

Nesneye Yönelimli Programlama – BLM 205

Hafta 11: Tasarım Kalıpları II



Fenerbahçe Üniversitesi

Öğretim Elemanları

Öğretim Üyesi: Dr. Vecdi Emre Levent

Ofis: 311

Email: emre.levent@fbu.edu.tr

Asistan: Arş. Gör. Uğur Özbalkan

Ofis: 307

Email: ugur.ozbalkan@fbu.edu.tr

Asistan: Arş. Gör. Ecenur Alioğulları

Ofis: 307

Email: ecenur.aliogullari@fbu.edu.tr

Ders Planı

- Tasarım Kalıpları II
 - Command
 - Model View Controller (MVC)

Tasarım Kalıpları II

Command

Bu tasarım kalıbında, client çalıştırılması istediği komutları içeren bir obje oluşturur.

Oluşturduğu objeyi komutları yürütecek olan sınıfa besleyerek çalıştırılmasını sağlar.

Tasarım Kalıpları II

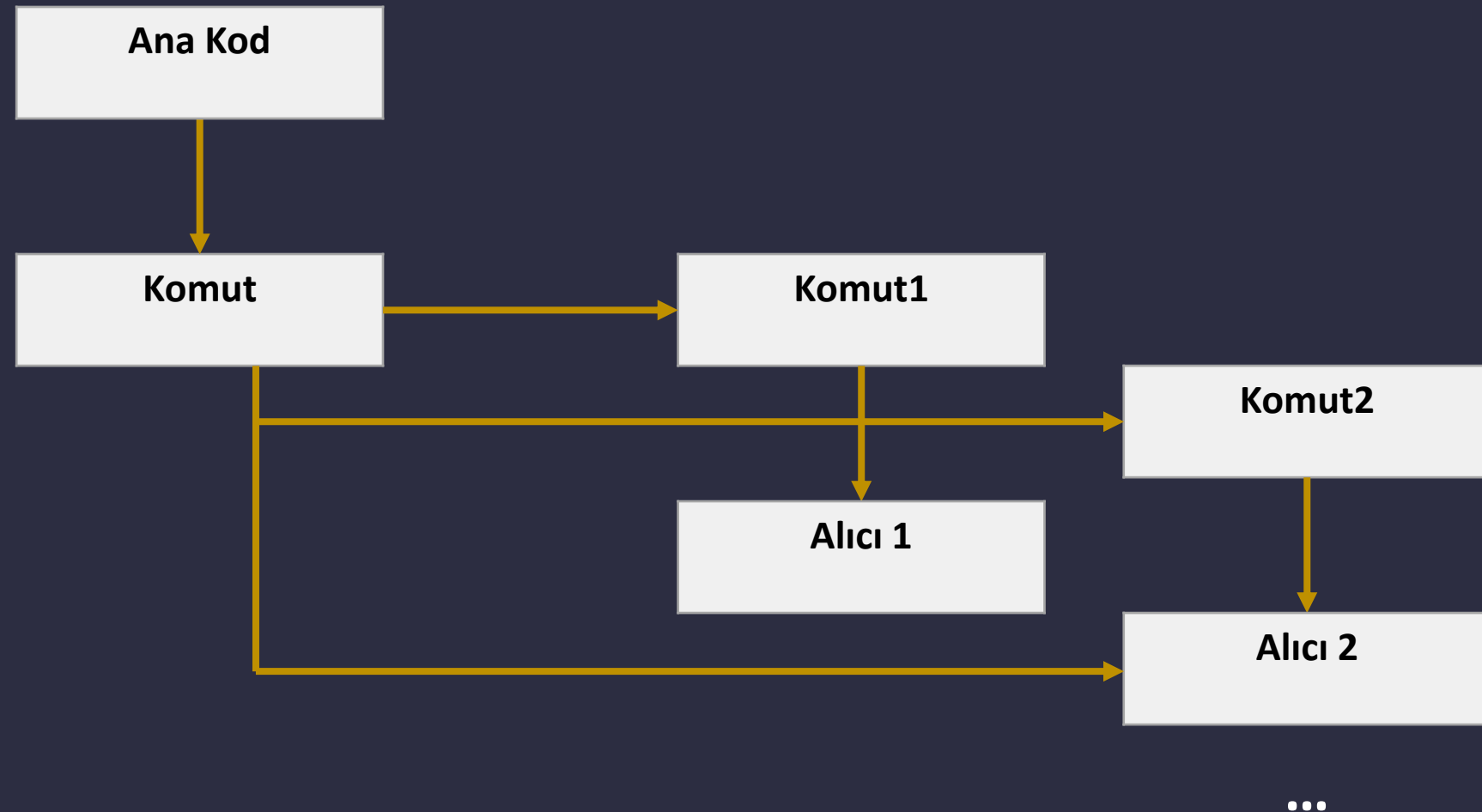
Command

Komutu çağıran client kodunun, komutun yürütme detaylarının bilmesi gerekmemektedir.

Kod yürütücü sınıf, aldığı birden çok komutu organize ederek, sıraya koyabilir.

Tasarım Kalıpları II

Command



Tasarım Kalıpları II

Command

Komutu çağıran client kodunun, komutun yürütme detaylarının bilmesi gerekmemektedir.

Kod yürütücü sınıf, aldığı birden çok komutu organize ederek, sıraya koyabilir.

Tasarım Kalıpları II

Örnek Kod Parçacığı

```
import os
class dosyaAdiDegistirme():
    def __init__(self, ilkAd, sonAd):
        self.ilkAd = ilkAd
        self.sonAd = sonAd

    def calistir(self):
        print("Dosya adi degisiyor: ", self.ilkAd, " -> ", self.sonAd)
        os.rename(self.ilkAd, self.sonAd)

    def geriAl(self):
        print("Dosya adi geri aliniyor: ", self.sonAd, " -> ", self.ilkAd)
        os.rename(self.sonAd, self.ilkAd)

class komutlar():
    def __init__(self):
        self.komutlar = list()

    def calistir(self, komut):
        self.komutlar.append(komut)
        komut.calistir()

    def geriAl(self):
        self.komutlar.pop().geriAl()

komutlar1 = komutlar()
komutlar1.calistir(dosyaAdiDegistirme('test.txt', 'test2.txt'))
komutlar1.calistir(dosyaAdiDegistirme('test2.txt', 'test3.txt'))
komutlar1.geriAl()
```

Çıktı

```
Dosya adi degisiyor: test.txt -> test2.txt
Dosya adi degisiyor: test2.txt -> test3.txt
Dosya adi geri aliniyor: test3.txt -> test2.txt
```


Tasarım Kalıpları II

Model View Controller (MVC)

Birden çok tasarım kalıbının kullanılarak yüksek seviyeli bir mimarisel kalıptır.

GUI (Graphics User Interface) içeren uygulamalarda sıklıkla kullanılır.

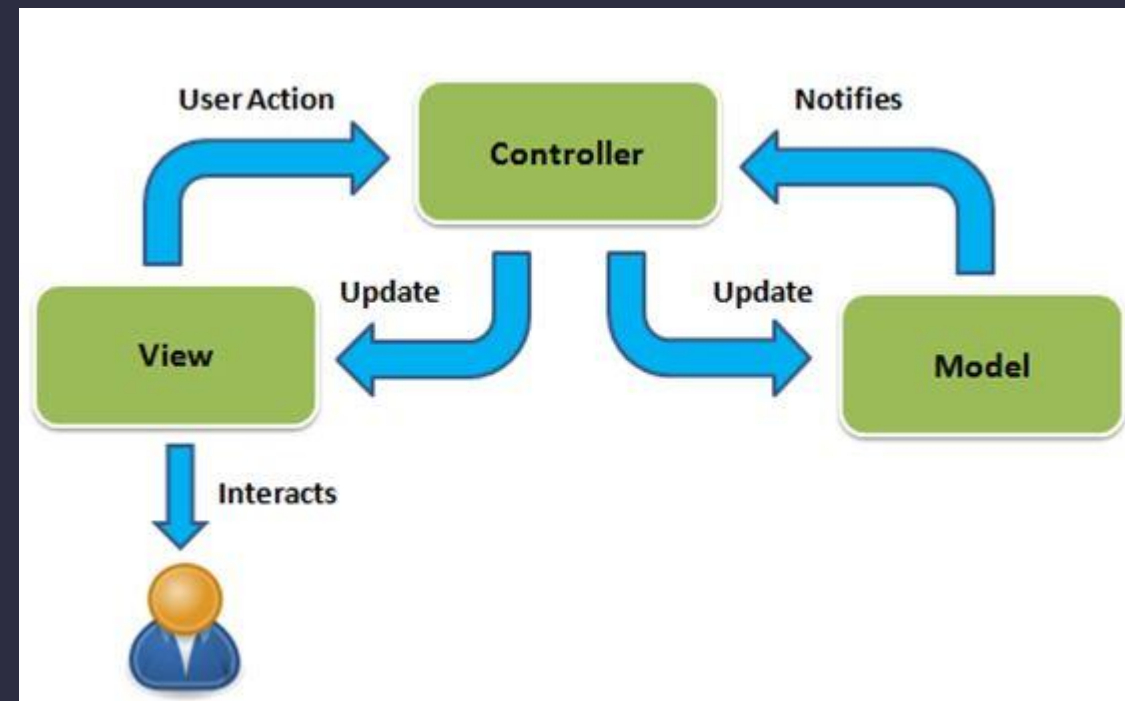
Tasarım Kalıpları II

Model View Controller (MVC)

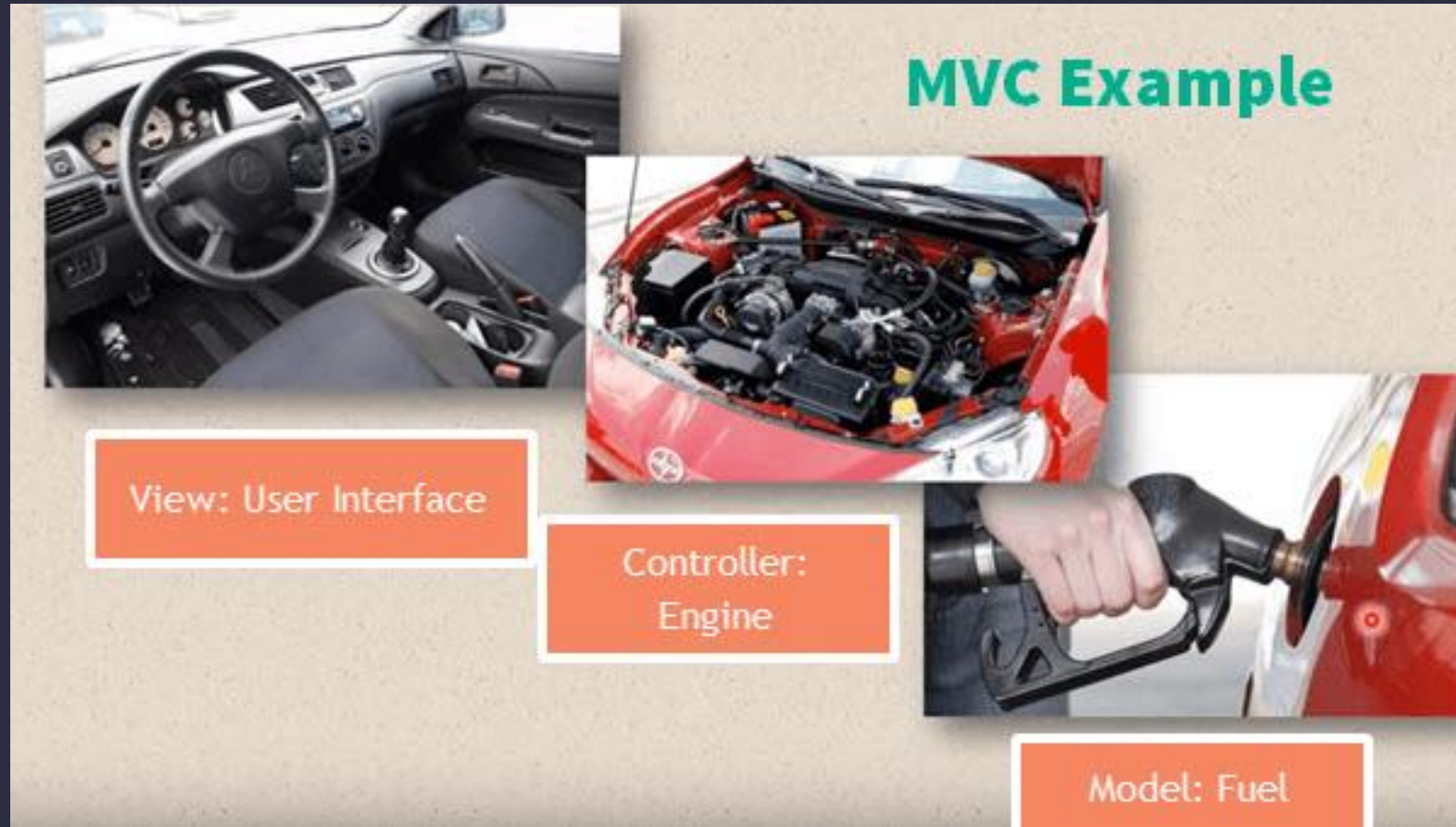
Yazılımın temel bileşenleri olan

- Model: Verilerin tutulduğu veya erişildiği bölümdür.
- View: Kullanıcı ile etkileşime girecek arayüzü oluşturur.
- Controller: Model ve view arasında, modelden alınan verileri işleyip gösteriminin yapılmasını sağlar.

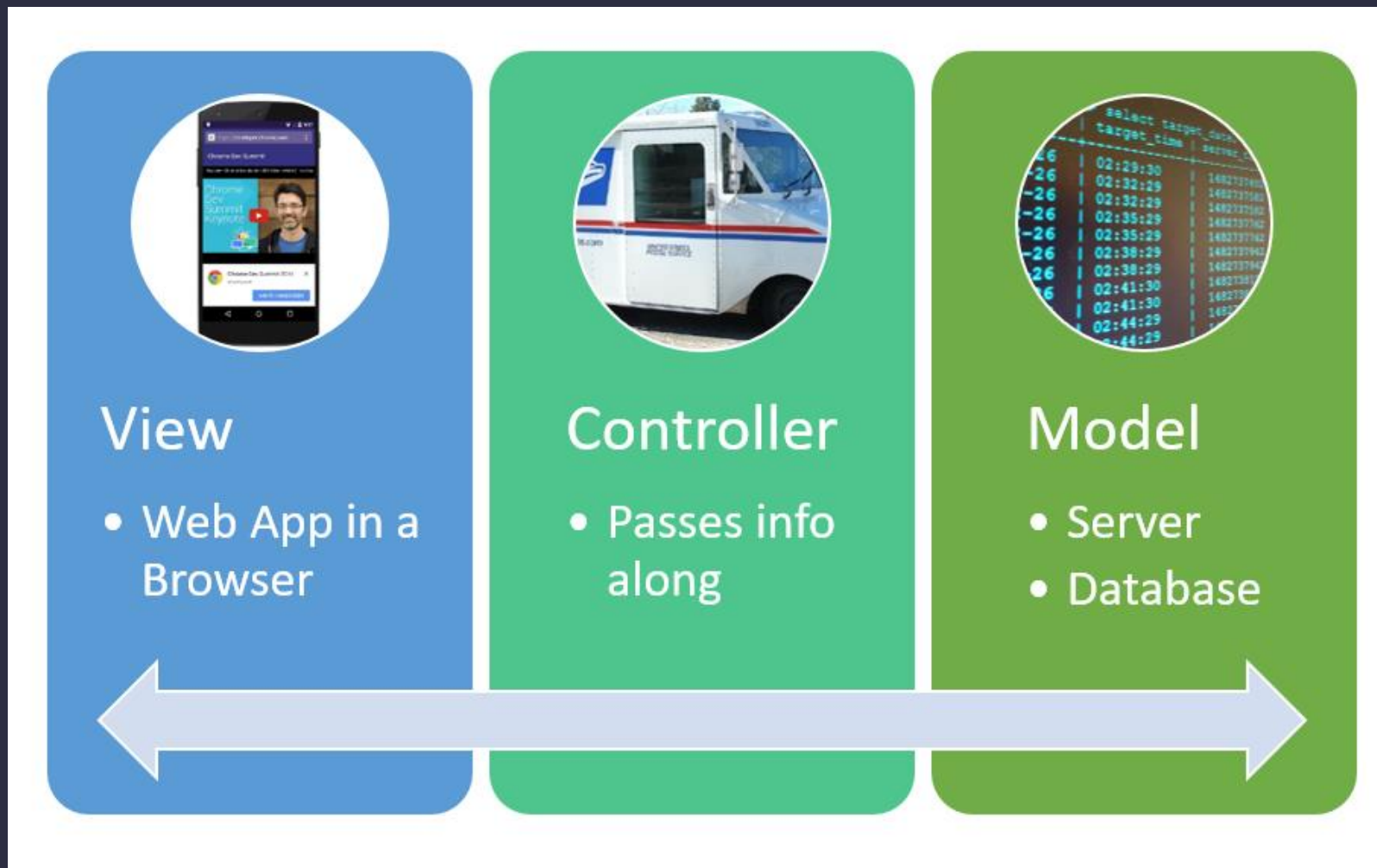
birbirinden ayırmak için kullanılır.



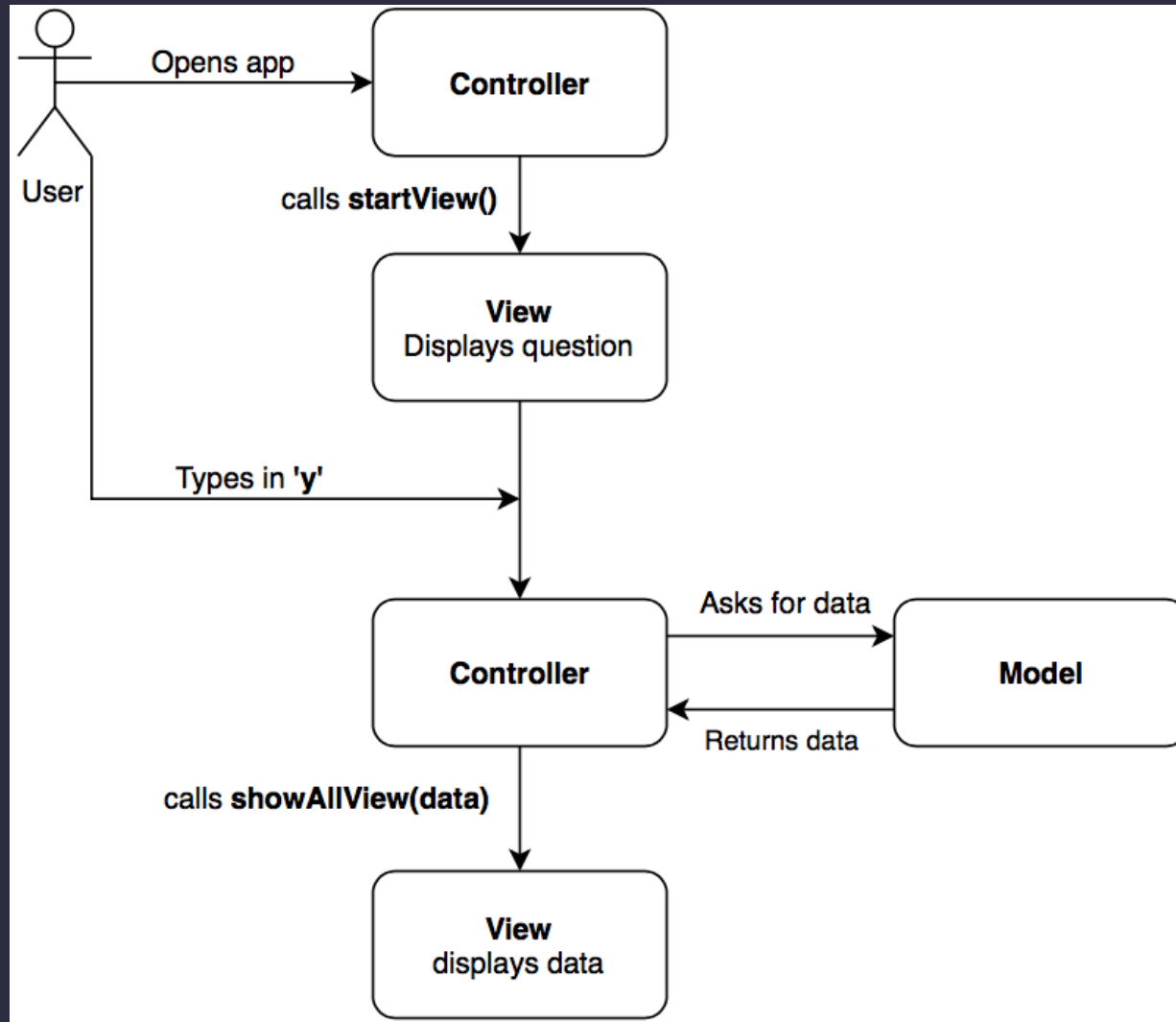
Tasarım Kalıpları II



Tasarım Kalıpları II



Tasarım Kalıpları II



Tasarım Kalıpları II

Örnek 1

db.txt

```
[
    {
        "isim": "Ahmet",
        "soyisim": "Yilmaz"
    },
    {
        "isim": "Mehmet",
        "soyisim": "Yilmaz"
    }
]
```

model.py

```
import json

class kisiler(object):

    def __init__(self, isim, soyisim):
        self.isim = isim
        self.soyisim = soyisim

    def isimDondur(self):
        return ("%s %s" % (self.isim, self.soyisim))

    def tumKisileriGetir():
        veritabani = open('db.txt', 'r')
        kisilerListesi = []
        json_list = json.loads(veritabani.read())
        for item in json_list:
            item = json.dumps(item)
            item = json.loads(item)
            kisi = kisiler(item['isim'], item['soyisim'])
            kisilerListesi.append(kisi)
        return kisilerListesi
```

Tasarım Kalıpları II

Örnek 1

view.py

```
from model import kisiler

def viewTumunuGoster(liste):
    print ('Toplam kullanıcı %i :' % len(liste))
    for kisi in liste:
        print (kisi.isimDondur())

def viewBaslat():
    print ('MVC - Örneği')
    print ('Veri tabanındaki tüm kullanıcıları listelensin mi?[e/h]')

def viewBitir():
    print ('View kapanıyor!')
```

Tasarım Kalıpları II

Örnek 1

controller.py

```
from model import kisiler
import view

def tumunuGoster():
    veritabani = kisiler.tumKisileriGetir()
    return view.viewTumunuGoster(veritabani)

def baslat():
    view.viewBaslat()
    giris = input()
    if giris == 'e':
        return tumunuGoster()
    else:
        return view.viewBitir()

baslat()
```

Çıktı

```
MVC - Ornegi
Veri tabanindaki tum kullanicilari listelensin mi?[e/h]
e
Toplam kullanıcı 2 :
Ahmet Yilmaz
Mehmet Yilmaz
```


Tasarım Kalıpları II

Örnek 2

main.py

```
from controller import Controller

c = Controller()
c.run()
```

model.py

```
import numpy as np

class Model:

    def __init__(self):
        self.xpoint = 200
        self.ypoint = 200
        self.res = None

    def calculate(self):
        x, y = np.meshgrid(np.linspace(-5, 5, self.xpoint), np.linspace(-5, 5, self.ypoint))
        z = np.cos(x ** 2 * y ** 3)
        self.res = {"x": x, "y": y, "z": z}
```

controller.py

```
import tkinter as Tk

from model import Model
from view import View

class Controller:
    def __init__(self):
        self.root = Tk.Tk()
        self.model = Model()
        self.view = View(self.root, self.model)

    def run(self):
        self.root.title("MVC Ornegi")
        self.root.deiconify()
        self.root.mainloop()
```

view.py

```
...
```

Tasarım Kalıpları II

Örnek 2

view.py

```
import tkinter as Tk

from matplotlib.backends.backend_tkagg import
FigureCanvasTkAgg
from matplotlib.figure import Figure
from side_panel import SidePanel

class View:
    def __init__(self, root, model):
        self.frame = Tk.Frame(root)
        self.model = model
        self.fig = Figure(figsize=(7.5, 4), dpi=80)
        self.ax0 = self.fig.add_axes((0.05, .05, .90, .90),
facecolor=(.75, .75, .75), frameon=False)
        self.frame.pack(side=Tk.LEFT, fill=Tk.BOTH,
expand=1)
        self.sidepanel = SidePanel(root)

        self.sidepanel.plotBut.bind("<Button>", self.plot)
        self.sidepanel.clearButton.bind("<Button>",
self.clear)

        self.canvas = FigureCanvasTkAgg(self.fig,
master=self.frame)
        self.canvas.get_tk_widget().pack(side=Tk.TOP,
fill=Tk.BOTH, expand=1)
        self.canvas.draw()
```

view.py

```
def clear(self, event):
    self.ax0.clear()
    self.fig.canvas.draw()

def plot(self, event):
    self.model.calculate()
    self.ax0.clear()
    self.ax0.contourf(self.model.res["x"],
self.model.res["y"], self.model.res["z"])
    self.fig.canvas.draw()
```

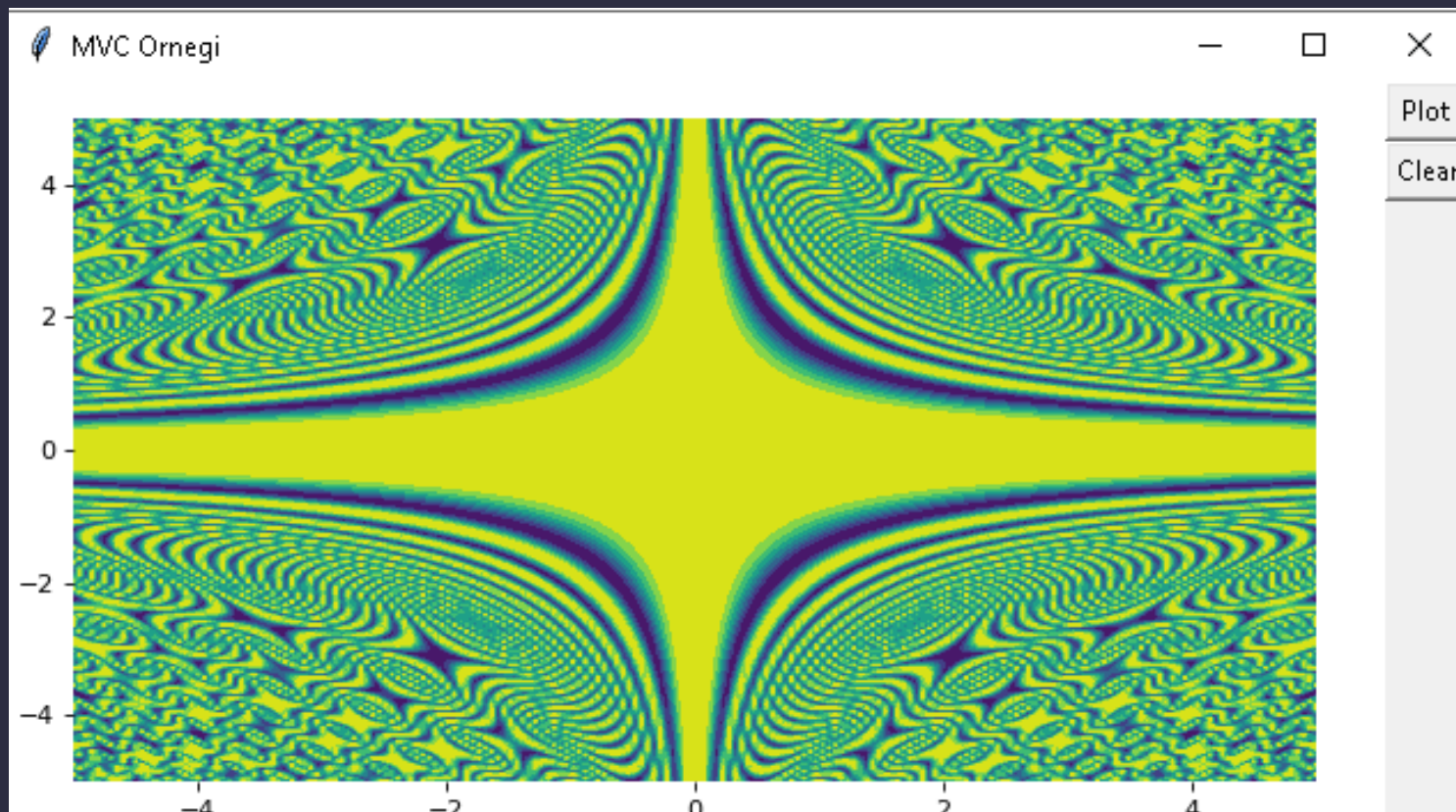
sidepanel.py

```
import tkinter as Tk

class SidePanel():
    def __init__(self, root):
        self.frame2 = Tk.Frame(root)
        self.frame2.pack(side=Tk.LEFT, fill=Tk.BOTH, expand=1)
        self.plotBut = Tk.Button(self.frame2, text="Plot ")
        self.plotBut.pack(side="top", fill=Tk.BOTH)
        self.clearButton = Tk.Button(self.frame2, text="Clear")
        self.clearButton.pack(side="top", fill=Tk.BOTH)
```

Tasarım Kalıpları II

Örnek 2



MVC Uygulaması Çıktısı