

Bu projede aynı işlevi gerçekleştiren konsol uygulamasını C#, Java, Python ve JavaScript dillerinde geliştirdim. Diller arasındaki temel farklılıklar özellikle söz dizimi (syntax), veri tipi kullanımı ve nesne yönelimli yapının uygulanma biçiminde ortaya çıktı. Örneğin C# ve Java'da sınıf yapıları ve tip güvenliği daha katıken, Python ve JavaScript'te daha esnek bir yapı bulunmaktadır. Proje tasarımı aşamasında her dilde aynı mantığı koruyarak, kullanıcıdan veri alan ve işleyen bir sistem oluşturmaya odaklandım. KuantumNesnesi, anasınıf ve diğer sınıflar (AntiMadde, KaranlıkMadde ve VeriPaketi) alt sınıflar oldu. IKritik ve KuantumException ise yardımcı sınıflar oldu. En çok zorlandığım kısım ise farklı dillerde aynı algoritmayı hem doğru hem de dilin yapısına uygun şekilde uyarlamak ve hatasız çalışmasını sağlamaktı. Bu proje sayesinde programlama dilleri arasındaki farkları daha net görme ve çoklu dilde uygulama geliştirme konusunda önemli bir deneyim kazandım.

Dillerdeki Farklılıklar:

C#

Kalıtım : işareti ile gösterilir. `public class VeriPaketi : KuantumNesnesi { }`

Üst sınıf özelliklerini işaret etmek için `base()` kullanılır.

Getter setter (property) kavramı bu dilde vardır. Pratiklik sağlar.

`public interface IKritik` (arayüz kavramı vardır ve bu şekilde kullanılır)

Kapsülleme işleminde `this` belirteci özellik ile parametre aynı olursa kullanılmak zorundadır.

Java

Kalıtım `extends` ile belirtilir.

Üst sınıf özelliklerini işaret etmek için `super()` kullanılır.

C#, Python gibi dillerdeki formatlama yani yazdırma işleminde `string` ve `int` değerleri birarada gösterme bu dilde mevcut değildir.

Kapsülleme işleminde `this` belirteci her zaman kullanılmak zorundadır.

`public interface IKritik` (arayüz kavramı vardır ve bu şekilde kullanılır)

`override` şöyle kullanılır:

`@Override`

`public void analizEt() throws KuantumCokusuException { }`

Python

Sınıf ve kalıtım böyle gösterilir:

```
class KuantumCokusuException(Exception):
```

`__init__(self)` metodu c# ve javadaki constructor olarak kullanılır.

Property bu şekilde kullanılır:

```
@property
```

```
def stabilite(self):
```

Javascript

`extends` ile kalıtım yapılır.

`constructor(id) { super(); }` ile constructor kullanılır.

`this` burada da kullanılır.

Fonksiyon işlemlerinde erişim belirteçleri kullanılmaz. `function x()`

Bu dilde daha okunaklı ve ürün odaklı olması için html içine gömülü olarak projeyi yazdım ve tarayıcıda kolay bir şekilde bunu görebiliriz.