

# PROJE 1: ÖĞRENCİ NOT SİSTEMİ

PYBTCMP-10 / 10-Proje-Grubu-2

## Proje İsterleri:

- Kendinize bir ders belirleyiniz. (Matematik, Fizik, Lineer Cebir vb.)
- Not aralığınızı oluşturunuz (100-80  $\Rightarrow$  A, 79-70  $\Rightarrow$  B vb.)
- Öğrenci Bilgilerini (Ad, Soyad, Okul No, sınav puanı) girebileceğiniz ve bu bilgilerin tutulabileceği bir sistem oluşturunuz.
- Girilen bilgilerden yola çıkarak öğrencinin dersi geçip geçmediğini göstermesi gerekmektedir.
- Öğrenci dersi geçti ise öğrencinin bilgilerinin tutulduğu alana “Geçti” yazısı, öğrenci dersi geçemedi ise “kaldı” yazısını göstermesi gerekmektedir.
- Notları girilen öğrencilerden dersi geçenleri ve geçmeyenleri gösteren bir Dataframe oluşturunuz.
- Oluşturulan Dataframe’i Excel tablosuna dönüştürünüz.

## Proje:

- Öncelikle Python'da Dataframe yapısını kullanabilmek için **"pip install pandas"** komutunu terminalimize yazıp Pandas kütüphanesinin kurulumunu yaptık
- Pandas kütüphanesini projemiz kapsamında kullanabilmek için **"import pandas as pd"** komutuyla projemize dâhil ettik

- Projemizin ilk aşamasında kullanıcımızdan ders bilgisini aldık;

```
lesson = input("\n" + "Lesson: ")
```

- Sonraki aşamada ad, soyad, okul no ve sınav puanı bilgilerini kaydedeceğimiz boş listemizi tanımladık;

```
students = list()
```

- Not aralığımızı oluşturduk;  
(100-90 ⇒ AA, 89-85 ⇒ BA, 84-80 ⇒ BB, 79-70 ⇒ CB, 69-60 ⇒ CC, 59-50 ⇒ DC, 49-45 ⇒ DD, 44-40 ⇒ FD, 39-0 ⇒ FD)

- Oluşturduğumuz boş listemize öğrencileri kaydedebilmek için **"class"** yapısını kullandık;

```
class Student():  
    def __init__(self, name, surname, school_number, grade, letter_grade,  
lesson, status):  
        self.name = name  
        self.surname = surname  
        self.school_number = school_number  
        self.grade = grade  
        self.letter_grade = letter_grade  
        self.lesson = lesson  
        self.status = status
```

- Öğrenci bilgilerini kullanıcıdan alacağımız döngüyü kullanıcı isteği ile sonlandırabilmek için **"control = 1"** ve tanımladığımız boş listemize öğrencileri sırası ile aktarabilmek için **"students\_index = 0"** tanımlamaları yaptık.
- While döngümüzü **"True"** koşuluyla başlattık. Döngü içerisinde control değişkeninin 1'e eşit olması durumunda kullanıcıdan alacağımız öğrenci ad, soyad, okul numarası ve not bilgilerine göre Student(): class'ının \_\_init\_\_ metodunu kullanarak öğrenci nesnelerimizi oluşturduk. Listemize index'lere göre oluşturduğumuz öğrenci nesnelerini .insert() komutuyla atadık ve student\_index değerimizi , control değişkenini her yeni öğrenci ekleme isteği için kullanıcı tarafından kontrol mekanizması olarak kullandık;

```
while True:
    if control == 1:
        student_name = input("\n" + "Name: ")
        student_surname = input("Surname: ")
        student_school_number = input("School number: ")
        student_grade = int(input("Grade: "))

        student_letter_grade, student_status =
set_student_status(student_grade)
        student_temp = Student(student_name, student_surname,
student_school_number, student_grade, student_letter_grade,
lesson, student_status)
        students.insert(students_index, student_temp)
        students_index += 1
        control = int(input("\n" + "[Press '1' to Continue or
Press '0' to Terminate]: "))
    else:
        break
```

- Bilgilerinin girişi yapılan öğrenciye ait sınav notunun belirlediğimiz not aralıklarından hangi aralıkta bulunduğuna göre dersi geçti-kaldı durumuna ve harf notuna karar verebilmek için fonksiyonumuzu yazdık, fonksiyon sonucunda elde ettiğimiz harf notu ve geçti/kaldı durum bilgisini bir önceki kod bloğunda bulunan döngü içerisinde tanımladığımız değişkenlerimize atadık;

```
def set_student_status(grade):  
    if grade >= 90:  
        letter_grade = 'AA'  
        status = 'Pass'  
    elif grade >= 85 and grade <= 89:  
        letter_grade = 'BA'  
        status = 'Pass'  
    elif grade >= 80 and grade <= 84:  
        letter_grade = 'BB'  
        status = 'Pass'  
    elif grade >= 70 and grade <= 79:  
        letter_grade = 'CB'  
        status = 'Pass'  
    elif grade >= 60 and grade <= 69:  
        letter_grade = 'CC'  
        status = 'Pass'  
    elif grade >= 50 and grade <= 59:  
        letter_grade = 'DC'  
        status = 'Conditional Pass'  
    elif grade >= 45 and grade <= 49:  
        letter_grade = 'DD'  
        status = 'Conditional Pass'  
    elif grade >= 40 and grade <= 44:  
        letter_grade = 'FD'  
        status = 'Fail'  
    elif grade <= 39:  
        letter_grade = 'FF'  
        status = 'Fail'  
    return letter_grade, status
```

- Excel'e aktaracağımız bilgileri anlaşılır şekilde ifade edebilmek için sütunlara başlık atamaları yaptık ve buna göre Dataframe oluşturduk;

```
my_columns = ['Name', 'Surname', 'Number', 'Grade', 'Letter  
Grade', 'Status', 'Lesson']  
df = pd.DataFrame(columns = my_columns)
```

- For döngüsü ile öğrenci bilgilerinin Excel’de karşılık gelen alanlara atamasını yaptık;

```
for student in students:
    df = df.append(
        {
            'Name': student.name,
            'Surname': student.surname,
            'Number': student.school_number,
            'Grade': student.grade,
            'Letter Grade': student.letter_grade,
            'Status': student.status,
            'Lesson': student.lesson
        }, ignore_index = True
    )
```

- Son olarak Excel’e aktardık;

```
df.to_excel("student_grading_system.xlsx")
```

### Terminal Ekran Görüntüsü:

```
Lesson: Biology

Name: Ahmet
Surname: Korkmaz
School number: 165
Grade: 90

[Press '1' to Continue or Press '0' to Terminate]: 1

Name: Zehra
Surname: Mutlu
School number: 172
Grade: 39

[Press '1' to Continue or Press '0' to Terminate]: 1

Name: Mert
Surname: Arslan
School number: 175
Grade: 45

[Press '1' to Continue or Press '0' to Terminate]: 0

[{'Name': 'Ahmet', 'Surname': 'Korkmaz', 'School number': '165', 'Grade': 90, 'Letter grade': 'AA', 'Lesson': 'Biology', 'Status': 'Pass'}, {'Name': 'Zehra', 'Surname': 'Mutlu', 'School number': '172', 'Grade': 39, 'Letter grade': 'FF', 'Lesson': 'Biology', 'Status': 'Fail'}, {'Name': 'Mert', 'Surname': 'Arslan', 'School number': '175', 'Grade': 45, 'Letter grade': 'DD', 'Lesson': 'Biology', 'Status': 'Conditional Pass'}]
c:\Users\ELIF\Desktop\Student Grading System\main.py:89: FutureWarning: The frame.append method is deprecated and will be removed from pandas in a future version. Use pandas.concat instead.
  df = df.append(
c:\Users\ELIF\Desktop\Student Grading System\main.py:89: FutureWarning: The frame.append method is deprecated and will be removed from pandas in a future version. Use pandas.concat instead.
  df = df.append(
c:\Users\ELIF\Desktop\Student Grading System\main.py:89: FutureWarning: The frame.append method is deprecated and will be removed from pandas in a future version. Use pandas.concat instead.
  df = df.append(

   Name  Surname  Number  Grade  Letter Grade      Status  Lesson
0 Ahmet  Korkmaz   165     90         AA        Pass  Biology
1 Zehra   Mutlu   172     39         FF        Fail  Biology
2 Mert   Arslan   175     45         DD  Conditional Pass  Biology
```

### Excel Ekran Görüntüsü:

student\_grading\_system.xlsx - Excel (Ürün Etkinleştirilemedi)

DOSYA GİRİŞ EKLE SAYFA DÜZENİ FORMÜLLER VERİ GÖZDEN GEÇİR GÖRÜNÜM Microsoft hesabı

Yapıştır Pano Yazı Tipi Hizalama Sayı Stiller Hücreler Düzenleme

F1 : Letter Grade

	A	B	C	D	E	F	G	H	I
1		Name	Surname	Number	Grade	Letter Grade	Status	Lesson	
2	0	Ahmet	Korkmaz	165	90	AA	Pass	Biology	
3	1	Zehra	Mutlu	172	39	FF	Fail	Biology	
4	2	Mert	Arslan	175	45	DD	Conditional Pass	Biology	
5									
6									
7									
8									
9									
10									

Sheet1

HAZIR %100