# -\*- coding: utf-8 -\*-

"""Day\_6.ipynb

Automatically generated by Colaboratory.

Original file is located at

https://colab.research.google.com/drive/17XRCl1XjlVEgnG-2sJUmVGyJPWVRgUdM

# Hatalar

\* Programcı Hataları (Error)

\* Program Kusurları (Bug)

\* İstisnalar (Exception)

"""

# error lara bir örnek,SyntaxError.

print "Hello World!"

print(2/0)

#bug lara bir örnek.

sayi1 = input("1. sayıyı giriniz: ")

sayi2 = input("2. sayıyı giriniz: ")

print(sayi1, "+", sayi2, "=", sayi1 + sayi2)

#exception lara bir diğer örnek, ValueError.

sayi3 = int(input("İlk sayı: "))

sayi4 = int(input("İkinci sayı: "))

print(sayi3, "/",sayi4, "=", sayi3/sayi4)

#exception lara bir örnek, ZeroDivisionError.

sayi3 = int(input("İlk sayı: "))

sayi4 = int(input("İkinci sayı: "))

print(sayi3, "/",sayi4, "=", sayi3/sayi4)

"""##Hata Yakalama

try:

> hata alabileceğimiz kısımlar.

except <Hata Adı>:

> hata durumunda yapılacak işlem.

"""

x = "Alan Turing"

int(x)

x="Fethi"

try:

int(x)

except ValueError:

print("Lütfen integer bir değer giriniz!!!")

sayi3 = input("İlk sayı: ")

sayi4 = input("İkinci sayı: ")

try:

sayi3\_int = int(sayi3)

sayi4\_int = int(sayi4)

print(sayi3\_int, "/",sayi4\_int, "=", sayi3\_int/sayi4\_int)

except:

print("Lütfen sayıları düzgün girin!!!")

sayi3 = input("İlk sayı: ")

sayi4 = input("İkinci sayı: ")

try:

sayi3\_int = int(sayi3)

sayi4\_int = int(sayi4)

print(sayi3\_int, "/",sayi4\_int, "=", sayi3\_int/sayi4\_int)

except ZeroDivisionError:

print("Lütfen 2. girdinizi 0 dan farklı bir integer değer giriniz!!!")

sayi3 = input("İlk sayı: ")

sayi4 = input("İkinci sayı: ")

try:

sayi3\_int = int(sayi3)

sayi4\_int = int(sayi4)

print(sayi3\_int, "/",sayi4\_int, "=", sayi3\_int/sayi4\_int)

except ValueError:

print("Lütfen integer bir değer giriniz!!!")

except ZeroDivisionError:

print("Bölen sayıyı 0 dan farklı bir integer değer giriniz!!!")

except:

print("Bilinmeyen br hata oluştu...")

sayi3 = input("İlk sayı: ")

sayi4 = input("İkinci sayı: ")

try:

sayi3\_int = int(sayi3)

sayi4\_int = int(sayi4)

print(sayi3\_int, "/",sayi4\_int, "=", sayi3\_int/sayi4\_int )

except (ValueError, ZeroDivisionError):

print("Bir hata oluştu!!!")

#try/except/as

sayi3 = input("İlk sayı: ")

sayi4 = input("İkinci sayı: ")

try:

sayi3\_int = int(sayi3)

sayi4\_int = int(sayi4)

print(sayi3\_int, "/",sayi4\_int, "=", sayi3\_int/sayi4\_int)

#AS MESAJ DENDİĞİ ZAMAN İLGİLİ HATANIN MESAJINI TUTAR

except (ValueError,ZeroDivisionError) as mesaj:

print("Bir hata var!!!")

print("Hata mesajı: ", mesaj)

#döngü yapılarında hata yakalama

while True:

sayi1 = input("Birinci sayı: (Programdan çıkmak için q tuşuna basın): ")

if sayi1 == "q":

break

sayi2 = input("İkinci sayı: ")

try:

sayi1\_int = int(sayi1)

sayi2\_int = int(sayi2)

print(sayi1\_int, "/", sayi2\_int, "=", sayi1\_int / sayi2\_int)

except (ValueError, ZeroDivisionError) as mesaj:

print("Bir hata oluştu!")

print("Lütfen tekrar deneyin! Hata:",mesaj)

break

"""

fonksiyonlarda hata yakalama

kendi yazdığımız fonksiyonlar içine "raise" komutu ile duruma özel hata

mesajları yayımlayabiliyoruz.

"""

def terscevir(s):

if (type(s) != str):

raise TypeError("Parametrenin tipi hatalı")

else:

return s[::-1]

try:

terscevir(1)

except (ValueError,TypeError) as mesaj:

print("Hata oluştu! Hata:",mesaj)

terscevir("fethi")

terscevir(12)

"""# Nesne Tabanlı Programlama (OOP)

Çevremizde gördüğümüz her varlık birer nesne(object) tanımına girer.

Her nesnenin de kendine ait özellikleri (attribute) ve yaptıkları bir takım işlevler(method) vardır.

Sınıf dediğimiz kavram da, nesneleri üretmemizi sağlayan veri tipleridir.

"""

Araba

-Renk

-Marka

-Kapi

-LPG

class Hayvan():

ses="Miyav"

ayak=4

hayvan1=Hayvan()

print(hayvan1.ses)

hayvan1.ses="Havhav"

print(hayvan1.ses)

hayvan2=Hayvan()

print(hayvan2.ses)

Hayvan.ayak

Hayvan.ses

dir(hayvan1)

"""'''

Her bir objeyi başlangıçta farklı değerlerle oluşturmamız için her bir

objeyi oluştururken objelerin değerlerini göndermemiz gerekiyor.

Bunun için de özel bir method kullanıyoruz: \_\_init()\_\_

init() methodu, yapıcı(constructor ) fonksiyon olarak tanımlanmaktadır.

Objelerimiz oluşturulurken otomatik olarak ilk çağrılan fonksiyondur.

Bu metodu özel olarak tanımlayarak objelerimizi farklı değerlerle oluşturabiliriz.

'''

"""

#constructor / yapıcı

#sınıf içersisinde sadece metod ve değişkenler olabilir

#sınıf içerisindeki metodlar alsında sınıfın yapmıuş olduğu aksiyonlardır

#sınıf içerisindeki değişkenler ise aslında sınıfın özellikleridir

class Hayvan():

#ses=""

#ayak=0

def \_\_init\_\_(self,ses,ayak):

print("Constructor çağrıldı!")

self.ses=ses

self.ayak=ayak

def Yazdir(self):

print("Ses:",self.ses," Ayak:",self.ayak)

tavuk=Hayvan("Gıdak",2)

tavuk.Yazdir()

kedi=Hayvan("Miyav",4)

kedi.Yazdir()

"""

self= Tanımlanan sınıfta yer alan herhangi bir değişkene veya

diğer method’lara erişmek için self kelimesini kullanıyoruz.

"""

class Person():

def \_\_init\_\_(self, personName, personAge):

self.name = personName

self.age = personAge

def welcomePerson(self):

print("Merhaba " + self.name)

def showAge(self):

print(self.age)

fethi=Person("Fethi",24)

fethi.welcomePerson()

fethi.showAge()

person1 = Person ("Asli", 28)

person1.welcomePerson()

person1.showAge()

#ses

#ayak

#renk

class Hayvan():

def \_\_init\_\_(self,ses,ayak,renkler):

self.ses=ses

self.ayak=ayak

self.renkler=renkler

def Yazdir(self):

print("Ses:",self.ses," Ayak:",self.ayak," Renkler:",self.renkler)

def RenkEkle(self,renkler):

self.renkler.extend(renkler)

self.renkler=list(set(self.renkler))

print(type(self.renkler))

kedi=Hayvan("Miyav",4,["Siyah","Beyaz","Gri"])

kedi.Yazdir()

kedi.RenkEkle(["Turuncu","Siyah","Mavi"])

kedi.Yazdir()

kedi.RenkEkle(["Sarı"])

kedi.Yazdir()

"""

hataliKedi=Hayvan("Havhav",7,("Siyah","Beyaz"))

hataliKedi.RenkEkle(("Mavi","Yeşil"))

"""

class Hayvan2():

isim = input("İsmini giriniz ")

ses= input("nasıl ses çıkardığını giriniz ")

ayak\_sayısı = int(input("Ayak sayısını giriniz "))

kullanıcıSes=input("ses gir:")

ayak=int(input("ayak sayısı gir:"))

renkler=input("renkleri gir aralarında boşluk olsun:")

print(renkler.split())

yeniHayvan=Hayvan(kullanıcıSes,ayak,renkler.split())

yeniHayvan.Yazdir()

class Person3():

def \_\_init\_\_(self, name, age):

self.name = name

self.age = age

self.language = []

def welcomePerson(self):

print("Merhaba " + self.name)

def showAge(self):

print(self.age)

def addLang (self, new\_lang):

print("Yeni dil ekleniyor..")

self.language.append(new\_lang)

def showInfo(self):

print("{} adlı kişinin yaşı {} ve bildiği diller: ".format(self.name,self.age))

for i in self.language:

print(i)

person3 = Person3("asli",28)

person3.welcomePerson()

person3.showAge()

person3.addLang("german")

person3.showInfo()

person3.addLang("english")

person3.showInfo()