|  |
| --- |
| class Araba:      def \_\_init\_\_(self, marka, model):          self.marka = marka          self.model = model      def bilgileri\_yazdir(self):          print(f"Marka: {self.marka}, Model: {self.model}")  # Kullanım  araba1 = Araba("Toyota", "Corolla")  araba1.bilgileri\_yazdir() |

1-Aşağıdaki kodu yorumlayınız

Araba isminde bir sınıf oluşturulmuş. Bu sınıfın iki özelliği bulunmakta ‘marka’ ve ‘model’ diye. Daha sonra bu sınıfın içine bilgileri\_yazdir diye bir metot oluşturulmuş. Bu metotta objenin marka ve model bilgisini ekrana yazdırıyor. Daha sonra mainde araba1 objesi oluşturulmuş. Bu Araba sınıfına ait bir obje ve marka sı Toyota modeli Corolla atanmış. Daha sonra bilgileri\_yazdir metoduyla bu marka ve model ekrana yazdırılmış.

2-Bir Dikdortgen sınıfı oluşturun. Bu sınıfın genişlik ve yükseklik bilgilerini alın ve bir metot aracılığıyla alanı hesaplayıp döndürün.

class Dikdortgen:

def \_\_init\_\_(self, genislik, yukseklik):

self.genislik = genislik

self.yukseklik = yukseklik

def alan\_hesapla(self):

alan = self.genislik \* self.yukseklik

return alan

dikdortgen1 = Dikdortgen(5, 10)

print("Dikdörtgenin alanı:", dikdortgen1.alan\_hesapla())

|  |
| --- |
| class Dikdortgen:      def \_\_init\_\_(self, genislik, yukseklik):          self.genislik = genislik          self.yukseklik = yukseklik      def alan\_hesapla(self):          return self.genislik \* self.yukseklik  # Kullanım  dikdortgen1 = Dikdortgen(5, 10)  alan = dikdortgen1.alan\_hesapla()  print(f"Dikdörtgenin alanı: {alan}") |

3-Bir Kare sınıfı oluşturun. Bu sınıfın bir kenar uzunluğu olsun ve kenarı verilen kareyi ekrana yazdıran bir metot yazın (metot sadece yazdırma yapacak, değer döndürmeyecek).

class Kare:

def \_\_init\_\_(self, kenar):

self.kenar = kenar

def kareyi\_yazdir(self):

for i in range(self.kenar):

print("\* " \* self.kenar)

kare1 = Kare(5)

kare1.kareyi\_yazdir()

|  |  |
| --- | --- |
| class Kare:      def \_\_init\_\_(self, kenar):          self.kenar = kenar      def kareyi\_yazdir(self):          for i in range(self.kenar):              print("\* " \* self.kenar)  # Kare sınıfından bir nesne oluşturma ve kareyi yazdırma  kare1 = Kare(5)  kare1.kareyi\_yazdir() | class Kare:      def \_\_init\_\_(self, kenar\_uzunlugu):          self.kenar\_uzunlugu = kenar\_uzunlugu      def kareyi\_yazdir(self):          for i in range(self.kenar\_uzunlugu):              print('\* ' \* self.kenar\_uzunlugu)  # Kullanım örneği:  kare = Kare(5)  kare.kareyi\_yazdir() |

4-Bir HesapMakinesi sınıfı oluşturun ve iki veya üç sayı arasında toplama yapabilen bir metot yazın. Üçüncü sayı verilmezse, iki sayıyı toplasın.

class HesapMakinesi:

def toplama(self, sayi1, sayi2, sayi3=None):

if sayi3 is not None:

toplam = sayi1 + sayi2 + sayi3

else:

toplam = sayi1 + sayi2

return toplam

hesap = HesapMakinesi()

print("2 sayının toplamı:", hesap.toplama(5, 10))

print("3 sayının toplamı:", hesap.toplama(5, 10, 15))

|  |
| --- |
| class HesapMakinesi:      def topla(self, sayi1, sayi2, sayi3=None):          if sayi3 is not None:              return sayi1 + sayi2 + sayi3          else:              return sayi1 + sayi2  # HesapMakinesi sınıfından bir nesne oluşturma  hesap\_makinesi = HesapMakinesi()  # İki sayı ile toplama  sonuc1 = hesap\_makinesi.topla(10, 20)  print("İki sayının toplamı:", sonuc1)  # Üç sayı ile toplama  sonuc2 = hesap\_makinesi.topla(10, 20, 30)  print("Üç sayının toplamı:", sonuc2) |

5-Bir Merhaba sınıfı oluşturun ve kullanıcıdan herhangi bir bilgi almadan "Merhaba Dünya" yazdıran bir metot oluşturun. Bu metot, herhangi bir değer almaz ve döndürmez.

class Merhaba:

def selam\_ver(self):

print("Merhaba Dünya")

mesaj = Merhaba()

mesaj.selam\_ver()

|  |
| --- |
| class Merhaba:      def merhaba\_yazdir(self):          print("Merhaba Dünya")  # Merhaba sınıfından bir nesne oluşturma ve metodu çağırma  merhaba = Merhaba()  merhaba.merhaba\_yazdir() |