

Bátfai Barnabás

Makróhasználat Excelben

Bártfai Barnabás

Makróhasználat Excelben

BBS-INFO Kiadó, 2010., 2016.

Bővített kiadás

© Bártfai Barnabás, 2010., 2016.

Minden jog fenntartva! A könyv vagy annak oldalainak másolása, sokszorosítása csak a szerző írásbeli hozzájárulásával történhet.

A betűtípus elnevezések, a Microsoft, a Windows, az Excel bejegyzett védjegyek.
A Microsoft Szoftver Információ telefonszáma: (06-1) 267-46-36

A könyv nagyobb mennyiségben megrendelhető a kiadónál:
BBS-INFO Kiadó, 1630 Bp. Pf. 21. Tel.: 407-17-07

A könyv megírásakor a szerző és a kiadó a lehető legnagyobb gondossággal járt el. Ennek ellenére, mint minden könyvben, ebben is előfordulhatnak hibák. Az ezen hibákból eredő esetleges károkért sem a szerző, sem a kiadó semmiféle felelőséggel nem tartozik, de a kiadó szívesen fogadja, ha ezen hibákra felhívják figyelmét.

Papírkönyv ISBN 978-963-9425-40-8

E-book ISBN 978-963-9425-50-7

Kiadja a BBS-INFO Kft.

1630 Budapest, Pf. 21.

Felelős kiadó: a BBS-INFO Kft. ügyvezetője

Nyomdai Munkák: Biró Family Nyomda, Budapest

Felelős vezető: Biró Krisztián

TARTALOMJEGYZÉK

Bevezető	7
1. A makrókról	9
2. Ami a makróhasználathoz szükséges	11
2.1. Vezérlők	12
2.1.1. Űrlap-vezérlőelemek és ActiveX vezérlők	14
2.1.2. Léptetőnyilak	15
2.1.3. Lista	16
2.1.4. Legördülő lista	17
2.1.5. Jelölőnégyzetek	18
2.1.6. Választógombok	18
2.1.7. Gördítősáv, csúszka	19
2.1.8. Nyomógombok	21
2.1.9. Feliratok és csoportpanelek	22
2.2. Hivatkozások	22
2.3. Lapvédelem és cellazárolás	22
2.4. Adatérvényesítés	25
2.5. Tulajdonságpanelek	25
2.6. Védelem a rosszindulatú makrók ellen	26
2.7. Fájltypusok	27
3. Hogyan rendezzük el a táblázatunkat?	28
4. Makrók rögzítése és futtatása	29
4.1. Rögzített makrók használata	30
4.1.1. Makrók rögzítése és indítása 2003-as verzióig	30
4.1.2. Makrók rögzítése 2007-es verziótól	30
4.1.3. Makrók indítása	31
4.1.4. Gyakoroljunk!	31
4.1.5. Rendeljünk gyorsbillentyűt makróinkhoz!	32
4.1.6. Gyakoroljunk egy komplexebb feladat segítségével!	32
4.1.7. Amire figyeljünk	34
4.1.8. Mit tehetünk, ha elrontottuk a rögzítést?	34
4.1.9. Ha hiba után nem indul a makró	35
4.2. Makrók szerkesztése és a Visual Basic	35
4.2.1. Hol vannak a makrók?	37
4.2.2. A Visual Basic súgója	38
4.2.3. A makrók felépítése	38
4.2.4. Hibakezelés	40
4.2.5. Utasítások tesztelése	41
4.2.6. Objektorientáltság	42

5. Makróprogramozás	45
5.1. Alapvető programozási lehetőségek	45
5.1.1. Cellaazonosítások	45
5.1.2. Értékadás.....	47
5.1.3. Függvények a cellákban.....	48
5.1.4. Változók	51
5.1.5. Változók adattípusai.....	52
5.1.6. Egyszerűbb számítások.....	53
5.1.7. Hatókörök	55
5.1.8. Üzenőablakok.....	56
5.1.9. Adatbeviteli ablak	57
5.2. Feltételvizsgálatok	58
5.3. Ciklusok	62
5.4. Tömbök	69
5.5. Alprogramok	72
5.6. Makróból elérhető Excel funkciók	73
5.6.1. Munkalapműveletek.....	73
5.6.2. Formázások.....	73
5.6.3. Sor és oszlopátméretezések	76
5.6.4. Beszúrás, törlés.....	76
5.7. Védelem	77
5.8. Nyomtatás.....	79
5.9. Fájl műveletek	79
5.10. További lehetőségek és példák.....	82
5.11. Ha már végképp nem megy máshogy... ..	83
6. Új függvények készítése	85
7. Eseménykezelés	86
8. Párbeszédpanelek használata	90
9. Programozási tanácsok	102
9.1. A felhasználó mindenre képes.....	102
9.2. Kommentezés	103
9.3. Rövidítések	103
9.4. Statikus adatok	103
9.5. Strukturáltság, tagolás	103
9.6. Ismeretlen adatok	104
9.7. Az adatok biztonsága	104
9.8. Problémák a ciklusok használata során	104
9.9. A makró nem táblázat	105
9.10. Mentések	105
10. Példák és lehetőségek	106
10.1. Árkalkuláció	106
10.2. Munkabeosztás tervező.....	111
10.3. Életjáték.....	116
10.4. Számológép Excel táblázatban párbeszédpanellel	120
10.5. Nyilvántartó	126
10.6. Új táblázat generálása ügyféladatokból.....	142
10.7. Tréfás munkafüzet.....	147
10.8. Prímszámkészítő függvény	148
11. Tárgymutató	149

Bevezető

Jelen könyv az Excel azon rejtett képességeit ismerteti, amelyek segítségével az Excelből olyan programot varázsolhatunk, amely a saját igényeinkhez a leginkább alkalmazkodik. Az Excel sokkal többet tud annál, mint amit első látásra mutat magából. Ha bekapcsolunk bizonyos rejtett eszköztárat, szalagokat, használjuk az azon lévő funkciókat, netán belevágunk a táblázatunk Visual Basic kóddal történő kiegészítésébe, szinte csodákat tehetünk.

A makrókról sokan hallottak már, de azt valami misztikus dolognak kezelve csak kevesen mertek belevágni a használatukba is. Pedig egyáltalán nem ördögtől való, s még programozni sem szükséges tudni hozzá, bár a komolyabb lehetőségek megvalósítása már lényegében programozásnak minősül.

A makrókat többféle célra is használhatjuk. Az egyik, ismertebb módszer a feladatok automatizálása, amikor rögzítünk egy tevékenységsorozatot, majd azt a későbbiekben – akár más adatokkal – ismételten vagy többszörösen végrehajtjuk. Ez a módszer a munkánkat könnyíti meg, hiszen ha a tevékenységsorozat összetett és többször is elvégzendő, úgy elég azt csak egyszer megtanítani a gépnek, ami aztán már később bármikor ismét végre tudja azt hajtani.

A másik lehetőség a makrók nyomógombokhoz, eseményekhez való rendelése. Ezzel a módszerrel az Excelből tetszőleges alkalmazást varázsolhatunk, hiszen a vezérlőelemekkel és különféle tevékenységeket végrehajtó gombokkal olyan paneleket hozhatunk létre, amelyeket a többi Windows alatti alkalmazásnál is megszoktunk. A tevékenységek előre megadhatók, végrehajtódnak pedig akkor fognak, ha a későbbi használó rákattint az adott gombra.

A mindezen funkciók mögött álló makrók pedig könnyedén rögzíthetők, vagy ha kissé komplikálni akarjuk a dolgot, úgy leprogramozhatók. Mindegy ugyanis, hogy rögzítünk, vagy programozunk, az eredmény ugyanaz, egy kódsor a Visual Basicben. (A

rögzítés során is egy kódsor jön létre, ami megfelel az általunk végzett tevékenységeknek.)

A könyv feltételezi, hogy a tisztelt olvasó már rendelkezik az Excel program középszintű ismeretével, azaz tudja azt használni, kellően ismeri a formázási lehetőségeket, ismeri a függvényeinek nagy részét, különös tekintettel a kereső, összegző, adatbázis, és feltételes függvényekre, továbbá nem okoz gondot az alakzatok kezelése sem.

Ha netán ezen ismeretek még hiányosak, úgy ajánljuk, hogy ismétlje át ezeket, például a szerző Táblázatkezelés c. könyvének segítségével.

A könyvben leírtak alapját az Excel 2007, 2010 és 2013 verziói adják, kezelést tekintve főként e verziókban szükséges lépéseket írjuk le. Ez azonban nem jelenti azt, hogy a leírtak nem használhatók más korábbi verziókban, hiszen néhány helyen ejtünk is szót ezekről az eltérésekről is.

A könyvben lévő példák letölthetők a kiadó www.bbs.hu címen található weblapjáról a **Letöltések** meüpont alatt, így aki nem akar vesződni a begépeléssel, az ezt így megspórolhatja.

1. A makrókról

Az Office programcsomagban lehetőség van arra, hogy a dokumentumba (Word szöveg, Excel táblázat, stb.) ún. makrókat építsünk, amellyel bizonyos feladatokat automatikussá tehetünk. Ezt megtehetjük egy gyakran végzett művelet automatizálása miatt, vagy egy gombhoz, legördülő listához, vagy egyéb űrlapelemhez rendelt funkció végrehajtása miatt is. A makró lényegében nem más, mint egy Visual Basic modulban tárolt program, vagy parancs- és függvénysorozat, előre rögzített tevékenységek sora. A makrók a dokumentumhoz kapcsolódva tárolódnak, azok részét képezik (ezért vírusok esetén veszélyt is jelenthetnek).

A makrók készítésének több módja is van, de a program kezelésének ismerete alapfeltétel. Egy makrót írhatunk Visual Basic nyelven is, de ehhez ismerni kell a nyelv szintaxisát. Az egyszerűbb megoldás a makró rögzítése, amikor lényegében felvesszünk egy tevékenységsorozatot. Ekkor az Office lépésenként eltárolja az egyes parancsok végrehajtásának adatait. (A lagpraktikusabb a kettő kombinálása.)

A makrók alkalmazásának igen hasznos és látványos módja, amikor például a Word dokumentumban vagy az Excel táblázatban létrehozunk űrlapelemeket, gombokat, stb., majd ezekhez makrókat rendelünk. Például készítünk egy **Készítés** feliratú gombot, amely beírja az adatot a megfelelő helyre, kinyomtat egy munkalapot, majd elmenti a táblázatot.

Az Excelből a makrók segítségével igen könnyedén látványos alkalmazásokat készíthetünk, amelyeket komolyabb célokra is felhasználhatunk, így akár készíthetünk egy komolyabb automatizált nyilvántartást, amely még a kimutatások külön fájlban való elkészítését is megoldja helyettünk.

Ha valamit lehet automatizálni, akkor azt a makrókkal tudjuk megtenni. Jól jönnek a makrók lehetőségei tehát minden olyan esetben, amikor valamilyen műveletet várhatóan a későbbiekben is

jópárszor el kell majd végezni, de egy kevésbé bennfentesnek szánt nyilvántartó elkészítésére, vagy szimulációra is tökéletes. Olyan esetekben tehát, ahol ismételni kell bizonyos lépéseket, esetleg azokat valamitől függően kell végrehajtani, s mindezt lehet automatizálni, használhatjuk a makrókat. Összetettebb Excel alkalmazások esetén is nagy valószínűséggel jól jönnek a makrók lehetőségei, de a táblázatok, szöveges dokumentumok felhasználóbarátabbá tételében is hasznos segítség lehet.

Nem érdemes azonban a makrókra, mint a mindenre jó, tökéletes megoldásra tekintenünk. Nem használhatjuk a makrókat, ha nem tudjuk magunk sem egyértelműen definiálni, hogy milyen esetben mit kell elvégezni. Nem érdemes makrót készíteni a ritkán használt feladatokhoz sem, és azt sem árt szem előtt tartani, hogy bizonyos esetekben sokszor nem elég az Excel teljesítménye, egy más adatbázis-kezelő alkalmazás ugyanazt a problémát sokkal gyorsabban megoldhatja, bár kétségtelen, hogy azok kezelése, programozása nem olyan egyszerű, mint az Excel makróinak kezelése.

Ne felejtjük el továbbá azt sem, hogy bár a programunkat a Visual Basic használatával készítjük, azok nem lesznek Visual Basic programok, csupán az Excel makróprogramjai, mivel nem működnének az Excel nélkül, utasításaik az Excelre épülnek, és nem is lehet őket befordítani sem.

Pontosan a fentiek miatt jelen könyvben sem a Visual Basic szigorú szabályai szerint fogjuk ismertetni a nyelv lehetőségeit, hanem az egyszerűséget és az érthetőséget szem előtt tartva mutatjuk be az utasításokat. Sok utasítást ezért minden bizonnyal a bemutatottnál komplexebben és szabatosabban is lehetne használni, melyekre az Excel makróprogramozásánál viszont többnyire nincs szükség. A Visual Basic egy objektumorientált programozási nyelv, amely objektumkezelési metódusa sokkal nagyobb lehetőségeket rejt, mint amit az Excelből érdemes volna kihasználni. Az Excel makróprogramozása azonban jó alapot nyújthat a Visual Basic önálló használatához is, de ehhez jelentősen túl kell lépni a jelen könyvben ismertetett lehetőségeken.

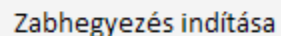
Nagyon fontos megemlíteni már előljáróban is, hogy a makró-funkciók nem vonhatók vissza! Ha tehát egy tevékenységet makróból végzünk el, ami hatással van a táblázatra, akkor hiába kattintunk a visszavonás gombra, az hatástalan marad, mivel a változtatást a makróprogram végezte és nem az Excel. (Praktikus megoldás egy biztonsági mentés készítése más néven.)

2. Ami a **makróhasználatához szükséges**

Ebben a fejezetben megismerjük az Excel azon funkcióit, melyek bár közvetlenül nem tartoznak a makrókhoz, viszont az általános lehetőségeknél kissé bonyolultabbak, vagy ritkábban használtak, de a makrók használatához sokszor nélkülözhetetlenek.

Azon túl, hogy ismernünk kell az Excel függvényeit, illetve azok lehetőségeit, van még pár hasznos dolog az Excelben, ami nem kerülheti el a figyelmünket, ha hatékonyan akarunk dolgozni a makrókkal.

Bár a makrók használatához elegendő csupán azok menüből, szalagról történő meghívása is, ez nem nevezhető esztétikus megoldásnak egy komolyabb Excel táblázat esetében. A legismertebb megoldás, ha egy nyomógombra kattintva indul el makró, úgy ahogy azt a rendes alkalmazásokban is már megszokhattuk.



De hogyan tudunk nyomógombot készíteni Excelben? Sokkal egyszerűbben, mint gondolnánk. Az Excel ugyanis lehetőséget biztosít az olyan ismert vezérlőelemek használatára a táblázaton belül, mint például a nyomógomb, léptetőnyíl, legördülő lista, választógomb, kijelölőnégyzet, stb. Persze ezeknek a vezérlőknek a beillesztéséhez szükséges eszköztár vagy szalag alapesetben nem látszik, mivel ezek magasabb szintű funkciók, így nem akarták a fejlesztők, hogy az átlagfelhasználót megzavarja. Sajnos a 2007-es és 2010-es verzióban a vezérlők és makrók használata még annyira sem elérhető, mint a korábbi változatokban volt, mert a funkciókat igencsak elrejtették, így alapesetben semelyik szalagon sem található meg.

Nekünk viszont a makróhasználatához létszükséglet, ezért kapcsoljuk is be azt minél hamarabb.

2003-as, illetve korábbi verziókban a vezérlők eszköztárat a **Nézet** menü **Eszköztárak** almenü **Vezérlők eszközkészlete** pontjával tudjuk bekapcsolni.

Office 2007-ben, 2010-ben és 2013-ban a Fejlesztőeszközök eszköztárat kell bekapcsolnunk 2007-ben az Office menü Excel beállítási gombján kattintva a Népszerű elemek funkció Fejlesztőeszközök lap megjelenítése a szalagon felirat előtti négyzet kipipálásával, 2010-ben és 2013-ban pedig a Fájll menü Beállítások pontjának Menüszalag testreszabása funkcióján, majd a jobboldali listában a Fejlesztőeszközök lap előtti négyzet kipipálásával.

Ha ezzel megvolnánk, akkor már lehetőségünk van ezeket az elemeket a táblázatunkban alkalmazni.

A vezérlők mellett nagyban segítheti munkánkat a hivatkozások használata is, hiszen ezekkel olyan gombokat, füleket, elemeket hozhatunk létre, amelyre kattintva könnyedén ugrálhatunk a táblázat különböző pontjai közt.

Fontos, hogy tisztában legyünk a lapok elrejtésének módjával, illetve a lapvédelem és cellazárolás lehetőségeivel is, mert ezek a lehetőségek nyújtanak majd kellő biztonságot ahhoz, hogy mások ne tudják könnyedén tönkretenni a munkánkat.

A fentiekén túl hasznos, ha ismerjük az adatérvényesítés nyújtotta lehetőségeket is, mert ez esetben nem a makróban kell kiszűrni az esetlegesen téves adatokat, amely a makró működésében problémákat okozhat.

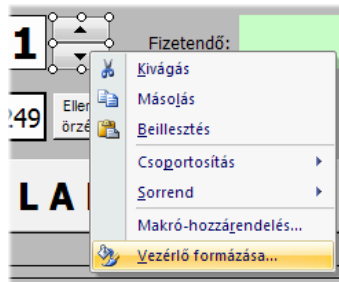
Mivel a makróhasználat biztonsági kockázatokat is rejt, ezért a használat előtt meg kell ismerkednünk azok engedélyezésének lehetőségeivel is, különben nem fognak működni makróink mások számítógépén.

2.1. Vezérlők

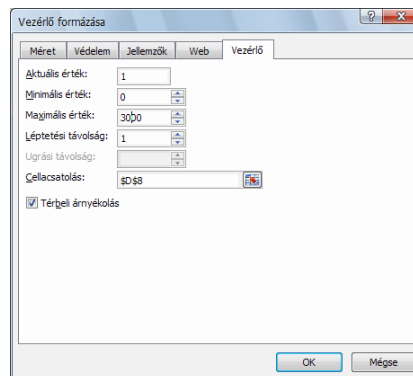
Egy űrlapvezérlő táblázatra való beszúrásához gördítsük le a Fejlesztőeszközök szalag Vezérlők mezőjében lévő Beszúrás ikont, kattintsunk annak a vezérlőelemnek az ikonjára, amit használni szeretnénk, majd rajzoljuk azt meg a táblázat kívánt helyén. (2003-as verzióban a vezérlők eszköztár megfelelő elemét kell választani, majd szintén a kívánt helyen megrajzolni a vezérlőt.) A vezérlő formázása és felparaméterezése a jobb egérgomb használatával történik, de a űrlap-vezérlők és ActiveX-vezérlők kezelése jelentősen eltér egymástól.



Az űrlapvezérlő konfigurálásához (azaz olyan adatok megadásához, hogy melyik cellára vonatkozzon a vezérlő, milyen értékeket vehet fel stb.) kattintunk a vezérlőre a jobb egérgombbal, majd a helyi menüből választjuk a **Vezérlő formázása** pontot. Ez nagyon fontos lépés, ezen adatok megadása nélkül a vezérlőnk nem lesz több, mint egy dísz.



A vezérlők egy-egy cellához kapcsolódnak, így ha a vezérlőt használjuk, az a kapcsolt cella tartalmát fogja módosítani. Például egy léptetőnyíl esetében megadhatjuk az induló, a minimális és maximális értéket, a lépésközt, illetve a cellacsatolást. Ez utóbbi nagyon fontos, hiszen a **Cellacsatolás** sornál adhatjuk meg azt a cellát, ami módosulni fog a vezérlő használata által. Más vezérlőknél más adatok megadása szükséges, például egy legördülő lista esetén a bemeneti tartomány, a cellacsatolás és a lenyíló sorok számának megadása kell.



Nyomógomb esetén kissé más a helyzet, az ugyanis egy makró indítását eredményezheti, az elvégzendő műveleteket tehát a makrónak kell tartalmaznia. (Makró indítását természetesen más vezérlő is kiválthat, de ez nem jellemző.)

Vezérlő utólagos áthelyezéséhez, vagy átméretezéséhez kattintunk rá a jobb egérgombbal. Ezt követően a vezérlő húzással áthelyezhető, illetve a szélein vagy sarkain lévő jelek húzásával átméretezhető.

Az ActiveX vezérlők szerkesztése során bekapcsolásra kerül a tervező mód. Ezért ha végeztünk a vezérlők megrajzolásával és felparaméterezésével, lépünk ki a tervező módból a **Fejlesztőeszközök** szalag **Vezérlők** mezőjében lévő **Tervező mód** ikon kikapcsolásával.



Egyes vezérlők csak akkor működnek normálisan, ha az oldalon lapvédelmet is használunk.

Fontos, hogy a csatolt cellának nem szabad zároltnak lennie, mert különben a vezérlő nem tudja módosítani annak tartalmát.

A kinézet szerkesztésekor a vezérlők használata esetén célszerű, ha a vezérlőket a csatolt cella mellé, vagy fölé helyezzük. A legpraktikusabb megoldás az, ha a legördülő lista vezérlő eltakarja a kapcsolt cellát, így érve el azt, hogy annak tartalma ne is látszódjon. A léptetőelemeket a kapcsolt cella mellé célszerű helyezni úgy, hogy a sormagasságot nagyobbra állítjuk. Egyes vezérlőknél (pl. választógombok) praktikus megoldás, ha a csatolt cella szövegszínét a háttérrel azonosra állítjuk, így nem lesz zavaró annak numerikus tartalma.

Mindemellett célszerű kikapcsolni a rácsvonalakat és a fejléceket, a panelek megjelenését pedig a hátterek színezésével kialakítani.

2.1.1. **Űrlap-vezérlőelemek és ActiveX vezérlők**

Az Excel táblázatokban vezérlő készítésénél ugyanazon vezérlőket megtaláljuk az Űrlap-vezérlőelemek és ActiveX vezérlők között is. Jogosan merülhet fel a kérdés, hogy mi a különbség a kettő között, és mikor érdemes az egyiket és mikor a másikat használni.

Nos a felhasználó gyakran csak annyit lát, hogy a hagyományos űrlap vezérlőelemeken szebben jelennek meg a betűk, míg az ActiveX vezérlőkön töredezetten. A különbség azonban nem ez. A hagyományos űrlap vezérlőelemek használata egyszerűbb, konfigurálása kézenfekvőbb, viszont a használhatóságuk köre korlátozottabb. Az Excel táblázataiban többnyire bőven elegendő ezeket használni, viszont ha már eseménykezelést is végzünk, vagy párbeszédpaneleket is készítünk, úgy az ActiveX vezérlőkre lesz szükségünk. Az ActiveX vezérlő megváltozása például képes elindítani egy makrót is, de alkalmazásukkal olyan külső funkciók is elérhetők, amelyek nem az Excel lehetőségei, hanem külső programok által használt vezérléseket valósítanak meg, így ezen speciális vezérlések ezáltal Excelből is elérhetővé válnak. Az ActiveX vezérlők lényegében olyan kis bővítmények, melyek az Excel lehetőségein kívül másutt, akár az interneten is használhatók, segítséget nyújthatnak különböző feladatok elvégzéséhez. Ennek megfelelően tehát ha csupán egy egyszerűbb funkciót kívánunk készíteni, ahol a vezérlőket az Excel táblázataiba integráljuk, úgy jobban járunk a hagyományos vezérlőkkel, mivel azokat egyszerűbben konfigurálhatjuk. Ha azonban ennél többre vágyunk, úgy jöhetnek az ActiveX vezérlők.

A legszembetűnőbb eltéréseket, a két vezérlőtípus tulajdonság-lapján táthatjuk. A hagyományos vezérlőknél a helyi menü **Vezérlő formázása** pontjának **Vezérlő** fülénél állíthatjuk be a legfontosabb jellemzőket, az ActiveX vezérlők esetében viszont a helyi menü

Tulajdonságok pontját használhatjuk. Ez utóbbi lényegesen összetettebb és magyar nyelvű Excel esetén is angolul jelenik meg. Ebből adódóan sokkal több paraméter állítható, így jobban konfigurálható, mint a hagyományos vezérlő, így számos szín, keret, megjelenítés, betűtípus, elrejtés, méret, stb. adat is meghatározható. Pl.:

LinkedCell: melyik cellához történjen a csatolás, azaz hogy hova kerüljön az eredmény.

ListFillRange: azt határozza meg, hogy a beviteli listában szereplő adatok honnan származzanak.

ListRow: azt adja meg, hogy legördítéskor hány sor látszódjon.

(Name): ez alapján tudunk hivatkozni az objektumra.

Fontos, hogy az ActiveX vezérlők szerkesztéséhez be, használatukhoz a tervező módot ki kell kapcsolni.

Néhány vezérlő esetében azonban egyéb jelentős, működésbeli eltérések is vannak a hagyományos és ActiveX vezérlők közt. Az egyik ilyen például a legördülő listán kiválasztott sor által szolgáltatott eredmény. A hagyományos űrlapvezérlő azt adja meg, hogy a lista hányadik elemét választottuk ki, így a cellacsatolásként megadott eredménycellába egy szám kerül. Az ugyanilyen ActiveX vezérlő viszont magát a kiválasztott listaelemet írja a csatolt cellába. Ennek megfelelően az eredmény feldolgozása is merőben más műveleteket fog igényelni.

2.1.2. Léptetőnyilak

Praktikus megoldás, ha a numerikus adatot nem csak begépeléssel tudunk rögzíteni.

Ha a beviteli mező mellé kis le és felfelé mutató nyilat helyezünk, használhatjuk ezeket is oly módon, hogy a lefelé mutató nyíllal csökkenthetjük, míg a felfelé mutató nyíllal növelhetjük a beviteli mezőben található értéket. A vezérlő formázásánál megadható a lépésköz és a mező által felvehető legkisebb és legnagyobb érték is, így ezen határokon túl már hiába nyomkodjuk a gombokat, nem fog az érték változni. A cellacsatolás sorban annak a cellának az azonosítója áll, amely értékét a vezérlő módosítja.

