**ENGİNCAN ÜNLÜ-150301009**

**Github adresi**: <https://github.com/engoloji>

**TASARIM ŞABLONLARI PROJE ÖDEVİ SENARYOSU**

Bizim ailede neredeyse herkes esnaf. Babam da uzun yıllardan beri bakkalcılık yapar. Ancak biz sadece parakende olarak ürün satmaktayız. Bizim bakkalın yıllardır çok sevilmesinin en büyük sebeplerinden biri kaliteli bir şarküterimizin olmasıdır.   
Babam yıllardır şarküteri ürünlerini en iyi satan toptancıları bulur ve ürünleri hep en kalitelilerini alır. Diğer bölgedeki diğer bakkal esnafı, babamın bu şarküteri ürünlerini kendilerine de göndermesini ister.

Bu noktada da babam, ürün vereceği yerleri kontrol etmek ve fiyat rekabetinden haberdar olmak için benden bir otomasyon istedi.Aklıma hemen Zeynep hocadan aldığım Tasarım şablonları dersinde öğrendiğim patternleri kullanarak bir otomasyon yapmak geldi.

Aslında otomasyonun temel amacı şehirdeki ürün aldığımız toptancıları tutmak(hatta bu ileride ülke bazlı olabilir o yüzden toptancı tarafı bizim için **genişletilmeye uygun** olmalı ).

Projenin toptancı kısmı bu yüzden bizim için **expansion point**.

Daha sonra bizden ürün alacak bakkalları kayıt altında tutmak ve bu bakkalların bizden aldığı ürünleri ne kadar karla satacaklarını göstermek.

Bakkal olarak bizden ürün alacak kişilerin de kayıtları titizlikle tutulması gerekiyor ve aynı zamanda da bizim ürün temin ettiğimiz toptancıları titizlikle tutmamız gerekiyor.

Aynı kodu tekrar yazmamak her seferinde değiştirilmeye kapalı olmalı.Burada bizi büyük bir dertten kurtarması için Kisi sınıfını oluşturduk. Burada kullandığımız değerler Kisi sınıfını miras alacak olan hem Bakkal hem de Toptancı için aynı olacağından dolayı Kisi sınıfımızı abstract yaptık(**Open Closed Principle**)

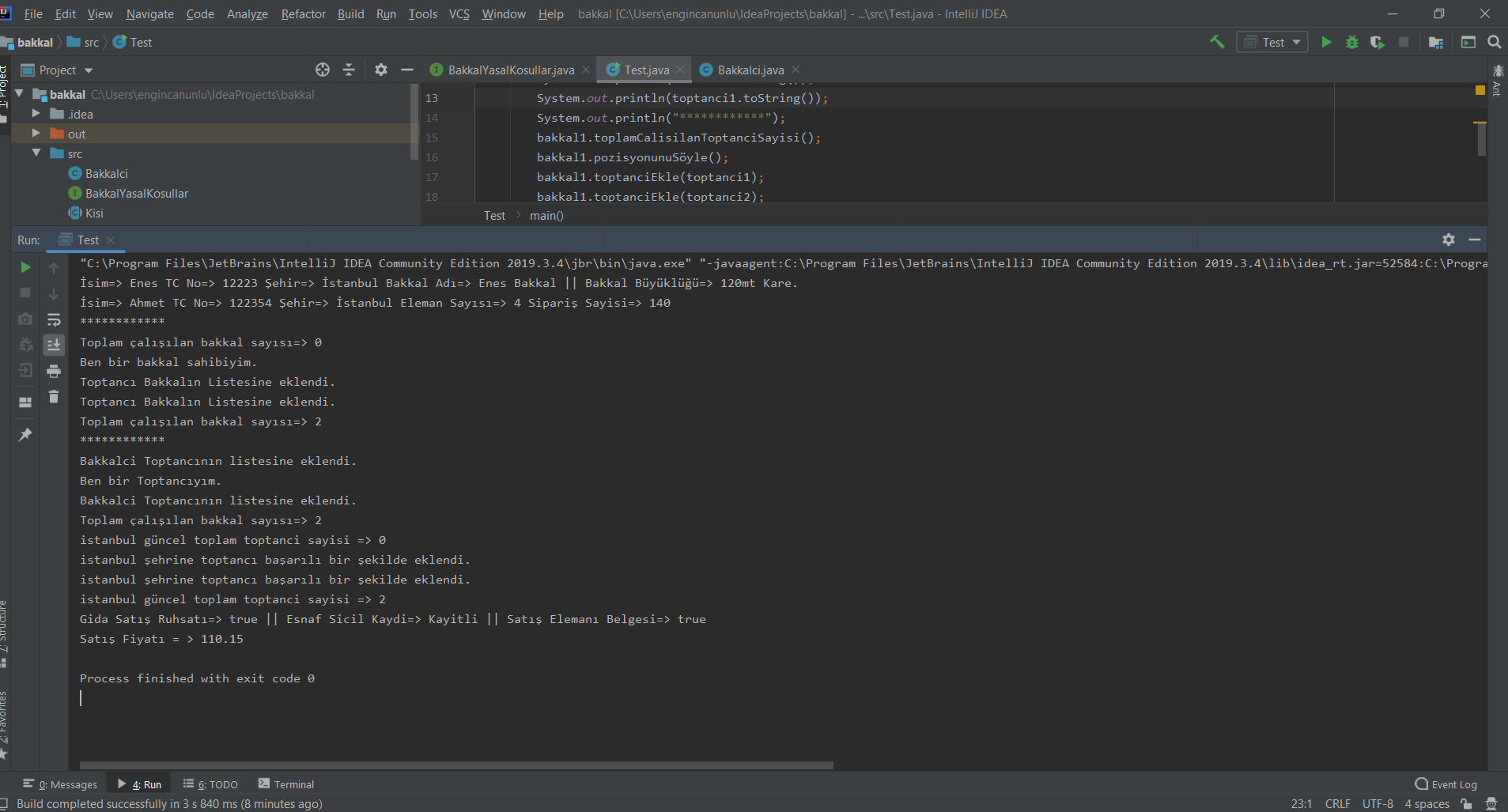
Ancak önemli bir nokta var ki bazı bilgiler kullanılması zorunlu olduğu gibi bazı özelliklerin böyle bir zorunluluğu olmamalıydı. Toptancı ya da bakkalda özellikle var ise kullanılabilir yoksa kullanılması gerekmeyen şeyler olabilirdi bu yüzden bazı özellikler zorunda tutulmamalıydı (**Interface Segregation Principle**) e göre tasarlandı.

Ayrıca BakkalYasalKosulları dediğimiz sınıfta olduğu gibi başka bir şeye bağımlı olmayan tek bir sorumluluğu üstlenen bir sınıf yapısı kullandık ki bu da **(Single Responsibility Principle**) yasasına uyuyor.

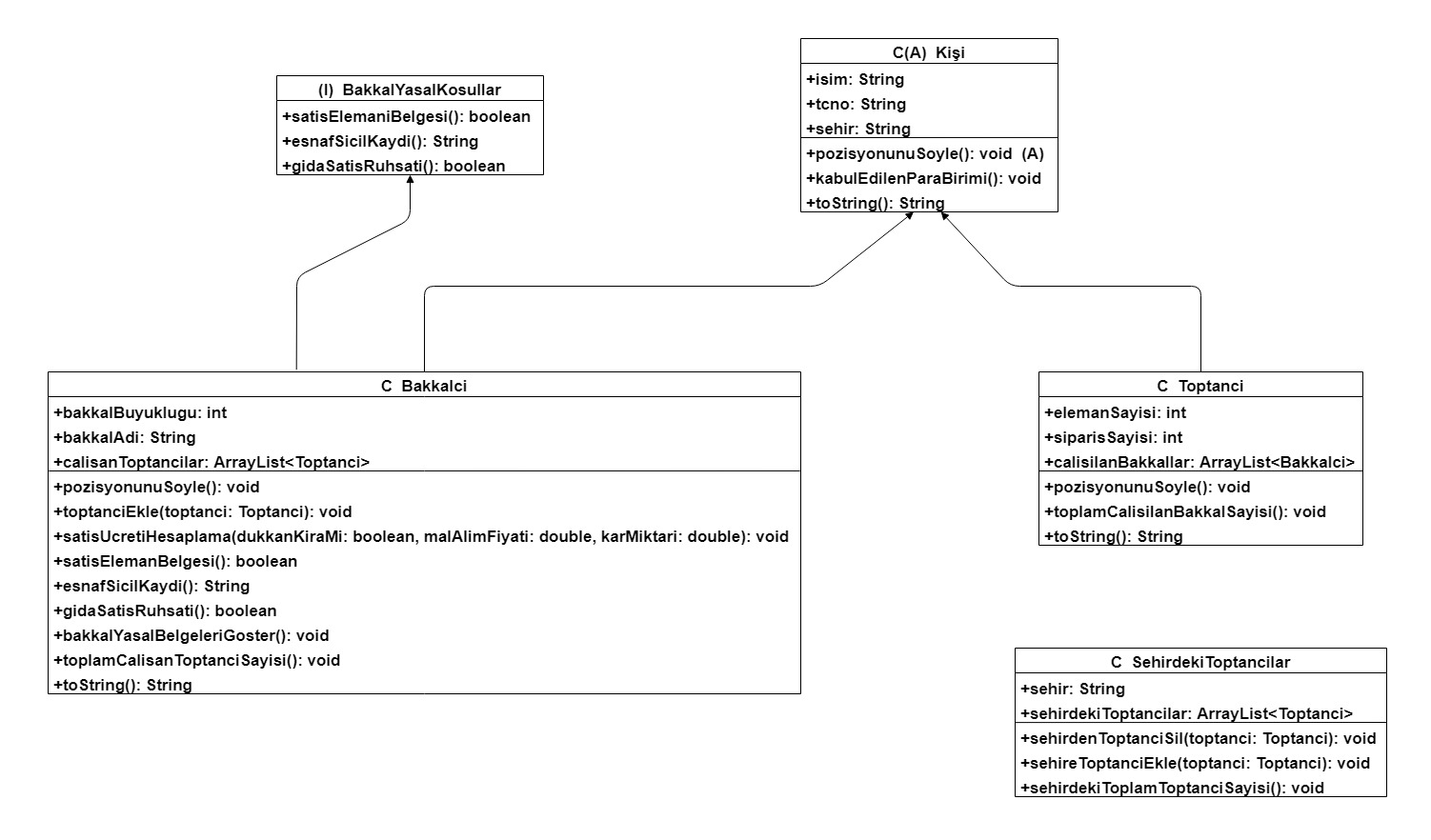
Bundan sonra projeyi benim dışımda geliştirecek biri olursa, otomasyonun kodunu rahatça okuyabilecek , değişiklik yapabilecek , geliştirmeleri hızlı bir şekilde otomasyona adapte edebilecek.

Proje kodları: <https://github.com/engoloji/designpattern>

Projenin başarı ile çalıştığı ekran çıktısı:



**UML DİYAGRAMI**



Kullanılan UML tool: draw.io