

YAPAY ZEKA VE PYTHON PROGRAMLAMA

Fonksiyonlar
Emir Öztürk

Dr. Öğr. Üyesi Altan MESUT'un "Programlama Dillerine Giriş" ders slaytları kaynak alınmıştır.

FONKSİYON

- Okunurluk
- Tekrarın önlenmesi
- Parametre ile çalışma
- Değer döndürme

Bir program içinde sıkça kullanılan hesaplama gibi bir işlemi fonksiyon olarak tanımlamak ve ihtiyaç duyulduğunda bu fonksiyonu çağırmak daha pratik ve okununca anlaşılması daha kolay olan bir yöntemdir.

Fonksiyonlar bir ya da daha çok parametre alarak, bu parametrelere göre işlemi yapar, ve geriye bir sonuç döndürürler.

Fonksiyonun parametrelerine dışarıdan gönderilen değere (değişken yada sabit bir değer olabilir) argüman denir.

Permütasyon için 2 defa, kombinasyon için 3 defa faktöriyel hesabı döngüsünü tekrar etmek yerine faktöriyel fonksiyonu yaratılabilir.

FONKSİYON TANIMI

- `def fonksiyonadı(parametre,parametre):`
 - Blok içi
 - `return`

Fonksiyon tanımlanırken `def` anahtar kelimesi kullanılır. `def` kelimesinden sonra fonksiyonun adı, parantez içerisinde argümanları ve en son olarak blok başlangıcını belirten `“:”` karakteri kullanılır. Bu satırın tamamına fonksiyonun imzası adı verilir.

Fonksiyonun altında bir tab içeriden itibaren yazılan kodlar fonksiyonun bloğunu oluşturmaktadır. Fonksiyonun bitiminde çağırıldığı yere değer döndürmesi için `return` ifadesi kullanılır.

PROSEDÜR

- Değer döndürmeyen fonksiyon
- return
- Fonksiyon tanımı ile aynı
- Tekrar çağırılan makrolar

Fonksiyonun değer döndürmeyen türüdür.

Eğer def ile tanımlanan yapı return ifadesi ile bir değer döndürmüyorsa, bu onun prosedür olduğunu gösterir.

Genellikle matematiksel hesaplama işlemlerinde değil, isim verilerek tekrar çağrılabilen kod bloklarını yaratma amacıyla kullanılır.

NOT: Basic ve Pascal gibi bazı dillerde fonksiyon tanımı function, prosedür tanımı ise procedure yada sub (subroutine) gibi isimler ile yapılır. C ve türevi olan C++, Java, C# gibi dillerde böyle bir tanım olmadığı için genellikle prosedüre de fonksiyon denir (geriye değer döndürmediği için fonksiyon isminden önce void yazılır). Benzer şekilde Python’da da her ikisinin de tanımlanması def ile yapıldığı için prosedür terimi kullanılmaz.

ÖRNEKLER

- Permütasyon hesabı yapan fonksiyon
- Kombinasyon hesabı fonksiyon
- Belirli aralıktaki asal sayıların tespitini gerçekleştiren fonksiyon

MODÜL

- Birden fazla dosya
- Py uzantısı
- Birden fazla metod içerebilme

Python IDLE kullanırken, yorumlayıcı (Shell) kapatıldığında veya Ctrl+F6 ile tekrar başlatıldığında tanımlanmış olan tüm fonksiyon ve değişkenler kaybolur. Bunu engellemek ve programları kaydedebilmek için Ctrl+N (veya File menüsünden New File) ile yeni bir kod yazma penceresi açılır. F5 ile çalıştırmak istendiğinde kaynak kodunun belli bir yere kaydedilmesi istenir. Kaydedilince uzantısı ‘.py’ olan bu kod dosyalarına aslında modül adı verilir.

MODÜL KULLANMA

- modül.istenilenfonksiyon
- import
- import modül
- import modül.istenilenfonksiyonlar
- import modül.*
- from modül import istenilenfonksiyon
- İsim tekrarı

Kod varsayılan klasör olan Python36 içine kaydedilirse, içindeki sınıflar, fonksiyonlar ve global değişkenler Shell içinden veya başka bir modülden import ile istenildiği zaman kullanılır hale gelir. Eğer ‘py’ dosyaları aynı klasörde ise Python36 içinde olmasalar da birbirlerini import edebilirler. Modül bu şekilde import edildikten sonra içindeki bir fonksiyon ile aynı isimde bir fonksiyon tanımlanırsa, lokalde tanımlanan fonksiyon geçerli olur.

LOKAL – GLOBAL

- Blok hiyerarşisi
- Değişkenin tanımlandığı blok
- En üst blokta tanımlanan değişken
- global ifadesi

Fonksiyon bloğunda tanımlanan bir değişken lokal değişkendir ve sadece o fonksiyon içinde kullanılabilir.

Fonksiyon dışında (modül içinde) tanımlanan değişken ise global değişkendir ve o modüle ait tüm fonksiyonlarda kullanılabilir.

Python’da bir değişkene değer atandığı anda o değişken oluşturulduğu için, global bir değişkenin değeri fonksiyon içinden değer atama yolu ile değiştirmek istendiğinde lokal olarak aynı isimde bir değişken yaratılır.

+= gibi bir operatör ile bu değer arttırılmak istediğinizde ise değişkenin tanımlı olmadığı hatası alınır.

Bunun önüne geçmek için ‘global’ ifadesi kullanılır.

RECURSIVE (ÖZYİNELEMELİ) FONKSİYONLAR

- Kendi kendisini çağıran fonksiyon
- İki parça
 - Kendisini çağırdığı parça
 - Durma koşulu
- Sonsuz döngü!

Bir fonksiyon kendi bloğu içinden kendisini çağırabilir. Buna özyineleme denir. Eğer kontrolsüz yapılırsa bu durum bir sonsuz döngüye neden olur. Eğer bir ‘if’ yapısı ile kontrol edilirse, bu özellik bir döngü gibi kullanılabilir.

NESNEYE YÖNELİK PROGRAMLAMA

- Prosedürler
- Fonksiyonlar
- Nesneler
- Soyutlama
- Yüksek seviyeli programlama

PASCAL ve C gibi yapısal programlama dilleri yaygınlaşmadan önce (60’lı yıllar) kullanılan BASIC gibi programlama dillerinde prosedür ve fonksiyon kullanılmadığı için, prosedür yerine blok yapılması gereken yerin başına bir etiket tanımlanıp o etikete GOTO komutu ile gidilerek program bloklara ayrılmaya çalışılırdı.

BASIC programlama diline de 80’li yıllarda yapısal programlamaya özgü fonksiyon ve prosedür tanımlanabilmesi özelliği eklenmiştir (QBASIC).

90’lı yıllarda QBASIC daha da geliştirilerek görsel programlama amaçlı Visual Basic ortaya çıkmıştır.

ÖRNEK

- Parametre olarak aldığı sayıyı yazı olarak yazdıran fonksiyon
- Yazılan metotları tek bir modüle toplayarak bu modülü kullanan bir uygulama
- Aldığı iki sayının bölen toplamlarının birbirlerine eşit olup olmadığını döndüren fonksiyon

ÖRNEK

- M.Ö. 2589-2566 yılları arasında yapıldığı tahmin edilen dünyanın 7 harikasıdan biri olan Keops Piramidi'nin en az 2.3 milyon taş bloktan oluştuğu tahmin edilmektedir.
- Piramidin tabanında kare biçiminde yerleştirilmiş $N \times N$ adet, ikinci katında $(N-1) \times (N-1)$, üçüncü katında $(N-2) \times (N-2)$, ... , N. katında (son kat) 1 taş blok olduğu kabul edilirse, piramidin kaç taş bloktan oluştuğunu ve kaç katlı olduğunu bulan programı yazınız (2.3 milyondan büyük olacak şekilde en az kaç kat olmalıdır).
- Piramidin tam 24 yılda bittiğini ve işçilerin günde 10 saat çalıştıklarını düşünelim. İşçilerin bir saatte ortalama kaç blok yerleştirdiklerini bulan SaatBasınaBlok fonksiyonu yazınız (1 yıl = 365 gün + 6 saat kabul edilecek).