

3.0.1 Merhaba tkinter

Bildiğiniz gibi, bütün önemli programlama dilleri işe '*Merhaba dünya!*' diyerek başlar. Geleneğe uymak için, biz de açılan pencereye '*Merhaba tkinter!*' yazalım. Bunun için aşağıdaki kodları IDLE'da yazabileceğimiz gibi, bu kodları Notepad ile yazıp `d:\pyGUI` adlı dizin içine `label01.py` adıyla kaydedebiliriz. Tabii, bunun yerine istediğiniz başka bir dizin ve başka bir dosya adı seçebilirsiniz.

Liste 3.1.

```
1 from tkinter import *
   pencere = Tk()
   e = Label(pencere, text="Merhaba tkinter!")
   e.pack()
6 pencere.mainloop()
```

Bu programı derleyip koşturmak için, Python'dan bildiğimiz gibi (bkz. [?])

```
python label01.py
```

komutunu yazmak yetecektir. Tabii, sisteminizde python'un yüklü ve ortam değişkenlerinin doğru tanımlandığını varsayıyoruz. Değilse, ortam değişkenlerinizi ayarlamalısınız.

Şimdi Liste 3.1 modülündeki deyimlerin işlevlerini açıklayalım.

1., 3. ve 7. satırın işlevlerini biliyoruz; onları önceki bölümde söyledik. 3. satırda `Tk()` kurucusu ile yaratılan nesneye `pencere` adı verildi. Yaratı-

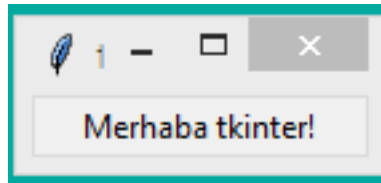
lan nesnelere birer ad vermek, o nesnelere program içinde kullanılmalarını kolaylaştıracaktır.

4.satırdaki `Label` tkinter içinde bir widget'tir; yani bir sınıftır. `Label()` metodu `Label` sınıfına ait bir nesne yaratır. `Label` etiket demektir. Görsel arayüzde widget'ler üzerine ya da yanına yazı (etiket) yazmamızı sağlar. İstenirse yazı yerine gif ya da ikonlar da konulabilir. `e` yaratılan `Label` nesnesine verilen `add`dir; yani nesneyi gösteren pointerdir. İstedığınız başka bir ad verebilirsiniz. `Label()` içine yazılan iki parametre var: `pencere` ve `text`. Birincisi olan `pencere`, etiketin pencere üzere konulacağını, ikincisi ise etiketin bir `text` olduğunu ve `text`'in içeriğini belirler. Sonuç olarak, pencere ana taşıyıcısı üzerine "*Merhaba tkinter!*" yazılır. Bu bir label'dir (etiket).

5.satırdaki `Pack()` sınıfı tkinter'in üç konuşlandırıcısından (`Layout Manager`) birisidir. `Pack()` metodu `e` widgetini pencerenin enine göre ortalar, `e`'yi taşıyıcının üst kenarına yanaşık konuşlandırır. Ayrıca, pencerenin genişliğini, ancak üzerine konulanlara yetecek kadar daraltır. Konuşlandırıcı (`layout manager`) bütün bu işleri kendiliğinden yapar. Bizim 5.satırı yazmamız yeterlidir.

7.satırdaki `mainloop()` metodu, hazırlanan `pencere` adlı ana taşıyıcının, kapatılana kadar ekranda görünmesini sağlayan döngüdür. Bu deyim yazılmazsa, pencere ekrana bir kez hızla gelir ve gider. Gözümüz onun gelişi ve gidişini farkedemez.

Şimdi komut istemi penceresini ya da daha güncel olan Windows Power Shell penceresini açıp önce `d:` sonra `cd \pyGUI` yazarak, `label01.py` dosyamızın bulunduğu dizine girelim. Başka bir deyişle, o dizini etkin kılalım. Sonra `python label01.py` yazarak Program 3.1 programını çalıştıralım. Ekranda Şekil 3.1 penceresinin açıldığını göreceğiz.



Şekil 3.1: Label01