**1. User Class**

* **Imports:**

import json

from data\_handler import DataHandler

* **Attributes:**
  + name : Kullanıcının adı.
  + surname: Kullanıcının soyadı.
  + student\_no: Kullanıcının öğrenci numarası.
  + attempts: Kullanıcının sınav giriş sayısı.
  + exam\_status: Kullanıcının sınav durumu (Geçti/Kaldı).
  + last\_attempt\_date: Son sınav tarihi.
  + total\_score: Kullanıcının toplam puanı.
* **Methods:**
  + login(self, name, surname): Kullanıcı girişini doğrular.
  + update\_attempts(self): Kullanıcının giriş hakkını günceller.
  + get\_attempts(self): Kalan giriş hakkını döndürür.
  + save\_user\_data(self): Kullanıcı bilgilerini JSON dosyasına kaydeder.
  + check\_attempts(self): Kullanıcının giriş hakkı olup olmadığını kontrol eder.
  + record\_exam\_score(self, score): Kullanıcının sınav puanını kaydeder.
* **Relationships:** DataHandler sınıfı ile kullanıcı verilerini JSON olarak kaydetmek ve almak için ilişkilidir.

**2. Question Class**

* **Imports:**

import random

* **Attributes:**
  + question\_text: Sorunun metni.
  + options: Soru seçenekleri.
  + correct\_answer: Doğru cevap.
  + question\_type: Soru türü (Doğru/Yanlış, Tek/Çoktan Seçmeli).
  + section: Sorunun bulunduğu bölüm.
  + points: Soruya atanmış puan değeri.
* **Methods:**
  + check\_answer(self, user\_answer): Kullanıcının cevabını kontrol eder.
  + validate\_answer\_format(self, user\_answer): Cevap formatını doğrular.
  + assign\_points(self, points): Soruya puan atanmasını sağlar.
* **Relationships:** Exam ve QuestionBank sınıflarıyla doğrudan ilişkilidir.

**3. Exam Class**

* **Imports:**

from user import User

from timer import Timer

from result import Result

from question import QuestionBank

* **Attributes:**
  + user: User sınıfından bir örnek.
  + timer: Timer sınıfından bir örnek.
  + remaining\_attempts: Kalan sınav hakkı.
  + exam\_status: Sınav geçme durumu.
  + section\_scores: Bölüm bazında puanlar.
  + total\_score: Genel puan.
* **Methods:**
  + start\_exam(self): Sınavı başlatır.
  + display\_question(self, question): Soruyu gösterir.
  + get\_next\_question(self): Bir sonraki soruyu alır.
  + submit\_answer(self, question, user\_answer): Kullanıcı cevabını işler.
  + calculate\_results(self): Sınav sonuçlarını hesaplar.
  + calculate\_section\_scores(self): Bölüm bazında puanları hesaplar.
  + calculate\_overall\_score(self): Genel başarı puanını hesaplar.
* **Relationships:** User, Timer, Result, QuestionBank ile doğrudan ilişkilidir.

**4. Timer Class**

* **Imports:**

import time

* **Attributes:**
  + time\_limit: Sınav süresi (dakika).
  + time\_left: Kalan süre.
* **Methods:**
  + start\_timer(self): Zamanlayıcıyı başlatır.
  + check\_time(self): Süreyi kontrol eder.
  + stop\_timer(self): Zamanlayıcıyı durdurur.
* **Relationships:** Exam sınıfı ile sınav süresi yönetimini sağlar.

**5. DataHandler Class**

* **Imports:**

import json

* **Attributes:** Yok
* **Methods:**
  + read\_json(self, file\_name): JSON dosyasını okur.
  + write\_json(self, file\_name, data): JSON dosyasına veri yazar.
  + validate\_json(self, file\_name): JSON dosyasının geçerliliğini kontrol eder.
* **Relationships:** User, Question, Result sınıfları ile veri kaydetme ve alma işlemlerinde ilişkilidir.

**6. Result Class**

* **Imports:**

from exam import Exam

* **Attributes:**
  + section\_scores: Bölüm bazında başarı puanları.
  + overall\_score: Genel başarı puanı.
* **Methods:**
  + calculate\_section\_scores(self, section\_results): Bölüm başarı puanlarını hesaplar.
  + calculate\_overall\_score(self): Genel başarı puanını hesaplar.
  + display\_results(self, results): Sonuçları gösterir.
* **Relationships:** Exam ve User sınıfları ile sonuçları kaydetme ve görüntüleme için bağlantılıdır.

**7. QuestionBank Class**

* **Imports:**

from question import Question

import random

* **Attributes:**
  + questionsBank: Soru listesi.
* **Methods:**
  + add\_question(self, section, question): Yeni soru ekler.
  + get\_question(self, section, index): Belirli bir soruyu getirir.
  + get\_random\_questions(self, section, num\_questions): Rastgele soruları seçer.
* **Relationships:** Question sınıfını kullanarak soruları depolar ve Exam sınıfına soruları sağlar.

**8. QuizApp Class (Tkinter Arayüz)**

* **Imports:**

import tkinter as tk

from tkinter import messagebox

from user import User

from exam import Exam

from data\_handler import DataHandler

from result import Result

from question import QuestionBank

from timer import Timer

* **Attributes:**
  + user: Mevcut kullanıcı.
  + exam: Mevcut sınav.
  + timer: Sınav zamanlayıcısı.
* **Methods:**
  + create\_login\_screen(self): Giriş ekranını oluşturur.
  + start\_exam(self): Sınavı başlatır.
  + create\_exam\_screen(self): Soru gösterim ekranını oluşturur.
  + display\_next\_question(self): Sıradaki soruyu gösterir.
  + submit\_answer(self): Kullanıcının cevabını işler ve bir sonraki soruya geçer.
  + end\_exam(self): Sınavı sonlandırır ve sonuçları gösterir.
  + clear\_screen(self): Ekranı temizler.
* **Relationships:** User, Exam, QuestionBank, Timer, Result sınıfları ile doğrudan ilişkilidir.

Bu yapı, kullanıcı ve sınav sürecini yönetmek için tüm sınıfların birbiriyle ilişkili olarak çalışmasını sağlar. QuizApp sınıfı kullanıcı arayüzünü yönetirken, diğer sınıflar sınav işlemleri ve veri yönetimini üstlenir.