

## Компоненти на интегрирана развојна околина (IDE), Запознавање на елементите на околина за визуелен развој - час 2

Програмската околина речиси секогаш се состои од едитор, компајлер, библиотека на готови програми и дебагер. Овие програми може да бидат посебни или меѓусебно поврзани – интегрирани во една програма која се вика интегрирана развојна околина (Integrated Development Environment – IDE) за програмскиот јазик.

**Програмски едитор** е специјално дизајниран текстуален едитор во IDE каде се пишува изворниот код на програмата. Најчесто програмските едитори овозможуваат синтаксичко маркирање во различна боја, автоматско порамнување на исказите, автоматско комплетирање при пишување на исказите, синтаксни проверки, спарување на загради и сл.

**IDE (integrated development environment) за визуелно програмирање** е софтвер апликација која со своите карактеристики им овозможува на програмерите на лесен начин да креираат апликација со визуелен пристап т.е. кориснички интерфејс со рамки, листи, командни копчиња и други графички елементи кои се лесни за употреба од страна на корисникот.

Денес посовремените програмски јазици нудат визуелно програмирање како што се Visual Basic, Visual C++, Java и др.

Кратко упатство за користење на Visual C++:


### VISUAL C++


1) форма (проект) се отвора со постапката:


File-> New-> Project-> Visual C++ -> CLR-> се става име на проектот во рамката Name и се избира локација каде да се сними проектот во рамката Location -> потоа се избира Windows Form Application -> OK

2) Се отвора прозорец со име Form1. Прво се поставуваат карактеристики на прозорецот Form1 во прозорецот Properties кој се отвора со десен клик на Form1 и се отвора обично десно во работниот прозорец. Најчесто користени карактеристики се:

- Text – се пишува насловот за форма - прозорецот,
- BackColor – се избира боја за внатрешноста на прозорецот,
- BackgroundImage – се поставува слика во внатрешноста на прозорецот,
- Font – избор на фонт, големина и стил на букви кои ќе се поставуваат во прозорецот,
- Size - ширина и висина на прозорецот (а можни и со влечење на карактеристичните точки кога е селектиран прозорецот)

3) Се поставуваат елементи во прозорецот со помош на алатките сместени во Toolbox  (кој се наоѓа на десниот раб од прозорецот – кај Properties , или горе десно во лентата со алатки). Најчесто користени алатки се:

 **Label** - за поставување на рамки во кои се испишува текст на прозорецот,

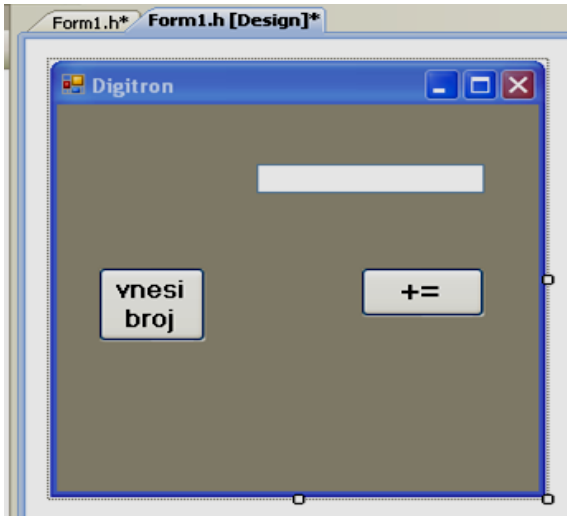
 **Button** - командно копче, со кое ќе се извршува некој програмски код,

## abl TextBox

- текстуална рамка за внесување на податоци и излез на резултати (замена за cin и cout наредбите), само што во ова поле каков и да е податокот што се внесува секогаш е од типот на string (текст), кој потоа во програмата треба да се конвертира со една од готовите функции за претворање во соодветен тип на податок.

**Забелешка:** За секој елемент што ќе се постави во прозорецот, секогаш постои Properties прозорецот за тој елемент во кој се избираат соодветни карактеристики.

### ПРИМЕР НА ФОРМА:



Во формата Digitron има:

-еден елемент TextBox чие име - Name е textBox1 (во Properties) и ова поле ќе се користи за внесување на броеви, а и тука ќе се гледа резултатот;

- два елемента Button т.е. „vnesi broj“ именувано како Button1 и „+=“ именувано како Button2 во полето Name (во Properties), додека во полето Text (во Properties) треба да се внесе текстот што сакаме да стои врз овие командни копчиња а тоа е vnesi broj и += соодветно.

За двете командни копчиња „vnesi broj“ и „+=“ треба да се напиши соодветен програмски код и тоа:  
-Со двоен клик на командното копче „vnesi broj“ ќе се отвори програмскиот код за формата и позиционерот ќе биде поставен на местото каде што треба да се пишува кодот:

```
private: System::Void button1_Click(System::Object^ sender, System::EventArgs^ e) {  
    | // позиционер  
}
```

Тука треба да се напиши следното:

```
br1=Convert::ToDouble(textBox1->Text);  
textBox1->Text=""; // за празнење на рамката откако ќе се прочита првиот број
```

**Објаснување:** Бидејќи во TextBox секој податок што се внесува е од типот на String (текст), со командата Convert::ToDouble(textBox1->Text) текстот од полето textBox1 се конвертира во податок од тип на Double и се сместува во променливата br1- променлива каде што се сместува првиот број.

-Со двоен клик на командното копче „+=“ ќе се отвори програмскиот код за формата и позиционерот ќе биде поставен на местото каде што треба да се пишува кодот, како на **сликата:**

```
private: System::Void button2_Click(System::Object^ sender, System::EventArgs^ e) {  
    | // позиционер  
}
```

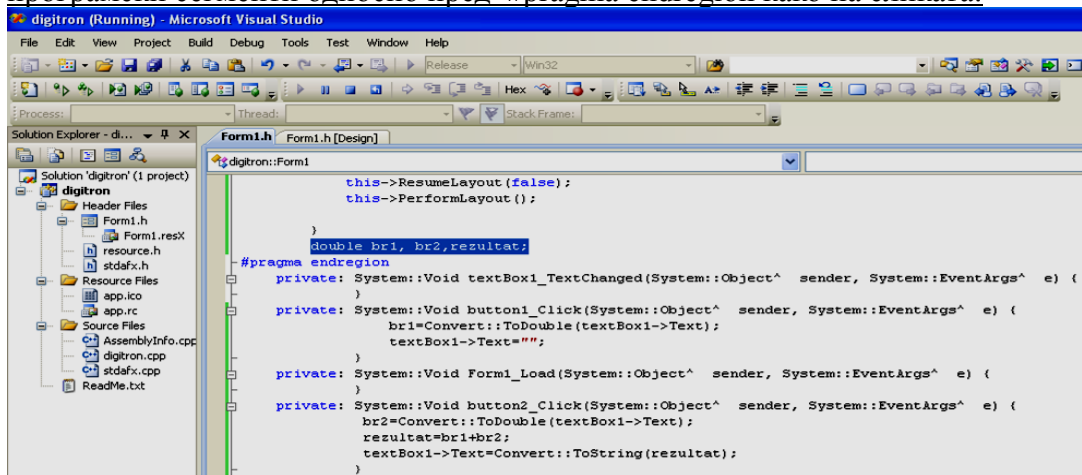
Тука треба да се напиши следното:

```
br2=Convert::ToDouble(textBox1->Text);  
rezultat=br1+br2;  
textBox1->Text=Convert::ToString(rezultat);
```

**Објаснување:** Со br2 се дефинира променлива каде што ќе се смести вториот број, а со rezultat се дефинира променлива каде што се сместува збирот на двата броја. Тој резултат треба да го видиме во TextBox рамката која е именувана со textBox1, при што резултатот од број треба да се конвертира во String (текст) , а тоа се прави со командата

```
textBox1->Text=Convert::ToString(rezultat);
```

Забелешка: Трите променливи br1, br2 и rezultat мора да се дефинираат како Double пред почетокот на овие програмски сегменти односно пред #pragma endregion како на сликата:

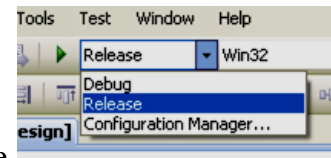
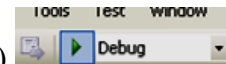


```
double br1, br2, rezultat;
#pragma endregion
private: System::Void textBox1_TextChanged(System::Object^ sender, System::EventArgs^ e) (
)
private: System::Void button1_Click(System::Object^ sender, System::EventArgs^ e) (
    br1=Convert::ToDouble(textBox1->Text);
    textBox1->Text="";
)
private: System::Void Form1_Load(System::Object^ sender, System::EventArgs^ e) (
)
private: System::Void button2_Click(System::Object^ sender, System::EventArgs^ e) (
    br2=Convert::ToDouble(textBox1->Text);
    rezultat=br1+br2;
    textBox1->Text=Convert::ToString(rezultat);
)
}
this->ResumeLayout (false);
this->PerformLayout ();
```

3) Проектот се стартува со Debug (зелена стрелка во лентата со алатки)

4) За да се направи извршна верзија на проектот (.exe) – односно да функционира само прозорцот на дигитронот без да се отвора Visual C++, постапката е:

Од мени Project-> Properties-> Configuration Properties-> C/C++-> Code Generation -> и во Runtime Library се избира Multi-threaded Debug (/MTd) -> Apply -> OK.



Потоа кај зелената стрелка за Debug од листата се избира Release и повторно на зелената стрелка за стартување. Во папаката каде што е снимен проектот има папка Release во која веќе има извршна верзија на digitron.

**Прашања поврзани со наставните единици може да се испишаат на email: [anetastojceska@gmail.com](mailto:anetastojceska@gmail.com)**