

**KASTAMONU ÜNİVERSİTESİ**  
**MÜHENDİSLİK VE MİMARLIK FAKÜLTESİ**  
**NESNEYE YÖNELİK PROGRAMLAMA DERSİ**  
**KURS VE EĞİTMEN YÖNETİM SİSTEMİ**  
**FİNAL PROJESİ RAPORU**

**Ad:** Halil İbrahim  
**Soyad:** Tosya  
**Öğrenci Numarası:** 234410070

**Projenin Amacı**

Bu proje, bir eğitim yönetim sistemi geliştirerek farklı türdeki kursların (dil, programlama, resim, müzik) ve bu kursları veren eğitmenlerin bilgilerini yönetmeyi amaçlamaktadır. Sistem, nesne yönelimli programlama (OOP) prensiplerine uygun olarak tasarlanmış olup, kurs ve eğitmen bilgilerinin eklenmesi, silinmesi ve görüntülenmesi gibi işlevleri desteklemektedir. Proje, esnek ve genişletilebilir bir yapı sunarak farklı kurs türleri ve eğitmen uzmanlık alanlarını kolayca yönetmeyi hedefler.

**Teknik Yapı**

- Programlama Dili: C# (Windows Forms).
- Veritabanı: Microsoft SQL Server.
- Veri Erişimi: ADO.NET (SqlDataAdapter, SqlCommand).
- Geliştirme Ortamı: Visual Studio 2022.
- İşletim Sistemi: Windows 11.

## Projenin İşlevleri

### 1. Kurs Yönetimi:

- Farklı kurs türleri (Dil, Programlama, Resim, Müzik) için kurs ekleme, silme ve bilgi görüntüleme.
- Her kurs türüne özgü özelliklerin (örneğin, dil kursu için seviye, programlama kursu için zorluk) kaydedilmesi ve yönetimi.

### 2. Eğitimci Yönetimi:

- Farklı uzmanlık alanlarına sahip eğitimcilerin (Dil, Programlama, Resim, Müzik) eklenmesi, silinmesi ve bilgilerinin görüntülenmesi.
- Eğitimcilere özgü özelliklerin (örneğin, dil eğitmeni için bildiği diller, müzik eğitmeni için çalabildiği enstrümanlar) kaydedilmesi.

### 3. Nesne Yönelimli Programlama Desteği:

- Kalıtım (inheritance) ve polimorfizm kullanılarak kurs ve eğitimci sınıflarının esnek bir şekilde tanımlanması.
- Her kurs ve eğitimci türü için özelleştirilmiş bilgi döndürme işlevselliği.

### 4. Form Tabanlı Kullanıcı Arayüzü:

- Kullanıcıların kurs ve eğitimci bilgilerini ekleyip silmesine olanak tanıyan formlar.
- Nesne oluşturma yoluyla veri girişi ve yönetimi.

## Projenin Yapısı

Proje, nesne yönelimli programlama prensiplerine dayalı olarak tasarlanmıştır ve aşağıdaki temel bileşenlerden oluşur.

### Kurs Sınıfları

- **Temel Sınıf:** Kurs
  - Soyut (abstract) bir sınıf olup, tüm kurs türleri için ortak özellikleri içerir: KursID, KursAdi, KursTuru, ve Egitmen.
  - KursBilgisi() metodu, kursa ait temel bilgileri döndürmek için sanal (virtual) olarak tanımlanmıştır.
- **Türetilmiş Sınıflar:**
  - DilKursu: Dil kurslarına özgü Dil ve Seviye özellikleri içerir. KursTuru "Dil" olarak sabitlenmiştir.
  - ProgramlamaKursu: Programlama kurslarına özgü ProgramlamaDili ve Zorluk özellikleri içerir. KursTuru "Programlama" olarak sabitlenmiştir.

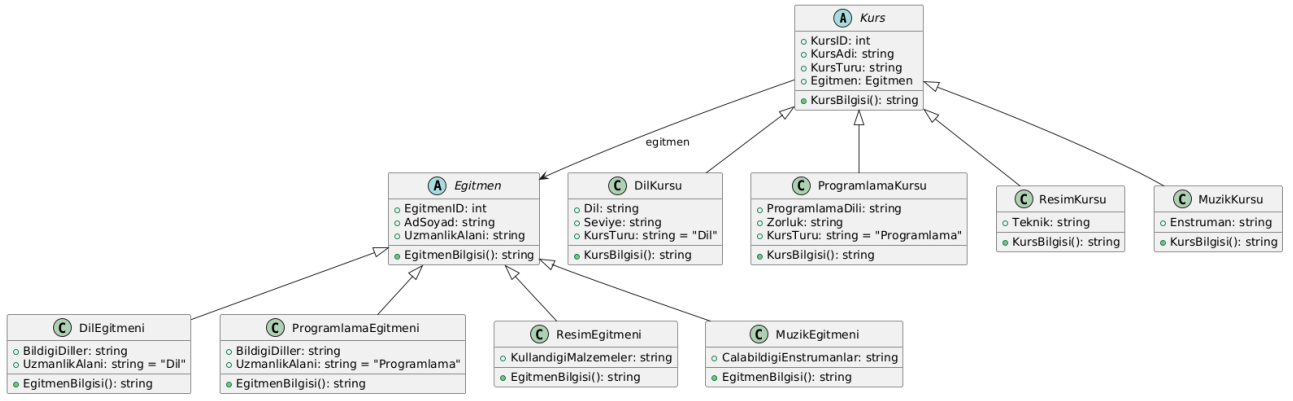
- ResimKursu: Resim kurslarına özgü Teknik özelliği içerir.
- MuzikKursu: Müzik kurslarına özgü Enstruman özelliği içerir.
- Her türetilmiş sınıf, KursBilgisi() metodu ile kendine özgü bilgileri döndürür.

## Eğitmen Sınıfları

- **Temel Sınıf:** Egitmen
  - Soyut bir sınıf olup, tüm eğitmen türleri için ortak özellikleri içerir: EgitmenID, AdSoyad, ve UzmanlikAlani.
  - EgitmenBilgisi() metodu, eğitmene ait temel bilgileri döndürmek için sanal olarak tanımlanmıştır.
- **Türetilmiş Sınıflar:**
  - DilEgitmeni: Dil eğitmenlerine özgü BildigiDiller özelliği içerir. UzmanlikAlani "Dil" olarak sabitlenmiştir.
  - ProgramlamaEgitmeni: Programlama eğitmenlerine özgü BildigiDiller özelliği içerir. UzmanlikAlani "Programlama" olarak sabitlenmiştir.
  - ResimEgitmeni: Resim eğitmenlerine özgü KullandigiMalzemeler özelliği içerir.
  - MuzikEgitmeni: Müzik eğitmenlerine özgü CalabildigiEnstrumanlar özelliği içerir.
  - Her türetilmiş sınıf, EgitmenBilgisi() metodu ile kendine özgü bilgileri döndürür.

## Nesne Yönelimli Programlama Özellikleri

- **Kalıtım (Inheritance):** Kurs ve Egitmen sınıfları, farklı türler için ortak özellikleri tanımlayan soyut temel sınıflardır. Türetilmiş sınıflar, bu temel sınıflardan özellikleri miras alır.
- **Polimorfizm:** KursBilgisi() ve EgitmenBilgisi() metotları, her türetilmiş sınıf için farklı bir şekilde uygulanarak polimorfik davranış sağlar.
- **Kapsülleme (Encapsulation):** Özellikler (get, set) kullanılarak veri erişimi kontrol edilir.
- **Soyutlama (Abstraction):** Soyut sınıflar (Kurs ve Egitmen), yalnızca gerekli arayüzleri tanımlayarak gereksiz detayları gizler.



Şekil: UML Diyagramı

## Veritabanı Yapısı

Sistem, SQL Server üzerinde KursEgitmenYonetim adında bir veritabanına dayanır ve aşağıdaki üç tablodan oluşur:

- **Egitmen:**

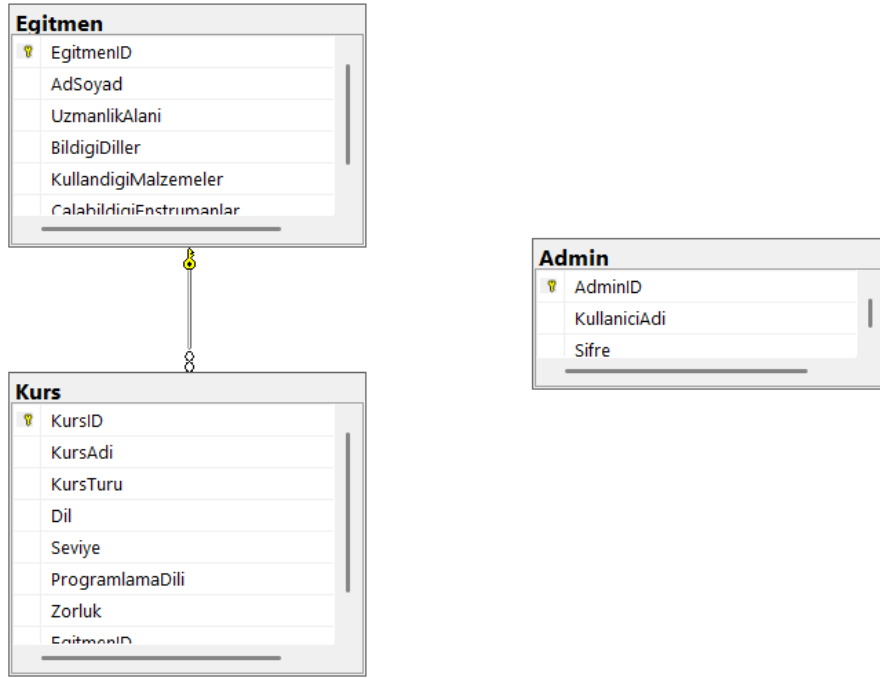
- EgitmenID (INT, PK, IDENTITY): Eğitmenin benzersiz kimliği.
- AdSoyad (NVARCHAR(100)): Eğitmenin tam adı.
- UzmanlikAlani (NVARCHAR(50)): Eğitmenin uzmanlık alanı (örneğin, Dil, Programlama, Resim, Müzik).
- BildigiDiller (NVARCHAR(200)): Dil veya programlama eğitmenlerinin bildiği diller.
- KullandigiMalzemeler (NVARCHAR(200)): Resim eğitmenlerinin kullandığı malzemeler.
- CalabildigiEnstrumanlar (NVARCHAR(200)): Müzik eğitmenlerinin çalabildiği enstrümanlar.

- **Kurs:**

- KursID (INT, PK, IDENTITY): Kursun benzersiz kimliği.
- KursAdi (NVARCHAR(100)): Kursun adı.
- KursTuru (NVARCHAR(50)): Kursun türü (Dil, Programlama, Resim, Müzik).
- Dil (NVARCHAR(50)): Dil kurslarına özgü dil bilgisi.
- Seviye (NVARCHAR(50)): Dil kurslarına özgü seviye bilgisi.
- ProgramlamaDili (NVARCHAR(50)): Programlama kurslarına özgü dil bilgisi.
- Zorluk (NVARCHAR(50)): Programlama kurslarına özgü zorluk seviyesi.
- EgitmenID (INT, FK): Kursu veren eğitmenin kimliği, Egitmen tablosuna referans.

- **Admin:**

- AdminID (INT, PK, IDENTITY): Yöneticinin benzersiz kimliği.
- KullaniciAdi (NVARCHAR(50), UNIQUE): Yöneticinin kullanıcı adı.
- Sifre (NVARCHAR(100)): Yöneticinin şifresi (güvenlik için hashlenmiş olarak saklanabilir).



Şekil: ER Diyagramı