Ünal Sarıemir 1.Hafta

1.Heap ve Stack alanları nedir? Nasıl çalışır?

Stack (Yığın):

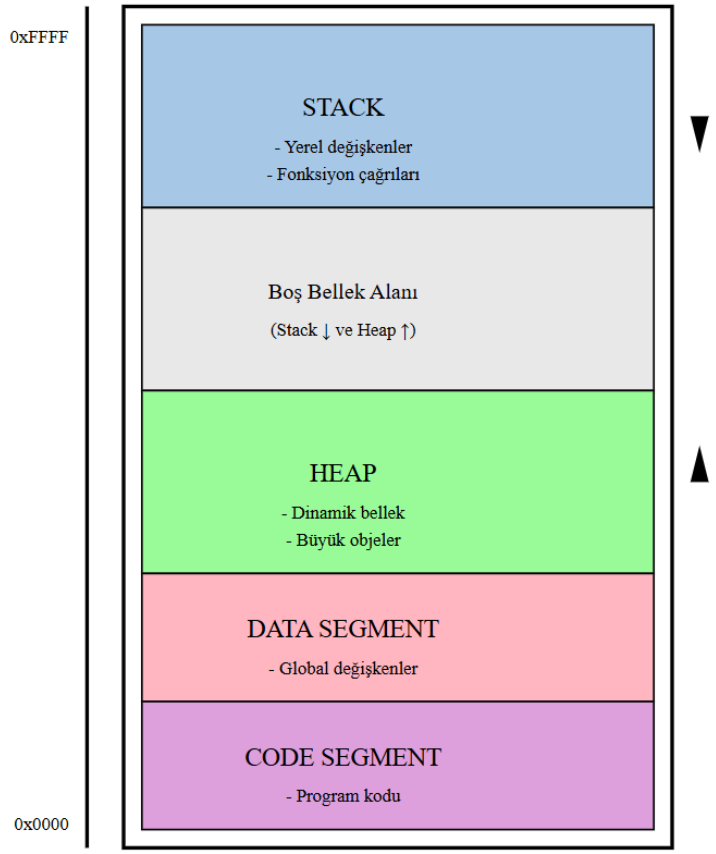
- Statik bellek tahsisi için kullanılır.
- LIFO (Last in First Out - Son Giren İlk Çıkar) prensibiyle çalışır.
- Fonksiyon çağrıları ve yerel değişkenler burada saklanır.
- Boyutu derleme zamanında belirlenir.
- Sınırlı bir bellek alanına sahiptir.
- Değişkenler otomatik olarak silinir (scope dışına çıkınca).

CPU'ya yakın, hızlı erişilen, LIFO (Last in First Out) yapısında bellek bölgesi

Heap (Öbek)

- Dinamik bellek tahsisi için kullanılır.
- Program çalışma zamanında bellek tahsisi yapılabilir.
- Daha büyük veri yapıları için kullanılır.
- Stack'e göre daha yavaş erişim sağlar.
- Bellek yönetimi programcı tarafından yapılmalıdır.
- Garbage Collection olmayan dillerde memory leak riski vardır.

Dinamik, büyük, dağınık bellek bölgesi



Stack:

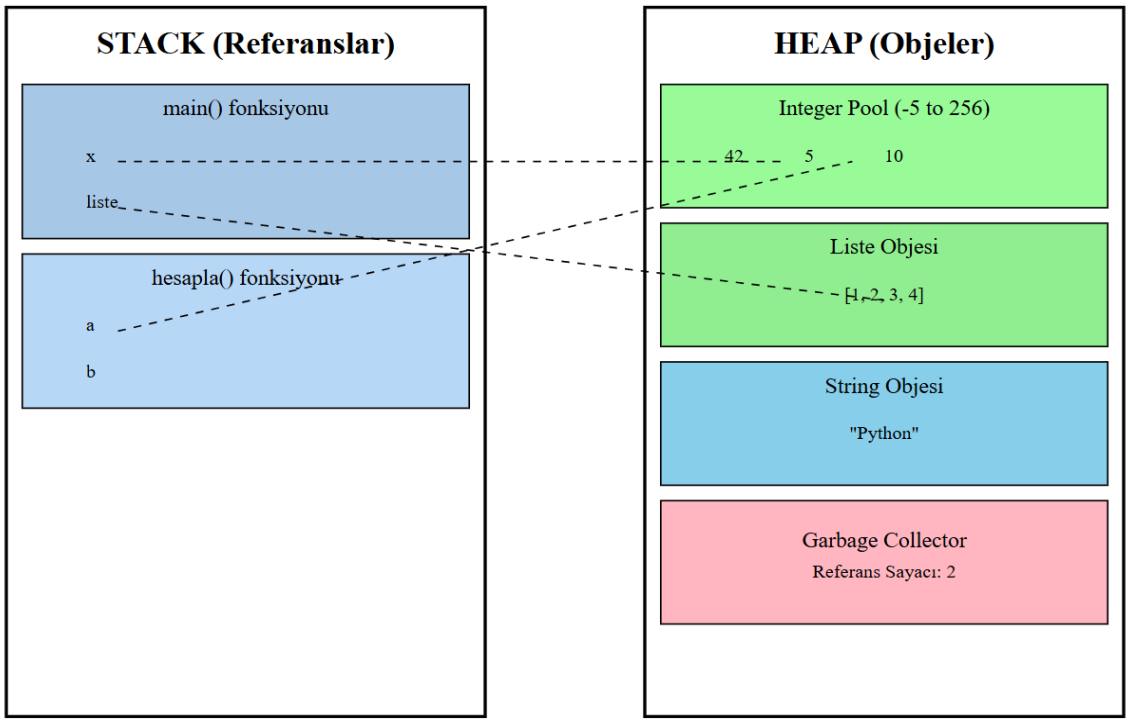
- Değişken isimleri ve referanslar burada tutulur
- Her fonksiyon çağrısı yeni bir frame oluşturur
- LIFO (Last in First Out) prensibiyle çalışır
- Sadece referansları içerir, gerçek veriler heap'tedir.

Heap:

- Tüm Python objeleri burada yaşar:
    - Integer Pool (-5 ile 256 arası sayılar)
    - Listeler
    - Stringler
    - Sınıf örnekleri
- Garbage Collector bu alanı yönetir

Referans Sistemi:

- Kesikli çizgiler stack'teki referansların heap'teki objeleri nasıl gösterdiğini belirtir
- Bir objenin birden fazla referansı olabilir
- Referans sayısı 0'a düşünce Garbage Collector objeyi temizler

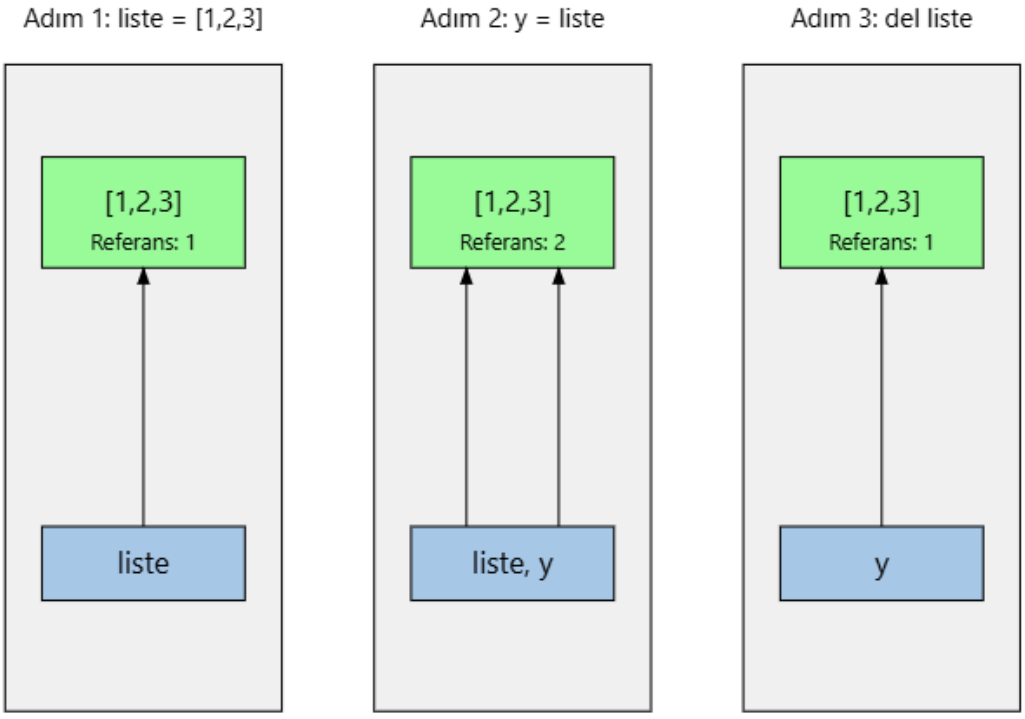


2.Garbage Collector nedir? Ne işe yarar? Nasıl çalışır?

- Kullanılmayan bellek alanlarını otomatik tespit eder ve temizler.
- Memory leak'leri önler.
- Programcıyı manuel bellek yönetiminden kurtarır.

```
x = [] # referans sayısı: 1
y = x # referans sayısı: 2
del x # referans sayısı: 1
del y # referans sayısı: 0, artık toplanabilir
```

Referans sayısı, bir objenin kaç değişken tarafından gösterildiğini (referans edildiğini) belirten sayıdır.



3.Örnekler

x = 2 #x ismi stack'te, 2 değeri heap'te

y = 2 #y de aynı 2'yi gösterir (heap'teki aynı yeri)

liste1 = [1,2,3] # liste1 ismi stack'te, [1,2,3] heap'te

liste2 = liste1 # liste2 de aynı listeyi gösterir

liste2.append(4) # liste1 de değişir çünkü aynı yeri gösteriyorlar

a = "Merhaba” #a ismi stack'te, "Merhaba" heap'te

b = "Merhaba” #b de aynı string'i gösterir

 Stack: Değişken isimleri (x, liste1, a gibi)

 Heap: Asıl değerler (2, [1,2,3], "Merhaba" gibi)