KAIZEN CASE STUDY SORU 2 FATURDAN TEXT'E ÇIKTI

Merhaba,

Clean code standartlarına uyulmuş bir kodun açıklamasının yapılmasına gerek yoktur. Fakat hem değerlendirme sürecinizi hızlandırmak hem de kurduğum mantığı açıklamak için bu dökümanı hazırladım.

İlk olarak bana gönderilen json üzerinde bir düzenleme yapmam gerekti, şöyle ki fiş için SaaS hizmetinden alınan response bilgisinin ilk elamanı zaten bütün faturaya ait text'i satır satır (\n ile ayrılmış şekilde) barındırmaktaydı.



Bu sebeple ilk elemanı kod üzerinden pas geçmek yerine, json üzerinde sadece ilk elamanı silerek json üzerinde sadece her elamanın desciription ve koordinat bilgisi olacak şekilde bıraktım.

 Json'u okuma mantığımı şu şekilde kurdum. Bana gönderilen her kelimenin 4 köşesinin de koordinat bilgisine sahibim. O halde her birinin sol üst köşesindeki koordinatı baz alarak bir satır sıralaması yaparsam kelimeleri aynı satır içinde sıralı olmasa bile ilk satırdan son satıra kadar olan kelimeleri satırlarına göre sıralamış olurum.

Bu sebeple kodun 16. satırında sol üst köşenin dikey eksendeki konumunu veren koordinat bilgisine (Verticle[0].Y) göre tüm json'ı sıraladım. Böylece artık json üzerinden kelimeler karışık düzende gelse bile ben sol üst köşe yüksekliğine göre doğru bir sıralama yapabiliyorum.

 Artık kelimelerim dikey eksende sıralandı. Bir sonraki aşamada aynı satırdaki kelimeleri yatay eksende sıralamam gerekiyor. Fakat bu aşamaya geçmeden önce dikey eksende sıraladığım kelimelerin satır bilgisini tutacak bir class oluşturup, bu dataları bir liste içinde tutuyorum.

```
int row = 0;
int coordinateInfo = 0;
InvoiceItems invoiceListForRowAssign = new();
InvoiceItems finalInvoice = new();

foreach (var invoice in sortedInvoiceByRow)
{
    InvoiceDetailInfo invoiceDetailInfo = new();

    if (invoice.BoundingPoly.Vertices[0].Y > (coordinateInfo + 15))
        row++;

invoiceDetailInfo.Description = invoice.Description;
invoiceDetailInfo.Row = row;
invoiceDetailInfo.BoundingPoly = invoice.BoundingPoly;
coordinateInfo = invoice.BoundingPoly Vertices[0].Y;
invoiceListForRowAssign.Items.Add(invoiceDetailInfo);
}
```

invoiceListForRowAssign ismindeki listemde kelimelerin description ve koordinat bilgisinin yanında bir de satır bilgisini tutmaktayım. Burada bir kelimenin bir sonraki satırda olduğunun kararını, kelimenin dikey eksende sol üst koordinat bilgisi bir önceki kelimeden 15 birim fazla olmasıyla veriyorum. Kelimelerin dikey eksendeki koordinatlarını incelediğimde 15 birim ve üzeri fazlalığın bir sonraki satırdaki kelimeye ait koordinat olacağı sonucuna vardım.

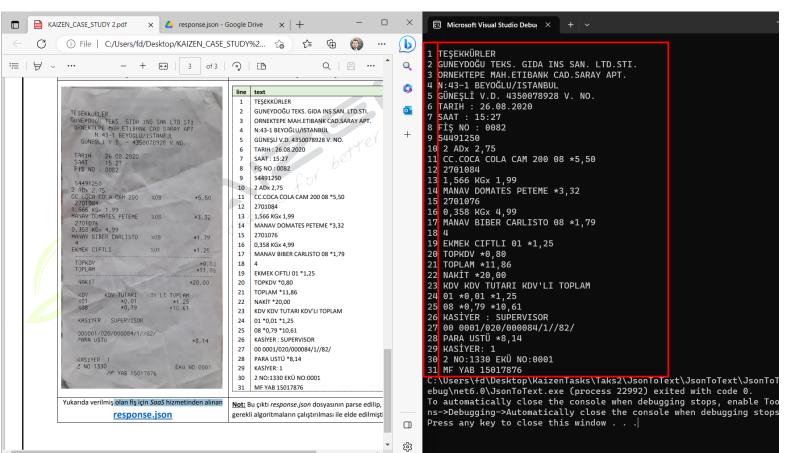
 Artık kelimelerin koordinat bilgisinin yanı sıra hangi satırda olduğuna ait bilgiyi de barındırdığıma göre son aşama olan aynı satırdaki kelimelerin yatay eksendeki sıralamasına geçebilirim.

```
for (int i = 0; i < invoiceListForRowAssign.Items.Max(i => i.Row) + 1; i++)
{
    List<InvoiceDetailInfo> invoiceItemsForLineOrder = invoiceListForRowAssign.Items.Where(x => x.Row == i).OrderBy(x => x.BoundingPoly.Vertices[0].X).ToList();
    foreach (var item in invoiceItemsForLineOrder)
    {
        finalInvoice.Items.Add(item);
    }
}
```

Satır sayısı kadar dönen bir for döngüsü içinde her döngüde o satırda bulunan kelimeleri Where ile çektiktek sonra kelimenin sol üst köşesinin yatak eksende konumunu veren koordinat bilgisine göre (Verticle[0].X) sıralıyorum. Böylece artık kelimelerim hem dikey hemde yatay eksende sıralanmış olarak finallnvoice listeme ekliyorum.

Son olarak finalInvoice listemdeki itemleri yazdırıyorum.

Sonuç



İyi çalışmalar dilerim.