

Bemutatkozik a Tkinter

Grafikus felületek készítése Pythonból gyorsan és egyszerűen.

A Tkinter segítségével Pythonban írt alkalmazásainknak könnyedén varázsolhatunk egyszerűen kezelhető grafikus felületet, mely – a Pythonnak köszönhetően – nemcsak Linux alatt, de Windowson vagy bármely Python által támogatott felületen elérhető, és a már megírt kód változtatás nélkül bárhol lefut. Mindemellett az egyes felületeken programjaink is egységesen néznek ki, eltekintve a betűtípusok miatt kialakuló különbségektől.

A Tkinter a grafikus felületek elkészítéséhez a Tcl/Tk-t hívja segítségül. Mint a legtöbb Python-bővítvény, a Tcl/Tk-hoz kapcsolódó Tkinter is C-alapokra épül. Megvalósítási szinten a Tkinter két rétegre bontható: a C-ben írt `_tkinter` programkönyvtárra, valamint a már Pythonban írt `Tkinter.py`-re és az ehhez kapcsolódó egyéb Python modulokra. A `_tkinter` kapcsolja össze a Python-modulokat a Tcl/Tk-val, de a programozó elől ezt a réteget teljesen elrejtja a `Python.py`, azaz az egyetlen olyan modul, melyet a tkinteres alkalmazásaink fejlesztéséhez valóban ismernünk kell.

A Tkinter felépítése

A Tkinter felépítése, ellentétben egyéb GUI-rendszerek osztályrendszerével, nagyon egyszerű. A szerkezetben a GUI-elemek egy úgynevezett alapelemre (basewidget) épülnek, amelyhez az elemek elhelyezéséért felelős osztályok (Pack, Place és Grid), illetve a Tkinter működését befolyásoló egyéb osztályok Mix-inként kapcsolódnak (lásd alább).

A Tkinter használata

Először is készítsük el a már jól ismert „Szia Világ!” program tkinteres változatát (1. lista). A program legelső sorában a Tkinterből használt elemeket vezetjük be a `from Tkinter import Label, mainloop` paranccsal. Ha korábban már foglalkoztál Pythonnal, különösnek tűnhet ez a sor. Vajon miért használjuk a `from modulneve import` írásmódot a már jól bevált `import modulneve` helyett? Nos, képzeljük csak el, milyen fárasztó lenne, ha a programunkban minden Tkinter-elemre hivatkozáskor be kellene írogatnunk, hogy az adott elem mely modulból származik. Mivel azonban a programozók kényelmes emberek, egyszerűbb megoldást találtak ki helyette, amely az adott modul névteréből minden elemet saját programunk névterébe emel át. Bánjunk vele nagyon óvatosan, mert ezáltal minden modul minden tagfüggvénye (method) egy névtérbe kerül, és előfordulhat, hogy egy azonos nevű, bár más feladatot tagfüggvény több modulban is megtalálható, így semmiképpen sem ajánlott több modul használatát hasonló elvű beillesztése. Lépjünk tovább a második sorra! Ebben nincsen különösebb

magyarázatra szoruló rész. A `Label`-lel címkét hozunk létre, benne a jól ismert szöveggel, majd a `pack()` tagfüggvényével az elemet alkalmazásunk ablakába helyezzük. Az utolsó sorban a vezérlést a felhasználónak adjuk át.

A `mainloop()` a felhasználó és a rendszer által keltett eseményeket dolgozza fel, pontosabban ez a szolgáltatás az esetleges

billentyűleütéseket „csípi el”, illetve az ő dolga az is, hogy elemeinket újrarajzolja, ha azok előzőleg valamilyen okból takarásba kerültek.

A Mix-inekről

A Mix-inek, azaz a bekeveredő osztályok a Tkinter szerves részét képezik. Mint neve is utal rá, a Mix-in-osztályok az öröklődés során nem elsődleges ősként vannak jelen, hanem az alapos tulajdonságait bővítik ki bizonyos feladatfüggő szolgáltatásokkal, azaz saját tulajdonságaiknak az elsődleges és tulajdonságaihoz történő keverésével. Az így létrejövő gyermekosztály mind az elsődleges és, mind a hozzákevert Mix-in-ös tulajdonságait tartalmazza. Gyakran előfordul, hogy egy Mix-in-osztály egyetlen más osztálynak sem elsődleges őse. (A Mix-inekről bővebben a *Linuxvilág* áprilisi számában olvashattatok.)

Egy fejlettebb program

Mint előző példánkban láttuk, pillanatok alatt létrehozhatunk olyan programot, mely a felhasználó számára adatot közöl. Igen ám, de mi történik olyankor, amikor a felhasználót nemcsak tájékoztatni szeretnénk valamiről, hanem a véleményére is kíváncsiak vagyunk? Példánkban programunkat egy gombbal bővítjük ki, amelyre kattintva megváltoztathatjuk az ablakban megjelenő szöveget (2. lista).

Ez a program már valamivel bonyolultabb. A legfontosabb különbség, hogy ezúttal egy osztályt hoztunk létre, melynek `__init__()` eljárása az objektum megalkotása után önműködően meghívódik, s az ablakunkat, illetve az abban megjelenő gombot ez az `__init__()` eljárás hozza létre. A gombon kezdetben a megszokott „Szia Világ!” szöveg található, majd ha rákattintunk, programunk *Guidó*-nak, a Python megalkotójának köszön.

A program legelső sorában bevezetjük a Tkinter-modul összes elemét. Ezt követően a futás a 13. sorra ugrik, ahol azt vizsgáljuk, vajon mi is csak egy modul vagyunk-e vagy pedig önálló programként futunk. Ha önállóak vagyunk, osztályunkat létrehozva annak `mainloop()` eljárását nekünk kell meghívunk. A `mainloop()`-ot ebben az esetben a `Frame`-osztálytól örököljük – erről nem elfeledkezve hívjuk meg szülőosztályunk `__init__()` eljárását is (még a `mainloop()` előtt).

A 7. sorban egy gombot hozunk létre, és az ezt követő két sorban ennek tulajdonságait módosítjuk, végül a `pack()` eljárással életre is keltjük. A 9. sorban azt állítjuk be, mi történjen, amikor valaki a gombunkra kattint. Esetünkben ilyenkor osztályunk `hello_callback()` eljárása hívódik meg. Ez az eljárás – talán már feltűnt – a programból közvetlenül sehol sem hívódik meg, e feladatot a `mainloop()`-ra bízta. Ha valaki a gombon kattint, egy esemény keletkezik, amelyet a `mainloop()` kap el, és az adott elemhez tartozó *command* tulajdonságból kiemeli, milyen eljárást kell ilyenkor hívnia. Fontos tehát, hogy lássuk: az `__init__()` eljárás még a `mainloop()` előtt hajtódik végre, a `hello_callback()` eljárást viszont már a `mainloop()` hívja meg. Miként a kife-

Ha a Python összes lehetőségét kiaknázó C/C++ programokat szeretnénk írni, rendelkezésre állnak a szükséges könyvtárak, amelyekkel hozzáférhetünk a Pythonban már megszokott típusokhoz és elemekhez is.

A Python IDE

Létezik egy szintén a Tkinter felhasználásával készített program, melynek jó hasznát vehetjük Python-programok írása során. Ez a felület – egyébként IDLE-nek hívják – *Guido van Rossum*-nak, a Python atyjának a műve. Segítségével feltérképezhetjük a szerkesztett modul osztályrendszerét, az olvashatóság végett a kódot a logikai egységeknek megfelelően színezi, így az egyes részek áttekinthetőbbek és egyszerűbben módosíthatók, illetve még egy sor olyan hasznos szolgáltatással rendelkezik, amelyek jól jönnek, ha Python-kódot szerkesztünk. Segítségükkel például egyszerűen változtathatunk a behúzásokon, vagyis az indentáción, amely azt határozza meg, hogy egy sor, illetve függvény hova, melyik függvénybe, ciklusba vagy osztályba tartozik. Aki szerkesztett már Python-kódot egyszerű szövegszerkesztőből, az tudja, mennyi vesződéssel jár, ha egy kódrészt egy más beosztású részbe akarunk átcsoportosítani, és soronként kell a tabulátorokat kitörölni vagy éppenséggel tabulátorok garmadáját kell minden sor elé beszúrni. Nos, az IDLE-ből ez sem gond.

A Tkinter telepítése

Az 1.5.2-es Python-változattól kezdődően a Tkinter az alapcsomag része. Ha a rendszerünk mégsem tartalmazná, a Python webhelyéről letölthetjük (lásd *Kapcsolódó címek*).

A Debian-felhasználók helyzete a legkényelmesebb, ugyanis az `apt-get install python-tk` parancs kiadása után a csomag, illetve szükség esetén az egész Python-alaprendszer telepítődik, továbbá a *python-pmw*- és *idle*-csomagok szintén ilyen módon telepíthetők a gépünkre.

Összegzés

A Tkinterrel nagyon gyors és egyszerűen használható eszközt kapunk a kezünkbe, amellyel pillanatok alatt elkészíthetjük a mindennapi munkánkat segítő programokat, vagy ha eléggé magabiztosak vagyunk, akár mindentudó szuperalkalmazások írásába is foghatunk – a lehetőségek adottak.



Gludovátz Gábor

(ggabor@sopron.hu)

1996 óta foglalkozik Linux-rendszerekkel.

Egyik kedvenc időtöltése a programozás, jelenleg éppen egy C++-ban írt KDE-s játékon dolgozik, de szívesen kódol Pythonban és

PHP-ben is. Honlapja ➔ <http://www.sopron.hu/~ggabor/>

Kapcsolódó címek

Python ➔ <http://www.python.org>

Tkinter on-line tanfolyam

➔ <http://www.pythonware.com/library/tkinter/introduction/>

PMW – Python megawidgets ➔ <http://pmw.sourceforge.net/>

