LABORATUVAR ÇALIŞMASI 12 – Sınıf Yapısı

Bu Çalışmanın Amacı

Bu çalışmadaki amacımız, Python programlama dilindeki sınıf yapısı konusunda öğrendiklerimizi pekiştirmektir.

Sınıf Kavramı

Programlama yaparken asıl amaç her ne kadar belirli bir işlevi mümkün ölçüde az maliyetle ve doğru bir biçimde gerçekleştirmek olsa da zaman zaman (özellikle de büyük ölçekli programlarda) programın okunabilirliği, anlaşılabilirliği ve doğallığı yönündeki kaygılar programcıyı modülerliğe, bunun bir sonucu olarak da nesne yönelimli programlamaya itmektedir.

Python programlama dili, nesne yönelimli programlamayı destekleyen yüksek seviyeli bir dildir. Nesne yönelimli programlamanın olmazsa olmazı olan sınıf yapısı, nesne yönelimliliği destekleyen diğer programlama dillerinde olduğu gibi Python' da da mevcuttur.

```
class Dikdortgen:
    """ Ornek bir sinif"""

    def __init__(self, arg_en, arg_boy):
        self.en = arg_en
        self.boy = arg_boy

    def cevre(self):
        return (2 * (self.en + self.boy))

    def alan(self):
        return (self.en * self.boy)

    def buyut(self, katsayi):
        self.en = self.en * 2
        self.boy = self.boy * 2

    def ciz(self):
        karakter = ''
        for i in range(0, self.en):
            karakter = karakter + '#'
        for j in range(0, self.boy):
            print karakter
```

Yukarıda, **Dikdortgen** isminde örnek bir sınıf verilmiştir. Bu sınıf içerisinde 5 farklı fonksiyon görülmektedir.

Bu fonksiyonlar içerisinde "**__init__**" fonksiyonu, **Dikdortgen** sınıfının yapıcı metodudur. Yapıcı metodun görevi, içerisinde bulunduğu sınıfın bir nesnesi yaratıldığında bu nesnenin hangi alt alanlara sahip olacağını ve bu alt alanların ilk değerlerinin ne olacağını belirlemektir. **__init__** fonksiyonunun aldığı ilk argüman, "**self**" adını taşımaktadır. Bu argümanın görevi, **Dikdortgen** sınıfının nesnelerini sembolik olarak göstermektir; yani, bu sınıfın nesnelerine ait olan alt alanlar, sınıfın tanımlanması esnasında **self** argümanının birer alt alanıymış gibi tanımlanırlar. Özel bir görev üstlenen bu argümana, "self" haricinde herhangi bir isim de verilebilir ancak bir programlama geleneği olarak "self" adının verilmesi daha uygun olmaktadır. Fonksiyonun 2. ve 3. argümanları, sırasıyla, bu sınıfın nesnelerinde alt alan olarak bulunacak olan **en** ve **boy** değişkenlerine verilecek olan ilk değerleri tutmaktadırlar. Bu yapıcı metodun çağrılmasında, metoda toplamda 2 argüman verilecektir (**arg_en** ve **arg_boy** yerine). İlk argüman olan "**self**", yapıcı metodun çağrılması sırasında argüman olarak verilmemektedir.

cevre fonksiyonuna bakıldığında, tek bir argüman aldığı görülmektedir. "**self**" isimli bu argüman aslında bu fonksiyona çağrı yapan **Dikdortgen** sınıfı nesnesini temsil etmektedir. **Dikdortgen** sınıfının bir nesnesi yaratıldığında, bu nesneyi refere eden referansın adından sonra "." işareti konarak "**cevre()**" yazılmasıyla **cevre** fonksiyonu çağrılmış olacaktır. Fonksiyonun çağrılması esnasında herhangi bir argüman verilmeyecek, fonksiyon tanımında görülen "**self**" argümanı ise "." işaretinden önce yer alan nesne referansı tarafından gösterilen nesneyi temsil edecektir. **cevre** fonksiyonu da bu nesnenin **en** ve **boy** alt alanlarının değerlerinin toplamını 2 ile çarparak bulduğu sonucu döndürecektir. **alan** ve **ciz** fonksiyonlarının çalışma prensipleri de **cevre** fonksiyonununkiyle aynıdır.

buyut fonksiyonu ise, **self** ve **katsayi** olmak üzere iki argüman almaktadır. Yani bu fonksiyon çağrıldığında kendisine parantezler içerisinde 1 argüman (**katsayi**) verilecektir. **buyut** fonksiyonu ise, kendisini çağıran nesnenin **en** ve **boy** alt alanlarında bulunan değerleri **katsayi** argümanıyla alınan değer ile çarparak güncelleyecektir.

Aşağıda, bu sınıfın kullanımına dair bir örnek yer almaktadır:

```
>>> dd1 = Dikdortgen(5, 2)
>>> dd1.en
5
>>> dd1.boy
2
>>> dd1.cevre()
14
>>> dd1.alan()
10
>>> dd1.ciz()
#####
#####
>>> dd1.buyut(2)
>>> dd1.en
10
>>> dd1.boy
4
>>> dd1.ciz()
##########
##########
##########
##########
>>>
```

Yukarıdaki örnek kullanımda, ilk başta **Dikdortgen** sınıfının bir nesnesi yaratılarak **dd1** referansı ile gösterilmiştir. Nesnenin yaratılması esnasında çağrılan **__init__** yapıcı metoduna, **arg_en** ve **arg_boy** argümanlarının yerine sırasıyla **5** ve **2** verilmiştir. İlgili nesnenin **en** ve **boy** alt alanlarına erişmek için "**dd1.en**" ve "**dd1.boy**" yazılması gerekmektedir. Nesne üzerinden sırasıyla **cevre**, **alan** ve **ciz** fonksiyonlarının çağrılmasından sonra **buyut** fonksiyonu çağrılmıştır. **buyut** fonksiyonuna **katsayi** argümanı yerine **2** verilmiştir. Bu sayede, **dd1** referansı ile gösterilen nesnenin **en** ve **boy** alt alanlarının sahip olduğu değerler 2 katına çıkarılmıştır. En son adımda çağrılan **ciz** fonksiyonu da, bir önceki çağrılışında ürettiği ekran çıktısından daha farklı bir değer üretmiştir.

Alıştırmalar

Alıştırma – 1

Görev

"**Kompleks**" isminde, kompleks (karmaşık) sayıları temsil etmeye yönelik bir sınıf oluşturunuz. Sınıfınızın nesnelerinde alt alan olarak "**gercel**" ve "**sanal**" değişkenleri bulunmalıdır. Bu alt alanların ilk değerleri, nesne yaratılması esnasında yapıcı metot yolu ile belirlenebilmelidir. Ayrıca, sınıfınızda "**Mutlak**" ve "**Yazdir**" isminde iki fonksiyon bulunmalıdır. **Mutlak** fonksiyonu, **gercel** ve **sanal** değişkenlerinde tutulan değerlerin karelerini toplayarak sonucun karekökünü döndürmelidir. **Yazdir** fonksiyonu ise, sınıf nesnesi ile temsil edilen kompleks sayıyı "*[gercel] + [sanal] i*" formatında ekrana yazdırmalıdır. Örnek kullanım:

```
>>> sayi1 = Kompleks(6, 8)
>>> sayi1.gercel
6
>>> sayi1.sanal
8
>>> sayi1.Mutlak()
10.0
>>> sayi1.Yazdir()
6 + 8 i
>>>
```

Sonuç

Gerçekleştiriminizi ve / veya karşılaştığınız problemleri raporunuza yazınız.

Alıştırma – 2

"**Karakter**" isminde, karakter dizileri üzerinde birleştirme, kesit alma ve tekrarlama işlemlerini yapmaya yönelik bir sınıf tasarlayınız. Sınıfınızın nesnelerinde, "**kdizi**" isminde tek bir alt alan bulunmalıdır. Sınıfınızda yer alması beklenen üç fonksiyondan **Birlestir** fonksiyonu, nesne alt alanı olan **kdizi** değişkeninde tutulan karakter dizisinin sonuna argüman olarak alacağı karakter dizisini ekleyerek elde edilen yeni karakter dizisini döndürmelidir. **KesitAl** fonksiyonu ise iki tamsayı argüman almalı, **kdizi** içerisinde tutulan karakter dizisinin 1. argümanda tutulan sayısal değere karşılık gelen indeksinden itibaren, 2. argümanda tutulan sayısal değer kadar karakterini alarak döndürmelidir (Örnek kullanımı incelemeniz bu fonksiyonun işlevini daha anlaşılır kılacaktır.). **Tekrarla** fonksiyonu ise, **kdizi** ile tutulan karakter dizisini, argüman olarak aldığı pozitif tamsayı değer kadar tekrarlayarak sonucu döndürmelidir. **Fonksiyonlara argüman olarak uygun değerler girilecektir, ekstra hata kontrolü yapılmasına gerek yoktur.** Örnek kullanım:

```
>>> sozcuk = Karakter("bilgi")
>>> sozcuk.kdizi
'bilgi'
>>> sozcuk.Birlestir("sayar")
'bilgisayar'
>>> sozcuk.KesitAl(1,3)
'ilg'
>>> sozcuk.Tekrarla(3)
'bilgibilgibilgi'
```

Sonuç

Gerçekleştiriminizi ve / veya karşılaştığınız problemleri raporunuza yazınız.

Alıştırma – 3

Alıştırma 2' de tasarlamış olduğunuz **Karakter** sınıfına, 3 fonksiyon daha ekleyiniz:

- **Birlestir_Tut → Birlestir** fonksiyonundan farklı olarak, birleştirme sonrasında elde edilen değeri döndürmek yerine bu değeri ilgili nesnenin **kdizi** alt alanına atar.
- **KesitAl_Tut → KesitAl** fonksiyonundan farklı olarak, kesit alma sonrasında elde edilen değeri döndürmek yerine bu değeri ilgili nesnenin **kdizi** alt alanına atar.
- **Tekrarla_Tut → Tekrarla** fonksiyonundan farklı olarak, tekrarlama sonrasında elde edilen değeri döndürmek yerine bu değeri ilgili nesnenin **kdizi** alt alanına atar.

Örnek kullanım:

```
>>> sozcuk = Karakter("bilgi")
>>> sozcuk.kdizi
'bilgi'
>>> sozcuk.Birlestir_Tut("sayar")
>>> sozcuk.kdizi
'bilgisayar'
>>> sozcuk.KesitAl_Tut(4, 4)
>>> sozcuk.kdizi
'isay'
>>> sozcuk.Tekrarla_Tut(2)
>>> sozcuk.kdizi
'isayisay'
>>>
```

Sonuç

Gerçekleştiriminizi ve / veya karşılaştığınız problemleri raporunuza yazınız.