

BİLGİLENDİRME

Kodlarda bazı eksiklikler olabilir. Ayrıca bazı haftalarda proje ödevlerinden dolayı ders işlenmedi eksik haftalar var şeklinde düşünmeyin...

10. HAFTA

► Class (Örnek 1)

```
class konu:  
    def ses(self):  
        print "Merhaba kardes"  
  
yarat = konu()  
yarat.ses()
```

► Class (Örnek 2)

```
class Oyun:  
  
    def __init__(self): #__init__ class'in yapici metodu #self: disaridan ulasilmasini istediginiz her  
fonksiyona self parametresini ekleyiniz  
        self.enerji = 50  
        self.para = 100  
        self.fabrika = 4  
        self.isci = 10  
  
    def goster(self):  
        print "enerji", self.enerji  
        print "para", self.para  
        print "fabrika", self.fabrika  
        print "isci", self.isci  
  
    def fabrikakur(self, miktar):  
        if self.enerji>3 and self.para>10:  
            self.fabrika = miktar + self.fabrika  
            self.enerji = self.enerji - 3  
            self.para = self.para - 10  
            print miktar, "adet fabrika kurdunuz!"  
        else:  
            print "para ve enerji yetersiz"  
  
yaz = Oyun()  
yaz.fabrikakur(5)  
yaz.goster()
```

► Class (Örnek 3)

```
class isci:  
    def __init__(self, isim, maas):  
        self.isim=isim  
        self.maas=maas  
  
    def soyisim(self):  
        return self.isim.split()[-1] #tersten ilk eleman  
  
    def artis(self, oran):  
        self.maas*=(1.0+oran)  
  
kamil=isci("Kamil KOC",50000)  
recep=isci("Ali AGAOGLU",10000)  
  
print kamil.soyisim()  
print recep.soyisim()  
recep.artis(.10)  
print recep.maas
```

► Class (Örnek 4)

```
class isim:
    def __init__(self,adim):
        self.adim=adim
    def soyle(self):
        print "Merhaba",self.adim

ad = raw_input("isim giriniz: ")
adam=isim(ad)
adam.soyle()
```

► Class (Örnek 5)

```
class Animal:
    def cevap(self):
        self.konus()
    def konus(self):
        print "Animal"

class Mammal(Animal): #kalitim yapilmis
    def konus(self):
        print "from animals to mammal"

class Cat(Mammal):
    def konus(self):
        print "Cat"

class Dog(Mammal):
    def konus(self):
        print "Dog"

class Human(Mammal):
    def konus(self):
        print "Human"

class Kisi(Human):
    pass

if __name__ == '__main__':
    attrib = Cat()
    attrib.konus()
    bilgi = Kisi()
    bilgi.konus()
```

► Class (Örnek 6)

```
class birinci:
    def __init__(self, deger):
        self.deger = deger
    def __call__(self, digerdeger):
        return self.deger * digerdeger

x = birinci(2)
print x(3)
print x(4)
```

► Class (Örnek 7)

```
class kriter(object):
    def __init__(self):
        print "birinci kriter dogum yeri"
    def talk(self):
        print "ikinci kriter sinifinin instance i"

kriter1 = kriter()
kriter2 = kriter()
kriter1.talk()
kriter2.talk()
```

► Class (Örnek 8)

```
class Araba(object):
    def __init__(self):
        self.araba_lastik_sayisi = 4
    def set_sayi(self, xYon):
        print "tekerlek yonu: " + str(xYon)
    def lastik_sayisi(self):
        return self.araba_lastik_sayisi

class Arazi(Araba):
    def __init__(self):
        self.araba_lastik_sayisi = 6

def tekerlekler(araba):
    for i in range(1, araba.lastik_sayisi()):
        print "tekerlek sayisi: " + str(i)
        araba.set_sayi(i)
    print "tekerlek saysi : " + str(araba.lastik_sayisi())
    araba.set_sayi(i+1)

a = Araba()
tekerlekler(a)
ar = Arazi()
tekerlekler(ar)
```

Arkadaşlar 11.Hafta dersimizde Python ile veritabanı işlemleri yaptık veritabanı işlemlerinde bize yardımcı olması için küçük bir uygulama indirmemiz gerekiyor. Uygulamamızın adı –SQLite– bu uygulamayı Mozilla’da eklenti olarak indirebilirsiniz.

► Veritabanı Bağlantısı (Örnek 1)

```
import sqlite3 as db
baglan = db.connect("test.db")
#baglan = db.connect("127.0.0.1")
imlec = baglan.cursor()
imlec.execute("""CREATE TABLE IF NOT EXISTS ogr (isim, soyisim, numara)""")
#imlec.execute("INSERT INTO ogr VALUES ('ahmet','nar','11')")
veriler = [
    ('okan','alp',1),
    ('ilker','yildirim',2),
    ('ahmet','turkmenoglu',3)
]
for i in veriler:
    imlec.execute("""INSERT INTO ogr VALUES (?, ?, ?)""", i)

baglan.commit()
imlec.execute("SELECT * FROM ogr")
yazdir = imlec.fetchall()
print yazdir
```

► Veritabanı Bağlantısı (Örnek 2)

```
import sqlite3 as db
baglan = db.connect("test.db")
imlec = baglan.cursor()

isim = raw_input("isim : ")
soyisim = raw_input("soyisim : ")
numara = input("numara : ")

imlec.execute("""INSERT INTO ogr VALUES (?, ?, ?)""", (isim, soyisim, numara))

baglan.commit()
imlec.execute("SELECT * FROM ogr")
yazdir = imlec.fetchall()
print yazdir
```

► Veritabanı Bağlantısı (Örnek 3)

```
import sqlite3 as db
baglan = db.connect("test.db")
imlec = baglan.cursor()

isim = raw_input("isim : ")

imlec.execute("""SELECT * FROM ogr WHERE isim='%s'""" %(isim))
imlec.execute("""SELECT * FROM ogr WHERE isim= :isim""", {"isim":isim})

yazdir = imlec.fetchone()
if yazdir:
    print u"bu isimde ogrenci var %s" %yazdir[0]
else:
    print u("bu isimde ogrenci yoktur")
```

Veritabanı işlemlerine devam ettik...

► Veritabanı Bağlantısı (Örnek 1)

```
import sqlite3

class Database:
    def __init__(self):
        try:
            self.baglan = sqlite3.connect('ogrenci.db')
        except sqlite.OperationalError:
            exit(1)
        self.cursor = self.baglan.cursor()

    def veritabani_yarat(self):
        self.cursor.execute("""CREATE TABLE IF NOT EXISTS ogrenci (id INT PRIMARYKEY, isim CHAR(50), soyadi CHAR(50), numara INT)""")

    def kayit_ekle(self):
        ogrdict = [(100, "ahmet", "isli", 123456),
                    (101, "kemal", "adil", 988745),
                    (105, "fahri", "dali", 456213)]
        self.cursor.executemany("""INSERT INTO ogrenci (id, isim, soyadi, numara) VALUES (?, ?, ?, ?)""",
                                ogrdict)

    def kayit_cek(self):
        print ("kayitlar cekiliyor...\n")
        self.cursor.execute("""SELECT * FROM ogrenci""")
        yazdir = self.cursor.fetchall()
        print yazdir

    def veritabani_kapat(self):
        self.baglan.commit()
        self.baglan.close()

deneme = Database()
deneme.veritabani_yarat()
deneme.kayit_ekle()
deneme.kayit_cek()
deneme.veritabani_kapat()
```

► Veritabanı Bağlantısı (Örnek 2)

```
import sqlite3
con = sqlite3.connect("my_db.db")
cur = con.cursor()
cur.execute("""CREATE TABLE IF NOT EXISTS clients (id INT PRIMARYKEY, firstname CHAR(60), lastname CHAR(60))""")

try:
    clients = [
        (107, "Ella", "Fitzgerald"),
        (108, "Louis", "Armstrong")
    ]
    cur.executemany("""INSERT INTO clients (id, firstname, lastname) VALUES (?, ?, ?)""", clients)
except:
    pass

try:
    new_clients = (110, "Benny", "Goodman")
    cur.executemany("""INSERT INTO clients (id, firstname, lastname) VALUES (?, ?, ?)""", new_clients)
except:
    pass

con.commit()

print("Show data row by row")
#-----
cur.execute("""SELECT id, firstname, lastname FORM clients ORDER BY lastname""")
for row in cur:
```

```

    print(row)
print('-'*40)
#-----or-----
cur.execute("""SELECT firstname FORM clients""")
for row in cur:
    print(row[0])
print('-'*40)
#-----or-----
cur.execute("""SELECT FORM clients WHERE lastname="Davis" """)
print(cur.fetchall())
print('-'*40)
#-----
#show the table header
#use only the first item of the tuple info
col_name_list = [tup[0] for tup in cur.description]
print("table header")
print(col_name_list)

con.close()

```

► Veritabanı Bağlantısı (Örnek 3)

```

import sqlite3
class Database():
    def __init__(self):
        try:
            self.conn = sqlite3.connect("songs.db")
        except sqlite3.OperationalError:
            exit(1)
        self.cursor = self.conn.cursor()

    def createdatabase(self):
        cmd = "CREATE TABLE allsongs(path VARCHAR(100), name VARCHAR(50), artist VARCHAR(50), album VARCHAR(50))"
        self.cursor.execute(cmd)
        self.conn.commit()

    def insertsongs(self, songdict):
        for song in songdict:
            cmd = "INSERT INTO allsongs(path, name, artist, album) VALUES ('%s', '%s', '%s', '%s')" % (song, songdict[song][0], songdict[song][1], songdict[song][2])
            print "inserting", song + "..."
            self.cursor.execute(cmd)
            self.conn.commit()

    def getsongs():
        songdict = {}
        cmd = "SELECT * FROM allsongs"
        self.cursor.execute(cmd)
        results = self.cursor.fetchall()
        for song in results:
            songdict[song[0]] = (song[1], song[2], song[3])
        return songdict

    def closehandle(self):
        self.conn.commit() #kaydedilmeyen birsey olmasin diye kontrol satiri...
        self.conn.close()

# update, insert, select, delete, where, print, verileri ekliyoruz
# id alani ekle primary key, autoincremental olmuyor. Yeni kayıt geldiginde en son kaydın id'sini çekip 1 arttıyorsun.
# console yap ayrıca pyqt ile yap

```