

MAIN FONKSİYON

```
import numpy as np

import functions

def main() :

    functions.hilal()
    katsayi = np.zeros(20)

    cikis = 1
    while cikis != 0:
        flag = 1
        print("*** SAYISAL ANALİZ YONTEMLERİ ***\n\n")
        print(
            " 1-NUMERİK KOK BULMA\n 2-NUMERİK INTEGRAL \n 3-NUMERİK
TÜREV \n 4-MATRİSİN İNVERSİ \n 5-DOĞRUSAL DENKLEM TAKIMLARININ
COZUMU \n")
        yontem = int(input("\nSeciminiz : "))
        print("\n")

        if yontem == 1:    #Verilen Denklemin Kokunu Farklı
Yontemlerle Hesapla
            while True:
                print("Kok Bulma Yontemini Seciniz: \n\n")
                print(" 1-Grafik Yontemi\n 2-Bisection Yontemi \n
3-Regula False Yontemi \n 4-Newton Raphson Yontemi")
                secim = int (input("\nSeciminizi giriniz : "))

                if secim == 1:    #GRAFİK YONTEMİ
                    print("\n * GRAFİK YONTEMİ * \n")
                    derece = int( input ("Denklemin maximum
derecesi = "))

                    print("\n")
                    epsilon = float( input("Epsilon (hata payi) =
") )

                    functions.katsayilar(katsayi, derece)
                    deltaX = float( input("Delta x degeri: ") )
                    xSifir = float( input("X'e bir baslangic
degeri veriniz "))

                    functions.grafikYontemi(derece, katsayi,
xSifir, deltaX, epsilon)
                    break
                elif secim == 2:    #BISECTION YONTEMİ
                    print("\n * BISECTIN YONTEMİ *          * YARIYA
BOLME YONTEMİ *\n")
                    derece = int( input("Denklemin maximum
derecesi = "))
```

```

        print("/n")
        functions.katsayilar(katsayi, derece)
        epsilon = float(input("Epsilon (hata payi) =
"))
        while True:
            print("DIKKAT! Denklemin kokunun icinde
bulunabilecegini dusundugunuz bir aralik (a,b) giriniz\n")
            print("DIKKAT! Bu girdiginiz sinir
degerlerinin fonksiyondaki goruntusu birbirinin zit isaretlisi
olmalidir.\n")
            a = float ( input("a= "))
            b = float ( input("b= "))
            if functions.denklemSonuc(a, derece,
katsayi) * functions.denklemSonuc(b, derece, katsayi) > 0 :
                print("\nGirdiginiz sinir degerlerinin
fonksiyondaki goruntusu birbirine zit degildir. Tekrar bir aralik
giriniz.\n\n")
            else :
                break
            functions.BisectionYontemi(derece, katsayi, a,
b, epsilon)
            break
        elif secim ==3 :      #REGULA FALSE YONTEMI
            print("\n * REGULA FALSE YONTEMI *\n")
            derece = int(input("Denklemin maximum derecesi
= "))
            epsilon = float(input("Epsilon (hata payi) =
"))
            functions.katsayilar(katsayi, derece)
            while True:
                print("DIKKAT! Denklemin kokunun icinde
bulunabilecegini dusundugunuz bir aralik (a,b) giriniz\n")
                print("DIKKAT! Bu girdiginiz sinir
degerlerinin fonksiyondaki goruntusu birbirinin zit isaretlisi
olmalidir.\n")
                a = float(input("a= "))
                b = float(input("b= "))
                if functions.denklemSonuc(a, derece,
katsayi) * functions.denklemSonuc(b, derece, katsayi) > 0:
                    print("\nGirdiginiz sinir degerlerinin
fonksiyondaki goruntusu birbirine zit degildir. Tekrar bir aralik
giriniz.\n\n")
                else:
                    break
                functions.regulaFalse(derece, katsayi, a, b,
epsilon)
                break
            elif secim ==4 :      #NEWTON RAPHSON YONTEMI
                print("\n * NEWTON RAPHSON YONTEMI * \n")
                derece = int(input("Denklemin maximum derecesi
= "))

```

```

        print("\n")
        epsilon = float(input("Epsilon (hata payi) =
"))
        functions.katsayilar(katsayi, derece)
        X = float (input ("X'e bir baslangic degeri
veriniz ") )
        functions.newtonRaphsonYontemi(derece,
katsayi, X, epsilon)
        break
    else: #Tekrar yontem secimi yapmaya yonlendir
        print("\n! Lutfen Gecerli Bir Yontem Seciniz !
\n")
elif yontem == 2 : #INTEGRAL ALMA YONTEMLERİ

    print("\n * INTEGRAL * \n")
    print("1-Trapez yontemiyle integral\n")
    print("2-Simpson yontemiyle integral\n")
    integralSecim = int( input("secim = "))
    print("Integrali almak istediginiz aralik ? \n")
    a = float(input("a= "))
    b = float(input("b= "))
    while True:
        derece = int ( input("Denklemin maximum derecesi =
") )

        if derece > 0 :
            break
        functions.katsayilar(katsayi, derece)
        n = int( input ("n degeri= "))
        if integralSecim == 1 :
            if a < 0 :
                negatifIntegral = functions.trapez(a, 0, n,
derece, katsayi)
                integral = negatifIntegral +
functions.trapez(0, b, n, derece, katsayi)
            else :
                integral = functions.trapez(a, b, n, derece,
katsayi)
            print("\n#Trapez Yontemiyle Integralin Sonucu :",
integral)
        elif integralSecim == 2 :
            if a < 0 :
                negatifIntegral = functions.simpson(a, 0, n,
derece, katsayi)
                integral = negatifIntegral +
functions.simpson(0, b, n, derece, katsayi)
            else :
                integral = functions.simpson(a, b, n, derece,
katsayi)
            print("\n#Simpson Yontemiyle Integralin Sonucu :",
integral)
    elif yontem == 3 : #Turev Fonksiyonu
        print("\n * Turev * \n")

```

```

print("Turevini almak istedigin fonksiyon icin :\n")
while True:
    derece = int ( input("Denklemin maximum derecesi =
") )
    if derece > 0 :
        break
    functions.katsayilar(katsayi, derece)
    x = float ( input("Hangi noktada turev almak
istiyorsunuz : "))
    while True:
        deltaX = float(input("Delta x degeri = "))
        if deltaX > 0:
            break
        functions.deneme(katsayi)
        print("\n")
        functions.turev(x,derece,katsayi)
        print("\n")
        functions.ileriFarkTurev(x, deltaX, derece, katsayi)
        print("\n")
        functions.geriFarkTurev(x, deltaX, derece, katsayi)
        print("\n")
        functions.merkeziFarkTurev(x, deltaX, derece, katsayi)
        print("\n")
    elif yontem == 4: #Matrisin inversi
        print("Henuz tamamlanmamistir.. ")
        #break
    elif yontem == 5 : #Gauss Jordan Yontemi
        denklemler = np.zeros([20,20])
        #print("dizi: ", denklemler)

        print("\n * GAUSS JORDAN YONTEMI * \n")

        dsayisi = int ( input("\nCozmek Istedigin Denklem
Sayisi :\n") )
        print("\n\nUYARI ! GIRECEGIN DENKLEMLER ASAGIDAKI
FORMATTA OLMALIDIR !\n")
        print("\naX1 + bX2 + cX3 = d *** ( Buradaki d istenen
sabit sayidir )\n")
        print("\nGirmeniz gereken denklem sayisi : ",
dsayisi)
        i=0
        for i in range(dsayisi) :
            print("\n",i+1, ". Denklemin Katsayilarini
Giriniz .. \n");
            for j in range(dsayisi) :
                print("oldu",i,j)
                denklemler[i,j] = float( input(j+1,"in
katsayisi"))

```

```
        denklemler[i,dsayisi] = float ( input("\nDenklemin
sabit sayisini giriniz : (verilen denklem formatindaki d degeri)
\n" ))
        print("dizi: ", denklemler)
        functions.GaussJordanYontemi(denklemler, dsayisi)
    else :
        print("\n! Lutfen Gecerli Bir Secim Yapin !\n")

        cikis = int ( input("\nCikmak icin 0'a basin, farkli bir
islem yapmak icin baska bir sayi girin..\n\n") )

main()
```