Juhász Tibor:

Visual Studio 2015 Express for Windows Desktop

Első lépések

Kiegészítések a Programozási ismeretek (Műszaki Könyvkiadó, 2011, 2015) tankönyvhöz

Tartalom

Bevezetés

Ez a tájékoztató a *Programozási ismeretek* tankönyv[1] (Műszaki Könyvkiadó, 2011, 2. kiadás: 2015) kiegészítése. A tankönyvben lehetőség szerint a programozási nyelvektől, illetve fejlesztőrendszerektől függetlenül ismertetjük a programozási tudnivalókat. Az alábbiakban bemutatjuk a Visual Studio 2015 Express Edition for Windows Desktop beállításait, használatát és a Visual Basic programok készítésének módját. A fejlesztőrendszer áttekintése csak a tankönyvhöz kapcsolódó elemekre vonatkozik.

A fejlesztőrendszert egy Windows-alkalmazás létrehozásával ismertetjük. Az egyes lépéseket olyan részletességgel írjuk le, hogy a teljesen kezdők számára is érthetőek legyenek. Ezek után már nem jelenthet gondot a tankönyv gyakorlatainak az értelmezése, elkészítése.

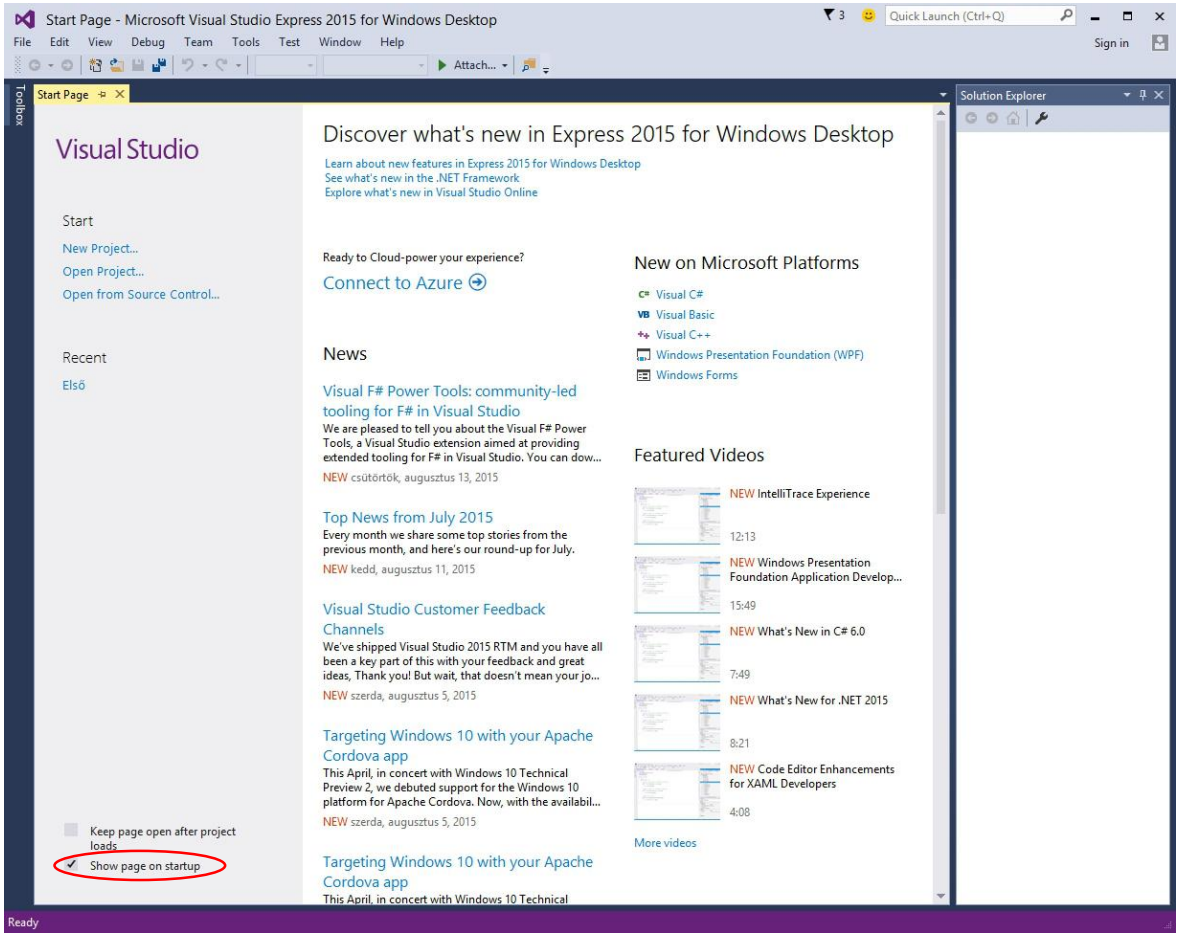
A Visual Studio letöltését, telepítését A Visual Studio Express 2015 telepítése dokumentum ismerteti.

A Visual Studio indítása és felhasználói felülete

Az alábbiakban az integrált fejlesztőrendszerre vonatkozó parancsoknál megadjuk a parancs helyét a menüben, majd zárójelben az eszköztár megfelelő ikonját, illetve a hozzá tartozó billentyűkombinációt.

A Visual Studio-t a *Start* menü vagy az *Asztal* parancsikonjának a segítségével indíthatjuk el (lásd a fenti telepítési útmutatót). Az első indítás egy kicsit hosszabb ideig tarthat. A megjelenő ablak tartalmazza a Windowsban megszokott menüsort, eszköztárat, alul pedig az állapotsort.

Az ablak jobb szélén helyezkedik el az egyelőre üres megoldástallózó (*Solution Explorer*), melyet a fájlok kezeléséhez használunk. A bal szélen találjuk az összecsukott eszközkészletet (*Toolbox*). Ha nem látjuk valamelyik munkaablakot, akkor az a *View* menü segítségével hívhatjuk elő (lásd később). A startlap (*Start Page*) megkönnyíti az új programok létrehozását (*New Project*), illetve az előzőleg készített programok megnyitását (*Open Project*, illetve a *Recent* lista). A startlap megjelenését a balra lent található *Show page on startup* jelölőnégyzet kikapcsolásával tilthatjuk le (eléréséhez szükségünk lehet a függőleges gördítősávra).



A Visual Basic 2015 integrált fejlesztőrendszere a startlappal

[1] Lásd: http://www.muszakikiado.hu/webaruhaz/kozepiskola/informatika/programozasi_ismeretek

Visual Basic programok létrehozása

A Visual Studio-nak több fájlra van szüksége a program létrehozásához. Ezeknek a fájloknak az összességét projektnek, az egymással kapcsolatban álló projektek csoportját pedig megoldásnak (*Solution*) nevezzük.[2]

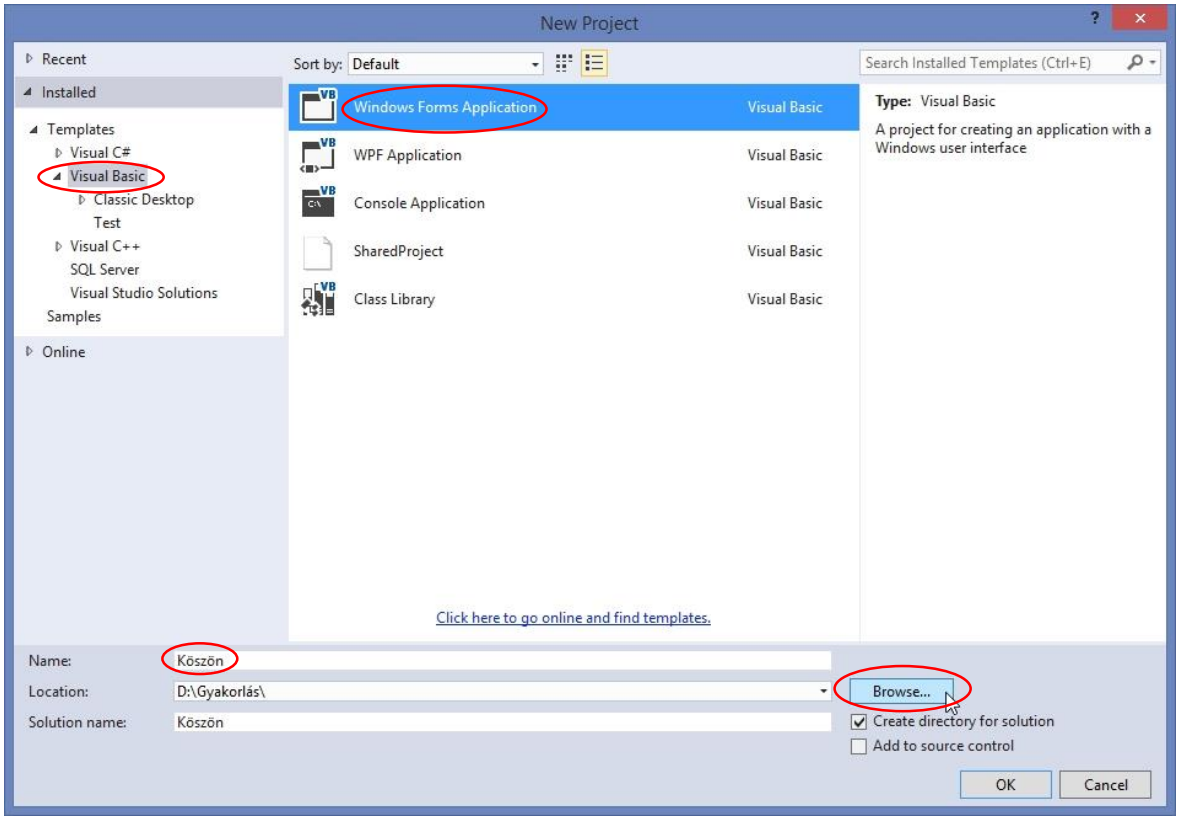
Új projektet (programot) a *File* menü *New project* parancsával hozunk létre (, *Ctrl+Shift+N*). A Visual Basic több, előre elkészített sablonnal segíti a programozást. A sablon a készülő programot keretbe foglaló utasításokat tartalmazza.

A megjelenő *New Project* ablakban látjuk a rendelkezésre álló sablonokat. A bal oldali listából válasszuk ki a Visual Basic-et, a sablonok közül pedig a *Windows Forms Application*-t!

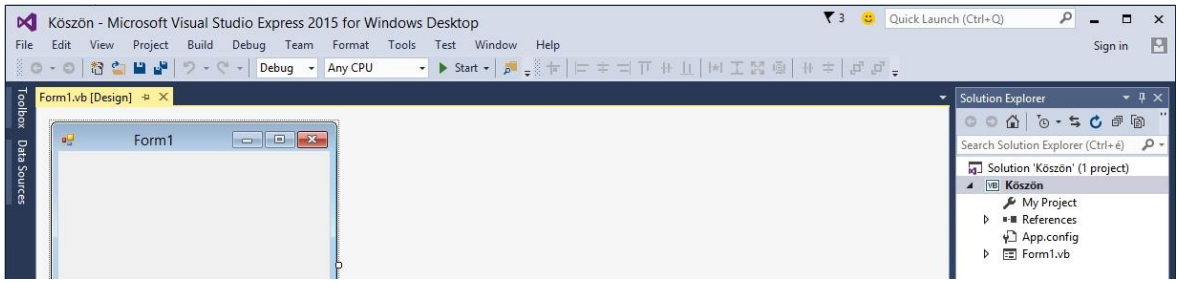
A *Name* szövegdobozba írjuk be a projekt (program) nevét (*Köszön*). Ne használjunk szóközt és írásjeleket! A *Solution name* szövegmezőt a fejlesztőrendszer automatikusan kitölti, de utólag módosíthatjuk.

Adjuk meg a projektfájlok helyét is! A *Browse* gombra kattintva megjelenik a *Project Location* ablak, amelyben a Windows fájlkezelőjével megszokott módon jelölhetjük ki a projektek tárolásához választott mappát. Mint a *Browse* gomb alatti feliratból látható, a projekt (megoldás) új mappába kerül (*Create directory for solution*).

Ha rákattintunk az *OK* gombra, létrejön a projekt.



Windows-alkalmazás létrehozása



A program az űrlappal és a megoldástallózóval

A képernyőn középen megjelenik a tervezőablak (*Design*), a jobb szélen a *Megoldástallózó* (*Solution Explorer*), alatta pedig a *Tulajdonságok* (*Properties*) munkaablak. A bal szélen az összecsukott *Eszközkészletet* (*Toolbox*) és az ugyancsak összecsukott *Adatforrás* (*Data Source*) munkaablakot látjuk.

---

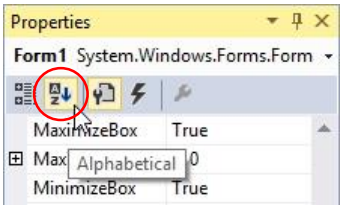
[2] Részletesebben lásd a *Programozási ismeretek* tankönyv 13. oldalán.

Kattintsunk rá a *Data Source* feliratra, ezzel megnyitjuk a munkaablakot! A *Bezárás* ( ) gombbal zárjuk be, nem lesz rá szükségünk.

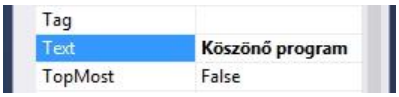
Nyissuk meg az *Eszközkészletet* is! A munkaablakok megjelenését az *Automatikus elrejtés* ( ) gombjával szabályozatjuk. Az álló gombostű folyamatosan nyitva tartja a munkaablakot. Az automatikus elrejtés bekapcsolása után (fekvő gombostű) rá kell kattintanunk a munkaablak feliratára a megnyitáshoz.[3]

Az ablak tulajdonságainak módosítása

A tervezőablakban a *Form1* nevű űrlapot látjuk, amely a képernyőn megjelenő programablakot jelképezi.[4] Kattintsunk rá az űrlapra. A *Tulajdonságok* munkaablakban megkapjuk az űrlap tulajdonságainak listáját. A kategóriák szerinti csoportosítás helyett válasszuk az ábécérendben történő megjelenítést, mert így könnyebben találjuk meg a keresett tulajdonságot.



A tulajdonságok ábécérendben történő megjelenítése



A címsor szövegének megadása

Keressük meg a *Text* (szöveg) tulajdonságot, ami az ablak címsorának a szövegét határozza meg. Írjuk be a *Köszönő program* címet. Ha lenyomjuk az Entert vagy az ablak más területére kattintunk, máris megjelenik az űrlap címsorában a begépelt szöveg.

Szükség esetén további tulajdonságokat is módosíthatunk. A *BackColor* például a háttérszínt, a *Size* a méretet jelenti pixelben. Ez utóbbit az űrlap szélein elhelyezkedő méretezőfogantyúk ( ) segítségével is megváltoztathatjuk. A (*Name*) tulajdonság az űrlapobjektum azonosítóját adja meg. Több ablak létrehozásánál ezt célszerű átírni, az ablak funkciójára utaló nevet választani. Most nem módosítjuk.

A folytatás előtt mentsük el a projektet! Ehhez válasszuk a *File* menü *Save all* parancsát (, *Ctrl+Shift+S*). Ne feledjük, hogy a projekt több fájlból (és több mappából) áll! Ezért ne használjuk a *Save* (, *Ctrl+S*) parancsot, amely csak az aktuális fájlt menti el! A projekt utólagos átnevezésére pedig nincs módunk.

Programunk elkészült. Itt az ideje, hogy elindítsuk. A fejlesztői környezeten belül ezt a *Debug/Start Debugging* paranccsal (Start, *F5* funkcióbillentyű) tehetjük meg. Így a programot az úgynevezett hibakereső üzemmódban hajtjuk végre. Jegyezzük meg az *F5* billentyűparancsot, mert használata nagymértékben felgyorsítja a munkát!

A képernyőn megjelenik egy üres ablak, a beállított tulajdonságokkal. A program futtatásakor további munkaablakok jelenhetnek meg (*Watch*, *Memory* stb.). Ezeket zárjuk be, mert nem lesz rájuk szükségünk!

Kattintsunk a programablak *Bezárás* gombjára vagy a menü *Debug/Stop Debugging* parancsára (, *Shift+F5*). Így visszatérünk a fejlesztőrendszerbe. Próbáljuk meg módosítani az ablak tulajdonságait. A módosítás után ismét futtassuk a programot!

Jegyezzük meg, hogy a módosítások végrehajtása előtt mindig be kell zárni a programot!

Hálózati környezetben végzett munka esetén előfordulhat, hogy nincs elegendő jogunk a hibakereső üzemmódban történő futtatáshoz. Ekkor használjuk a *Debug/Start Without Debugging* (*Ctrl+F5*) parancsot (futtatás hibakeresés nélkül)!

---

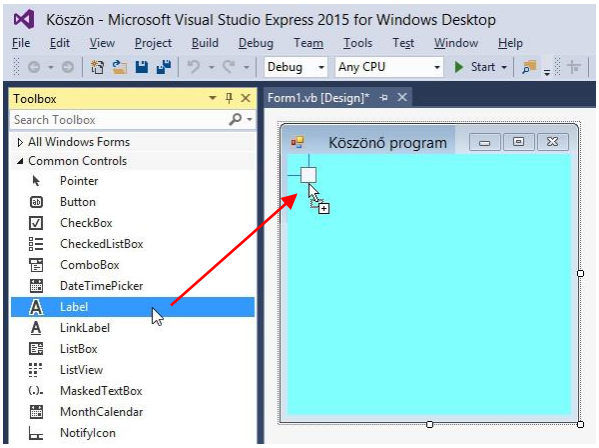
[3] A *Tools/Options* menüparancsra megnyíló *Options* párbeszédablakban szabályozhatjuk a rejtett munkaablak megnyitásának a módját. Az *Environment* csoport *Tabs and Windows* paneljén kapcsoljuk be a *Show auto-hidden windows on mouse over* jelölőnégyzetet (a panel alján találjuk)! Ennek hatására elegendő az egeret a párbeszédablak neve fölé vinni a megnyitáshoz.

[4] Nem tartjuk szerencsésnek az ablakra vonatkozó *űrlap* elnevezést, de elterjedtsége miatt mi is ezt fogjuk használni.

Objektumok az űrlapon

Programunk egy parancsgombot, egy szövegdobozt és két címkét fog tartalmazni, melyeken szöveget jelenítünk meg. Ha a felhasználó beírja a nevét a szövegdobozba, majd rákattint a parancsgombra, akkor a program köszönni fog neki.

A címkeobjektumot (*Label*) az eszközkészletben találjuk, a *Common Controls* csoportban. Fogjuk meg az egérrel a címke ikonját, és helyezzük az űrlap bal felső részére. (Hasonló eredményt érhetünk el, ha duplán kattintunk az ikonra.) Megjelenik az űrlapon a címke.



Címke elhelyezése az űrlapon

A készülő ablak

Kattintsunk a címkére, és a tulajdonságok munkaablakban módosítsuk a *Text* tulajdonságot: *Írd be a neved, majd kattints az OK-gombra!* Hosszú szövegek beírásakor kattintsunk a *Text* tulajdonság jobb szélén lévő, lefelé mutató nyílra! Így nagyobb terület áll rendelkezésünkre a gépeléshez.

Vegyük észre, hogy a *Tulajdonságok* munkaablak mindig a kijelölt objektum jellemzőit mutatja! A létrehozott objektumok között a munkaablak felső részén lévő legördülő lista segítségével is válogathatunk.

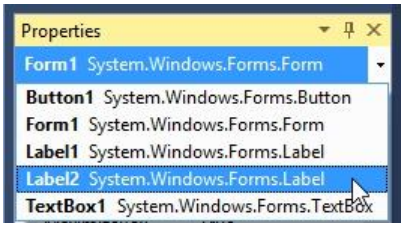
Helyezzünk el az űrlapon egy szövegdoboz (*TextBox*) objektumot, melynek szükség esetén töröljük a *Text* tulajdonságát! A szövegdoboz mellé pedig tegyünk egy parancsgomb (*Button*) objektumot! Írjuk át a parancsgomb feliratát *OK*-ra (*Text* tulajdonság)!

Végezetül tegyünk a szövegdoboz alá még egy címkeobjektumot, melynek töröljük a *Text* tulajdonságát, majd mentsük a projektet!

Futtassuk a programot! A szövegdobozba írhatunk karaktereket, a parancsgomb azonban még nem működik. El kell készítenünk az egérkattintáshoz tartozó eseménykezelő eljárást.

Mielőtt továbblépnénk, zárjuk be a programot, majd a tervezőablakban kattintsunk az egyes objektumokra! A tulajdonságok ablakban tekintsük meg azonosítóikat (*Name* tulajdonság a lista elején). A szövegdoboz a *TextBox1*, a parancsgomb a *Button1*, a két címkeobjektum pedig a *Label1*, illetve *Label2* nevet kapta. A későbbiekben ezeket az elnevezéseket célszerű olyan azonosítóval helyettesíteni, amely utal az objektum szerepére. Így sokkal könnyebben tudjuk az eseménykezelő eljárásokat áttekinteni.

Az üres címkét jelölő téglalap egy másik objektum kiválasztása esetén eltűnik az ablakból. A legegyszerűbben úgy találhatjuk meg, hogy kiválasztjuk az objektumot a tulajdonságablak legördülő listájából.



A címkeobjektum kiválasztása a Properties ablak listájában

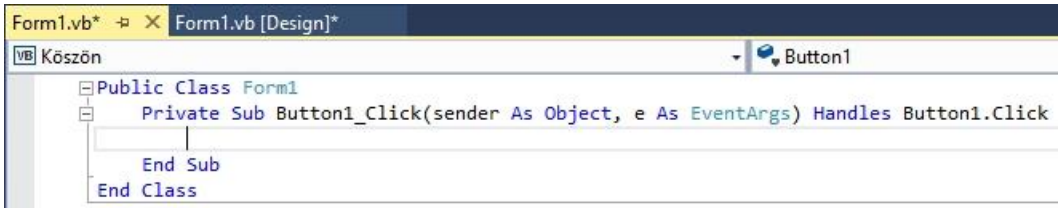
Megjegyezzük, hogy ha egymást takaró vezérlőelemeket helyezünk az űrlapra, akkor a jobb egérkattintásra megjelenő helyi menü *Bring to Front* parancsával hozhatjuk előre, illetve a *Send to Back* paranccsal küldhetjük hátra.

Eseménykezelés

A Visual Basic programok elkészítése során úgynevezett eseménykezelő eljárásokat írunk. Az eseménykezelő eljárások szabják meg, hogy a felhasználó által létrehozott események (például egérkattintás) bekövetkezéskor mit kell tenni.

A tervezőablakban kattintsunk duplán az OK-gombra, melyhez hozzárendeljük az eseménykezelő eljárást. Automatikusan megnyílik az úgynevezett kódszerkesztő ablak (*Form1.vb*). A kódszerkesztő és a tervezőablak között az ablak tetején lévő fülek segítségével válthatunk (vagy pedig a Windowsban szokásos módon, a *Ctrl+Tab* billentyűkkel). A fájlnév mellett lévő csillag azt jelzi, hogy az utolsó mentés óta módosult az ablak tartalma.

Egy eseménykezelő eljárás utasításait a *Sub … End Sub* utasítások határolják. Maga az eljárás pedig az osztálydefiníció része, amely a *Class … End Class* utasítások között helyezkedik el. Bár a szerkezet bonyolultnak tűnik, a fejlesztőrendszer nagymértékben segíti a kialakítását!



A kódszerkesztő ablak a fülekkel és a forráskód sablonjával

Megfigyelhetjük, hogy a kódszerkesztő ablakban az utasítások kulcsszavai kék színnel és nagy kezdőbetűvel jelennek meg. **A Visual Basic nem különbözteti meg a kis- és nagybetűket a forráskódban. A kódszerkesztő a kulcsszavak kezdőbetűit átírja nagybetűre még akkor is, ha kisbetűvel gépeljük be.** Ezzel megkönnyíti a hibakeresést. A behúzások alkalmazása pedig áttekinthetővé teszi a forráskódot.

A kódszerkesztő elkészíti a program keretét. Az eseménykezelő eljárás utasításait a *Sub … End Sub* kulcsszavak közé írjuk.

Az eljárás neve célszerűen az objektum (*Button1*) és az esemény (*Click*: kattintás) nevéből áll, ezeket aláhúzásjel köti össze.[5] Mint látjuk, a dupla kattintás hatására a kódszerkesztő kiválasztotta az objektumhoz kapcsolódó leggyakoribb, úgynevezett alapértelmezett eseményt (parancsgomb esetén az egérkattintást). Ezt természetesen szükség esetén módosíthatjuk. Az objektumhoz kapcsolódó eseményeket az ablak tetején, a jobb oldalon található legördülő lista tartalmazza.

> Az eseménykezelő eljárás két paraméterrel rendelkezik. Az első paraméter (*sender*) segítségével az eljárást meghívó objektumot, a második paraméterrel (*e*) pedig az esemény tulajdonságait érhetjük el. A paramétereket most nem fogjuk felhasználni. A sor végén álló *Handles* (kezeli) kulcsszó után soroljuk fel azokat az objektumokat és eseményeket, amelyekre az eseménykezelő eljárás vonatkozik. Egy eljárás több objektum több eseményét is kezelheti. A listában az objektum és az esemény közé pontot teszünk.
> A *Sub* előtt álló *Private* kulcsszó azt jelzi, hogy eljárásunk egy másik osztályból/modulból nem hívható meg.
> A kódszerkesztő ablakban megfigyelhetjük, hogy az eseménykezelő eljárás a *Form1* osztály egy eljárása. A *Class* után áll az osztály neve. A *Public* kulcsszó arra utal, hogy osztályunkra más kódfájlokból is hivatkozhatunk.

Ha egy projekt megnyitásakor nem látjuk a kódszerkesztő ablakot, akkor a megoldástallózóban kattintsunk a jobb egérgombbal az űrlapra (*Form1*) és válasszuk a *View Code* parancsot! Szükség esetén ugyanígy (vagy dupla kattintással) nyithatjuk meg az űrlapot a tervező nézetben (*View Designer*).

A szöveg másolása a címkére

Mint látjuk, a kódszerkesztő automatikusan elkészíti helyettünk programunk keretét. Nekünk csak az eseményt kezelő utasításokat kell begépelnünk. Feladatunk, hogy a *TextBox1* szövegdobozba írt szöveget az üdvözlettel kiegészítve átmásoljuk a *Label2* címkére. A szövegdoboz szövegét az objektum *Text* tulajdonsága adja meg. Egy objektum tulajdonságára az objektum és a tulajdonság nevével hivatkozunk, amiket ponttal választunk el egymástól:

objektumnév.tulajdonságnév

Például:

```
TextBox1.Text
```

[5] Az eljárást nem kötelező így elnevezni. Az eseményt és az objektumot az eljárásfej végén (a zárójeles paraméterlista után) álló *Handles Button1.Click* kapcsolja az eljáráshoz.

A *Label2* objektum *Text* tulajdonságát a következő utasítással módosítjuk:

```
Label2.Text = TextBox1.Text
```

Ennek hatására a szövegdoboz szövege átkerül a címkeobjektumba.

Gépeljük be a fenti utasítást a *Sub … End Sub* sorok közé, majd futtassuk a programot! Ha beírjuk a nevünket a szövegdobozba, és rákattintunk a parancsgombra, a szövegdoboz szövege megjelenik a címkén. Működik a programunk! ☺

Az intelligens súgó

A Visual Studio intelligens súgója nagymértékben megkönnyíti a forráskód begépelését. A súgó kilistázza szóba jöhető elemeket. A listából az egérrel vagy a kurzormozgató billentyűkkel választhatunk. Ha kiválasztottunk egy elemet, akkor az utána következő karakter gépelésével folytathatjuk a munkát. A billentyű lenyomásakor a súgó beilleszti a karakter elé a kiválasztott elemet a forráskódba. Ugyanezt az eredményt érjük el, ha a következő karakter helyett a tabulátor billentyűt nyomjuk le, vagy duplán kattintunk az elemre. Ezt a módszert akkor szoktuk alkalmazni, ha már nem folytatjuk a gépelést.

Töröljük az általunk begépelt sort, majd ismételjük meg a beírást az intelligens súgó felhasználásával!

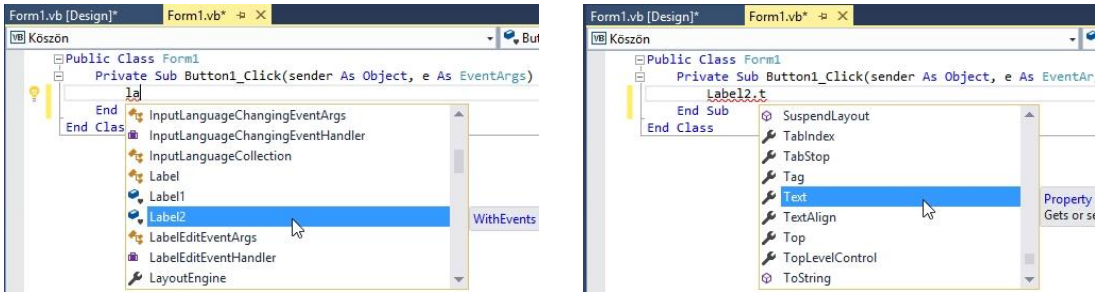
Kezdjük el gépelni a fenti utasítást (kisbetűvel kezdve). Az első karakter (*l* betű) beírásakor megjelenik a lista. Kijelzi az összes lehetséges folytatást. Írjuk be az *a* betűt is! Az egérrel vagy a kurzormozgató billentyűvel válasszuk ki a listából a *Label2*-t, majd nyomjuk le a forráskódban következő pontot! Megjelenik az objektum tulajdonságainak és metódusainak a listája, ahol kijelölhetjük a *Text* tulajdonságot.

A *Text* kiválasztása után a következő karakter, az egyenlőségjel begépelésével folytassuk a forráskód szerkesztését! A kódszerkesztő automatikusan elhelyezi a forráskódban a *Text* szót, majd megjelenik egy újabb lista. Ezzel ne foglalkozzunk, hanem gépeljük be az alábbi folytatást:

```
Label2.Text = "Szia " & TextBox1.Text & "!"
```

Közben az intelligens súgó folyamatosan megjeleníti az adott utasításra vonatkozó rövid, angol nyelvű magyarázatot. Ügyeljünk arra, hogy a kezdő idézőjel beírásakor megjelenik a párja is, ezt nem kell begépelnünk! Az & jel (*Alt Gr+C*) az Excelhez hasonlóan a szövegek összefűzését végzi.

Megjegyezzük, hogy az intelligens súgót utólag a *Ctrl + szóköz* billentyűparanccsal hívhatjuk elő.



Az intelligens súgó a forráskód begépelésénél

Az Enter lenyomásával lépjünk a következő sorba! A létrejövő új sor behúzása megfelel az előző sornak. A kódszerkesztő elvégzi helyettünk a behúzás megfelelő kialakítását. Szóközök beillesztésével pedig tagolja, áttekinthetővé teszi az utasításokat.

Ha nincs szükség újabb sor begépelésére, akkor az Enter helyett a Tabulátor billentyűvel is befejezhetjük az utasítás bevitelét.

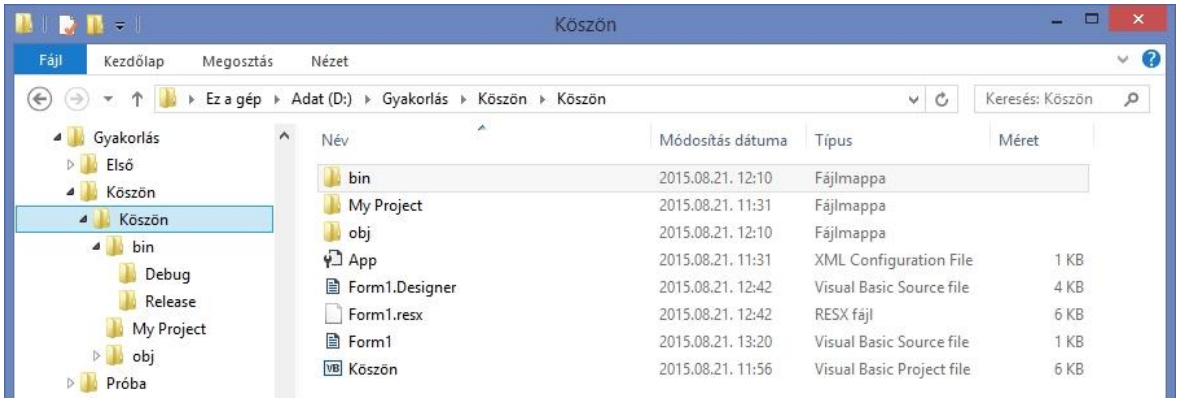
Mentsük a projektet, majd futtassuk a programot! Próbáljuk ki a működését! Az üdvözlő szöveg megjelenése után írjunk be egy másik nevet, majd ismét kattintsunk az *OK*-gombra!

A Visual Basic korszerű vizuális fejlesztőrendszerének a segítségével gyorsan és egyszerűen tervezhetjük meg az ablakokat és írhatjuk meg eseményvezérelt programjainkat.

A program futtatása

A projektmappa szerkezete

A megoldás (*Solution*) és a projekt az elkészítésénél megadott mappába kerül. Nyissuk meg a mappát a fájlkezelővel, és tekintsük át a tartalmát!

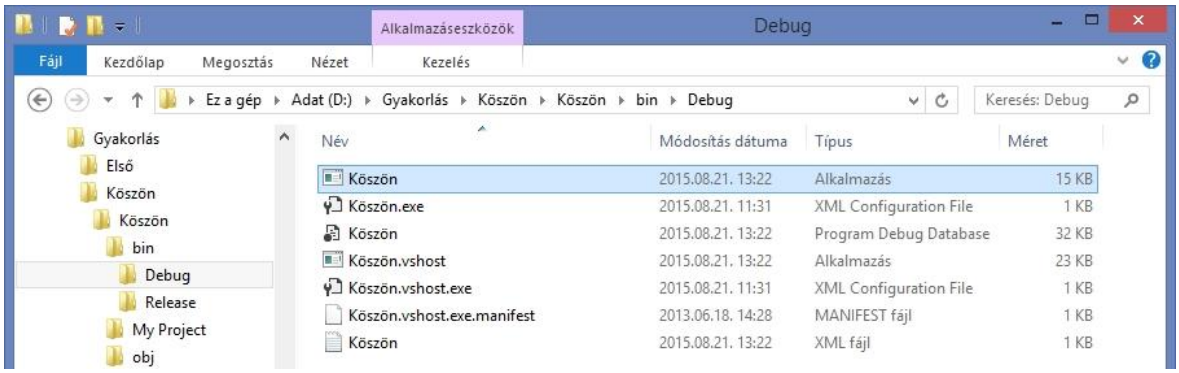


A megoldás és a projekt mappája

A megoldás mappájában (*Köszön*) találjuk a (most ugyanolyan nevű) projektmappát.

A projektmappában van (többek között) a *Form1.Designer.vb* fájl, amely az űrlap objektumait és azok tulajdonságait tárolja. A fájlt a fejlesztőrendszer a grafikus felületen elvégzett beállításoknak megfelelően automatikusan elkészíti. A forráskódot a *Form1.vb* fájl tartalmazza. Ezek a fájlok egy szövegszerkesztővel, akár a Jegyzettömbbel is megnyithatók.

A fejlesztőrendszertől függetlenül futtatható programfájl eléréséhez lépjünk be a *bin*/*Debug* mappába! A programfájl neve megegyezik a projekt nevével (például *Köszön*). Dupla kattintással indíthatjuk el.



A futtatható fájl a Debug mappában

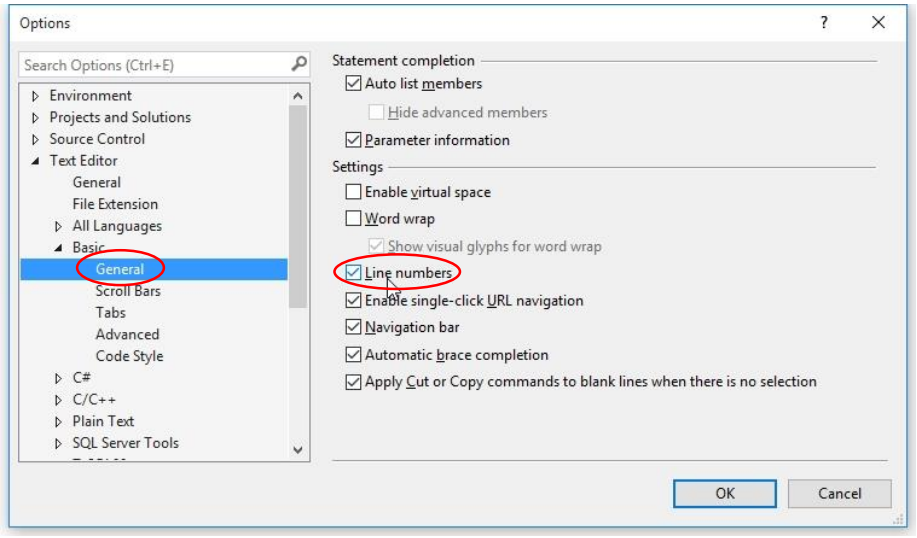
Ügyeljünk arra, hogy az *Alkalmazás* típusú fájlt ne keverjük össze a többivel, például a *Köszön.exe XML Configuration File* típusú fájllal!

A fejlesztőrendszer beállításai

A következő táblázatban áttekintjük a fejlesztőrendszer azon beállításait, amelyek nagymértékben megkönnyítik a munkánkat. A beállításokat a *Tools*/*Options* parancs segítségével érjük el.

A táblázatban az *Options* párbeszédablak listájában szereplő csoportosítást használjuk. A ☺ jel utal arra, hogy egy beállítást célszerű alkalmazni/bekapcsolni, a ☹ jel pedig a kikapcsolást javasolja.

Environment		
Tabs and Windows	*Tool Windows/Show auto-hidden windows on mouse over* Megjeleníti a rejtett munkaablakot, ha a név fölé visszük az egeret.	☺
Text Editor		
General	*Display/Highlight current line* Az aktuális sor bekeretezése.	☹
Basic/General	*Line numbers* **A sorok számozása, feltétlenül kapcsoljuk be!**	☺
	Automatic brace competition Zárójelpárok automatikus megjelenítése.	☹



A sorok számozásának bekapcsolása az Options párbeszédablakban

A munkaablakok kezelése

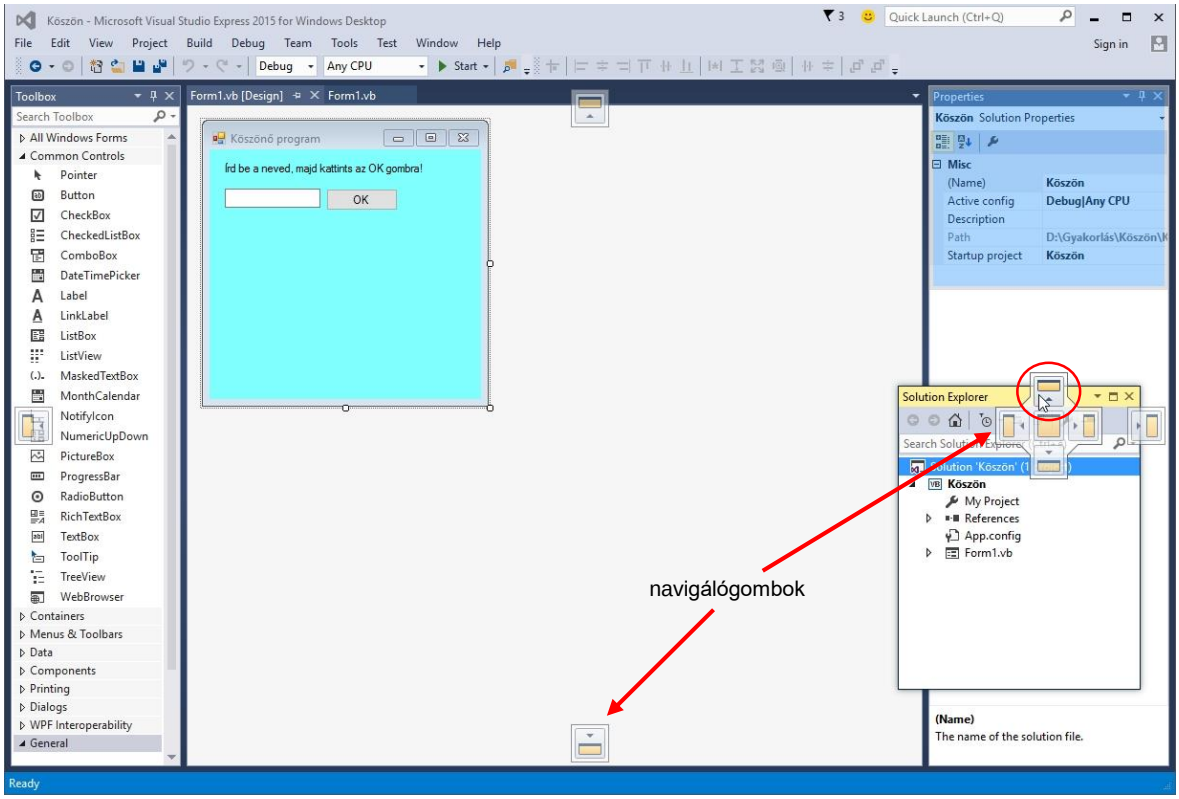
A szükséges munkaablakok a *View* menü segítésével vagy az ott látható billentyűparancsokkal jeleníthetők meg. Javasoljuk a *Toolbox*, a *Solution Explorer* és a *Properties* munkaablak megnyitását. Ügyeljünk arra, hogy az űrlapra helyezhető objektumok (*Button, Textbox* stb.) csak tervező nézetben jelennek meg az eszközkészletben (*Toobox*)!

A lebegő ablakok elrendezése

A munkaablakok automatikus elrejtésének beállítását a 4. oldalon már bemutattuk.

A munkaablakok alapértelmezés szerint a programablak széléhez vannak rögzítve. Címsoruk mozgatásával azonban szabadon elhelyezhetők a Visual Studio ablakában.

Az ismételt rögzítéshez fogjuk meg a munkaablak címsorát, és mozgassuk a képernyőn. Az így megjelenő navigálógombok alapján a kiválasztott helyre illeszthetjük be. A *Solution Explorer*-t és a *Properties* munkaablakot először mozgassuk a Visual Studio ablakának jobb széléhez, majd a megjelenő újabb navigálógombok segítségével állíthatjuk vissza az eredeti helyét.



A Solution Explorer visszatétele a szokásos helyére