YAZILIM MÜHENDİSLİĞİ

30 Ekim 2022

ÖDEV 1

Elif TOSUN

2103022921

Recursive(Özyineleme) Fonksiyon Nedir?

Özyineleme , bir işlevin kendi kodu içinde kendisini çağırması ve böylece içinde bulunan talimatları tekrar tekrar yürütmesidir.

Iteration(Yineleme) Fonksiyon Nedir?

Yineleme , bir döngünün "for" döngüleri ve "while" döngüleri gibi komut setini tekrar tekrar yürütmesidir.

Özyineleme Ne Zaman Kullanılır?

Bazen doğrudan çözülemeyecek kadar karmaşık bir sorunla karşılaşırız. Çoğu durumda, bu tür sorunları daha küçük parçalara ayırmaya çalışırız. Ardından, bu küçük parçaları çözmenin bir yolunu bulabilir ve ardından tüm soruna bir çözüm oluşturabiliriz. Özyinelemenin arkasındaki tüm fikir budur.

Yineleme Ne Zaman Kullanılır?

Yineleme, döngüyü kontrol eden koşul yanlış olana kadar bir dizi talimatı tekrar tekrar yürüttüğümüz zamandır. Başlatma, karşılaştırma, yineleme içindeki ifadeleri yürütme ve kontrol değişkenini güncellemeyi içerir.

Özyineleme ve Yineleme Arasındaki Temel Farklılıklar

Yineleme, özyinelemeye göre daha hızlı ve alan açısından daha verimlidir. Öyleyse neden özyinelemeye ihtiyacımız var? Nedeni basit - belirli bir problem için özyinelemeli bir yaklaşım kodlamak daha kolaydır.Özyineleme ve yineleme, bir dizi talimatı tekrar tekrar yürütmenin farklı yollarıdır. Bu ikisi arasındaki temel fark, özyinelemede, ifadeleri işlev gövdesi içinde tekrar tekrar yürütmek için işlev çağrılarını kullanmamız, yinelemede ise aynısını yapmak için "for" ve "while" gibi döngüler kullanmamızdır.
Yineleme, özyinelemeye göre daha hızlı ve alan açısından daha verimlidir. Öyleyse neden özyinelemeye ihtiyacımız var? Nedeni basit - belirli bir problem için özyinelemeli bir yaklaşım kodlamak daha kolaydır.

```
# Bir dizeyi tersten yazdırma
def iterativeReverse(string):
    return "".join(list(reversed(string)))

def iterativeReverse2(string):
    newList = []
    listStr = list(string)
    for i in range(len(string)):
        newList.append(listStr[len(listStr)-1])
        listStr.pop(len(listStr)-1)
    return "".join(newList)

def recursiveReverse(string):
    if string == "":
        return ""
    return string[-1] + recursiveReverse(string[:-1])

recursiveReverse("Software Engineer")
```

Şekil 1: Bir dizeyi tersten yazdıran programı recursive ve iterative olarak yazma

```python
# listenin artan olup olmadıgını belirleme
def iterativeIsAscending(listA):
    if listA == sorted(listA):
        return True
    return False

def iterativeIsAscending2(listA):
    result = True
    result = 0 if len(listA) == 0 else result
    for i in range(len(listA)-1):
        temp = listA[i]
        if listA[i+1] > temp:
            continue
        else:
            result = False
    return result

def resursiveIsAscending(listA):
    result = False
    result = 0 if len(listA) == 0 else result
    result = True if len(listA) == 1 else result
    if len(listA) > 1:
        firstValue = listA[0]
        if listA[1] > firstValue:
            listA.pop(0)
            resursiveIsAscending(listA)
    return result
iterativeIsAscending([10,20,30,40])
iterativeIsAscending([10,30,80,20])
```

Şekil 2: Bir listenin artan olup olmadığını belirleyen programı recursive ve iterative olarak yazma