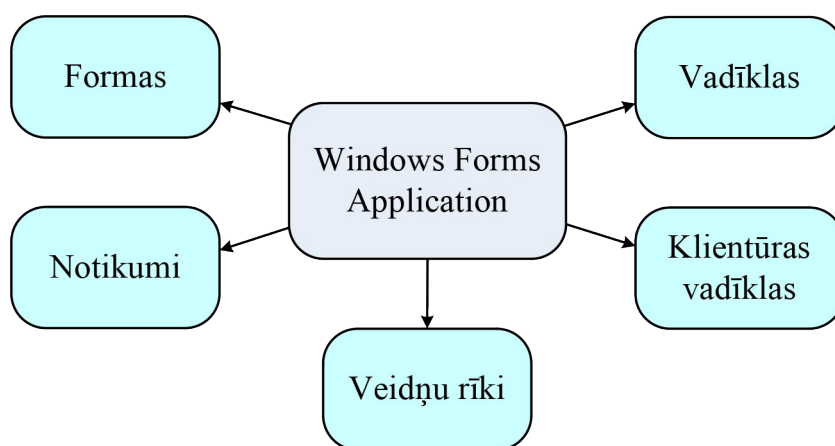


2. Vienkāršās Windows aplikācijas izveidošana

Šajā nodaļā tiks parādīts Windows aplikācijas izveidošanas process ar MS Visual Studio 2005.

2.1. Windows aplikācijas jēdziens

Windows Forms Applications ir Visual Studio sastāvdaļa, kura ļauj izveidot spēcīgas Windows aplikācijas. Tā piedāvā dažādus šablonus rīkus, struktūras un sākotnējo izejas kodu. Galvenās Windows formu īpatnības ir parādītas 2.1.attēlā.



2.1.att. Windows formu aplikācijas sastāvdaļas

Formas – vizuālā virsma, kura izpilda konteīnera funkcijas dažādu vadīklu izvietošanai, kuras attēlo izvada informāciju un ļauj saņemt ieejas datus no lietotāja.

Notikumi (*events*) – darbības, kuras tiek izpildītas pēc kāda notikuma izpildes (notikumu piemēri ir peles taustiņa klikšķis, programmas koda reakcija uz ievadītiem datiem un citi).

Vadīklas (*controls*) – iepriekš definētas vadīklas lietotāja saskarnes veidošanai (piemēram, teksta laukums (*text box*), poga (*button*) un citi).

Klientūras vadīklas (*custom controls*) – programmētāja definētās vadīklas, kad parasto vadīklu funkcionalitāte nav piemērota.

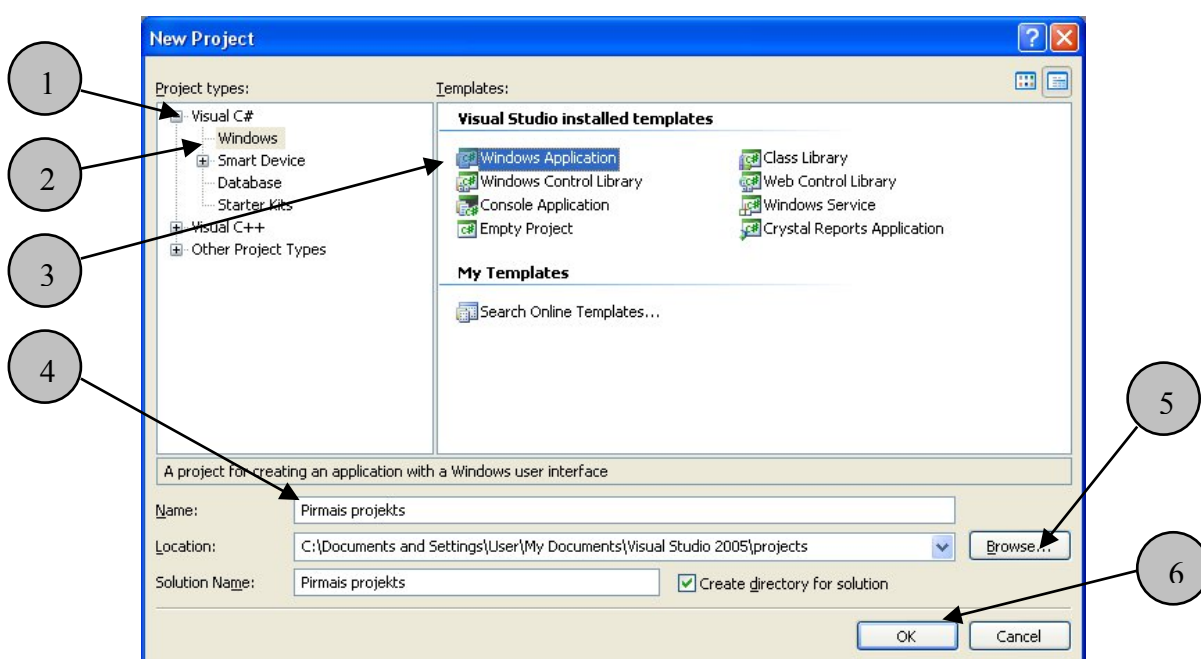
Veidņu rīki (*designer tools*) – Windows aplikācijas veidošanas paātrināšanai paredzētie grafiskie veidņi.

Windows aplikācijas parasti lieto gadījumos, kad programma tiks lietota uz lokālā datora; kad svarīga ir ātra programmas izpilde (reakcija uz lietotāja

darbībām); kad ir nepieciešama jaudīga un daudzfunkcionālā lietotāja saskarne. Tīmekļa aplikācijas, savukārt, izmanto tad, ja ir nepieciešams izplatīt un sniegt pieeju programmai globāli (caur tīklu) un programmas reakcijas laikam nav augstākā prioritāte.

2.2. Windows aplikācijas projekta izveidošana

Darba sākums ar MS Visual Studio ir šāds: **Start – Programs – Microsoft Visual Studio 2005 – Microsoft Visual Studio 2005**. Parādīsies izstrādes vides galvenais logs, kurā izvēlnes joslā izpilda komandas **File – New – Project**. Tālāk izvēlas projekta opcijas (2.2.attēls).

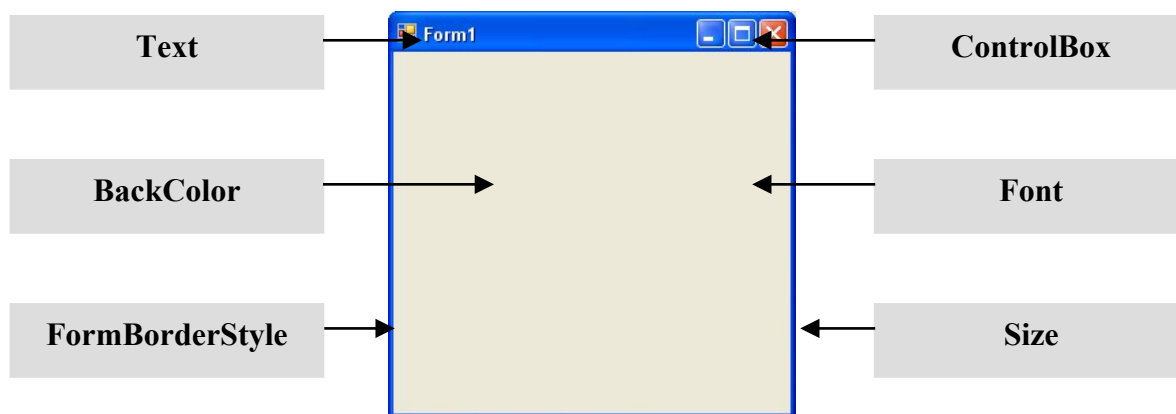


2.2.att. Windows aplikācijas projekta izveidošana

- 1- sākumā izvēlas programmēšanas valodu (Visual C#);
- 2- tad projekta tipu (Windows);
- 3- pēc tam izvēlas šablona veidu (Windows Application);
- 4- ieraksta projekta nosaukumu;
- 5- nepieciešamības gadījumā var pameinīt projekta saglabāšanas vietu;
- 6- apstiprina ar OK.

Projektam vienmēr var pievienot papildus objektu. Jaunu objektu pievieno ar komandu **Add – New Item**, eksistējošo (iepriekš saglabāto) ar komandu **Add – Existing Item**.

Izveidojot jaunu Windows aplikāciju ekrānā parādās forma dizaina skatā. Kā jebkurai objektam Visual Studio, arī formai ir savas īpašības, svarīgākas no kurām ir redzamas 2.3.attēlā.



2.3.att. Formas svarīgākās īpašības

(Name) – formas nosaukums, kurš tiek izmantots izejas kodā formas identificēšanai;

Text – nosaka formas virsraksta tekstu (piemērā tas ir „Form1”);

BackColor – formas fona krāsa;

FormBorderStyle – definē formas apmales izskatu un uzvedību;

ControlBox – minimizācijas, maksimizācijas un darba beigas pogu esamība uz formas;

Font – nosaka teksta īpašības objektiem, kuri tiks izvietoti uz formas (fonta nosaukums, izmērs un stils);

Size – formas izmēra definēšana;

StartPosition – definē formas atrašanās vietu monitora ekrānā.

2.3. Vadīklas un to pievienošana formai

Vadīklas paredzētas lietotāja saskarnes veidošanai. Visas vadīklas Visual Studio vidē ir sakārtotas šādās grupās:

Common Controls – biežāk lietojamās vadīklas, tādas, ka Button, Label vai TextBox;

Containers – objekti, kas paredzēti citu objektu grupēšanai; objektus konteinerā var apstrādāt kopā;

Menus & Toolbars – izvēlņu un rīkjoslu vadīklas;

Data – vadīklas, kuras nodrošina pieeju ārējiem datu avotiem;

Components – vispārīgās vadīklas bez lietotāja saskarnes (neredzamas programmas izpildes laikā), kurus izmanto kopā ar citām vadīklām (piemēram, ErrorProvider tiek izmantots lietotāja ievadīto datu pārbaudei);

Printing – drukas un lappuses iestatījumu vadīklas;

Dialogs – dialoga logu vadīklas;

Crystal Reports – atskaišu veidošanai paredzētas vadīklas.

Vadīklu īpatnības izpaužas šādos veidos:

Mijiedarbība ar lietotāju – vadīklas spēj reaģēt uz dažādiem notikumiem (piem., peles taustiņa nospiešana); katram notikumam var definēt savu funkcionalitāti.

Īpašības – katrai vadīklai ir savas gan specifiskās, gan vispārīgās īpašības, kā, piemēram, (Name) – nosaukums vai Text – teksta virsraksts.

Izkārtojums (Layout) – dažām vadīklām ir specifiskās izkārtojuma īpašības, kuras nosaka izvietošanu, noformējumu un piesaistes veidu.

Vadīklas pievienošanai **Toolbox** rīku paletē atrod nepieciešamo elementu un transportē uz formu. Vairāku vadīklu sakārtošanai var izmantot dažādus paņēmienus:

Vadīklu izlīdzinājums – atlasa nepieciešamas vadīklas, izpilda komandu Format – Align un izvēlas izlīdzinājuma veidu (izlīdzinājums tiek veikts attiecībā uz vadīklu, kura tika atlasīta pirmā).

Vienāda izmēra vadīklu iegūšanai atlasa vadīklas un izpilda komandu Format – Make Same Size.

Brīvo vietu sakārtošanai starp vadīklām izmanto komandu Format un horizontālo vai vertikālo atstarpes variantu (Horizontal vai Vertical Spacing).

Tab secības kontrole tiek nodrošināta ar komandu View – Tab Order (programmas izpildes laikā lietotājs var pārvietoties starp vadīklām ar tastatūras taustiņu Tab).

Apskatīsim vadīklas pievienošanu un tās funkcionalitātes definēšanu uz pogas piemēra, kad pogas nospiešanas gadījumā parādās paziņojumu logs.

1. Ievieto vadīklu **Button** uz projekta formu.
2. Ar peles kreiso taustiņu divreiz nospiež uz tās.
3. Teksta redaktora logā automātiski parādīsies notikuma apstrādes sākotnējais kods:

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

kur figūriekavās pievienojam pogai funkcionalitāti.

4. Informācijas izvadei paziņojuma logā ieraksta šādu kodu:

```
MessageBox.Show("Hello, World!", "Mans paziņojums",
    MessageBoxButtons.OKCancel, MessageBoxIcon.Asterisk);
```

5. Rezultātā nospiežot pogu uz ekrāna parādīsies šāds paziņojumu logs:



Apskatītajā piemērā informācijas izvadam tika izmantots **MessageBox** objekts un tās metode **Show**. Šai metodei var būt dažādi argumenti (šajā piemērā četri):

Text – paziņojuma teksts:

```
MessageBox.Show("Hello, World!");
```

Caption – paziņojuma loga virsraksts:

```
MessageBox.Show("Hello, World!", "Mans paziņojums");
```

Buttons – standarta pogu esamība paziņojumu logā:

```
MessageBox.Show("Hello, World!", "Mans paziņojums",  
MessageBoxButtons.OKCancel);
```

Icon – attēls paziņojumu logā:

```
MessageBox.Show("Hello, World!", "Mans paziņojums",  
MessageBoxButtons.OKCancel, MessageBoxIcon.Asterisk);
```

2.4. Projekta kompilācija un izpilde

Pirms projekta izpildes (palaišanas) ir nepieciešams uzbūvēt (build) projektu (citiem vārdiem nokompilēt visas projekta sastāvdaļas). Pastāv divi kompilācijas veidi:

Debug – aplikācija tiek kompilēta ar pilno kompilācijas informāciju (klūdas un informatīvie paziņojumi) un bez izejas koda optimizācijas (izmanto projekta izplatīšanas fāzē).

Release – aplikācijas kompilācija ar izejas koda optimizāciju un bez papildus informācijas (izmanto projekta sagatavošanai izplatīšanai).

Kompilācijas veidu var pamainīt projekta īpašībās (risinājumu pārvaldniekā uz projekta nosaukuma nospiež ar peles labo taustiņu un izvēlas komandu **Properties**, tad sadaļā **Build** no saraksta **Configuration** var izvēlēties nepieciešamo būvēšanas režīmu).

Kompilācijas laikā izejas kodā var būt dažādas kļūdas. Šajā gadījumā programmas kompilācija tiek pārtraukta un parādās kļūdu logs (**Error List Window**). Šajā logā ir visu kļūdu un brīdinājumu saraksts, divreiz nospiežot uz konkrētā ieraksta var uzreiz pāriet tajā projekta datnē un koda daļā, kur bija pielaista kļūda.

Pēc projekta kompilācijas to var izpildīt. Izstrādes laikā izmanto komandu **Debug - Start Debugging** vai **F5**. Programma tiks izpildīta ar ieslēgto kļūdas meklēšanas režīmu, ja notiek kļūdas gadījums, programmas izpilde tiek pārtraukta. Programmas izpildi manuāli var pārtraukt ar **Shift+F5** taustiņu kombināciju.

Nodaļas kopsavilkums

Windows aplikācijas galvenās īpatnības ir formas, notikumi, vadīklas, klientūras vadīklas un veidņu rīki.

Formas galvenās īpašības ir (Name), Text, BackColor, FormBorderStyle, ControlBox, FontSize un StartPosition.

Vadīklas paredzētas lietotāja saskarnes veidošanai. Tās spēj reaģēt uz dažādiem notikumiem, un katram notikumam var definēt savu funkcionalitāti.

Pirms projekta izpildes (palaišanas) ir nepieciešams uzbūvēt projektu. Pastāv divi kompilācijas veidi: Debug un Release. Pirmo veidu izmanto aplikācijas izstrādes laikā, otro tās sagatavošanā nosūtīšanai klientiem.

UZDEVUMI PAŠPĀRBAUDEI

1. Raksturojiet katru Windows aplikācijas sastāvdaļu.
2. Aprakstiet Windows aplikācijas izveidošanas secību.
3. Kā pievienot vadīklu formai un kādas ir būtiskākās Toolbox vadīklu grupas?
4. Ar ko atšķiras Debug un Release kompilācijas?

UZDEVUMI PATSTĀVĪGAJĀM DARBAM

1. Izveidojiet jaunu Windows aplikāciju, norādot projekta nosaukumu „Pirma programma” un tās atrašanās vietu mapē My Documents.
2. Pamainiet formas īpašības un šādām vērtībām:

Text: **Mana jauna forma;**

Size: **256, 238;**

StartPosition: **CenterScreen.**

3. Izvietojiet uz formas trīs pogas (Button) un divus teksta laukumus (TextBox), kā arī definējiet Tab secību līdzīgi parādītajām attēlā:

