# Otopark Simülasyon Uygulaması

## Mimari

“Otopark Simülasyon Uygulaması” bir otopark içerisindeki araçların yönetimini sağlayan bir uygulamadır. Katmanlı mimari kullanılan bu uygulama hazırda çalışabilir durumdadır. Developerdan beklenen hazır yapıdaki bileşenleri değiştirmemesi, bu yapıların üzerine ekleyerek geliştirmelerini yapmasıdır.

Uygulamadaki hazır bileşenler hakkında detaylı bilgiler aşağıdaki gibidir:

1. **Client Katmanı – ParkingLotSimulator.Client**

Windows Console uygulamasıdır.

* *Program.cs:* Konsol uygulaması çalıştırıldığında programın başlangıç noktasıdır. Bu sınıftaki program akışının developer tarafından doldurulması beklenmektedir.

Mevcutta VehicleRepository sınıfının kullanımına örnek oluşturması için tek satırlık bir kod vardır.

* *App.config:* Uygulama config dosyasıdır. İçerisinde uygulamanın başlatıldığında okunan araç listesinin tutulduğu json dosyasının path bilgisi ve otopark kapasitesini belirten “ParkingLotCapacity” parametreleri vardır.

*Not: ParkingLotCapacity yazılımcı tarafından test amaçlı güncellenebilir.*

1. **Business Katmanı – ParkingLotSimulator.Business**

Otopark simülasyon uygulamasının business katmanıdır.

* *VehicleRepository.cs:* **IRepository** kontratından türeyen, mevcut araç bilgilerini barındırdığı bir singleton instance aracılığı ile yöneten sınıftır. Sınıf üzerindeki mevcut kodlar kesinlikle değiştirilmemelidir; yalnızca araç ekleme ve çıkarma işlemleri için sınıf içerisindeki “*Developer Implementation*” isimli region içindeki metotlar implement edilmelidir.

Sınıf singleton olduğu için ancak “*Current*” isimli property üzerinden kullanılabilir. Program.cs içerisinde bu sınıfın bir metodunun kullanım örneği bulunmaktadır, bu örnek uygulamanın son halinde silinmelidir.

* *Vehicle.cs:* Otoparkiçerisindeki araçları temsil eden sınıftır. Plaka ve aracın otoparka giriş tarihini barındırır. Sınıf üzerinde değişiklik yapılmamalıdır.
* *VehicleDataParser:* Uygulamada datasource olarak yer alır. Uygulama ilk çalıştığında solution içindeki bir json dosyasından araç listesini alır ve sonrasında kullanılmaz. Sınıf üzerinde değişiklik yapılmamalıdır.

## Talimatlar

1. Uygulama mimarisini dikkatlice inceleyin.
2. İlk olarak uygulamadaki bileşenleri (komponentleri) incelemek için zaman ayırın; uygulamada hali hazırda var olan Repository ve Parser yapılarını kullanmanız beklenmektedir.
3. **Kod kalitesine önem verin! Temiz kod sınav için önemli bir değerlendirme kriteridir! Kodunuz mümkün olduğu kadar okunabilir, modüler ve parametrik olmalıdır. Yorum satırları ile kodun okunabilirliğini arttırmayı unutmayın.**
4. **Uygulamanız istisnaları ele almalıdır (exception handling); var olmayan plaka, geçersiz input, dosya I/O vs.**

## Gereksinimler

Bu uygulama bir otopark simülasyonu yaratarak otoparka ait kapasite takibi ve araçlar için ücret hesaplama işlemleri yapması beklenmektedir.

Sistem, çalıştırıldığında kullanıcıya yapmak istediği işlemi sormalıdır ve bir input okuyarak yapılacak işlemin akışını başlatmalıdır. Tüm akışların sonunda kullanıcının yeni bir işlem yapabilmesine olanak verilerek uygulamanın devamlı olarak kullanılabilmesi sağlanmalıdır. Bu gereksinimden aşağıdaki tüm akışlarda da bahsedilmiştir.

1. Kullanıcının ilk yapabileceği işlem otoparkın o anki doluluğunu ve otopark içerisindeki araçların bir listesini ekranda görüntülemektir.
   1. Doluluk gösteriminde otoparkın kapasitesi ve mevcutta otoparkın içerisindeki araç sayısı görüntülenmelidir.
   2. Doluluk gösteriminin altında ise araçların plakalarının ve giriş saatlerinin listelenmesi gerekmektedir.
   3. Sistem “*Devam etmek için bir tuşa basın*” ifadesi ile kullanıcının yeni bir işlem yapmasına izin vermelidir.
2. Kullanıcının bir diğer yapabileceği işlem otopark içerisindeki bir aracın o anki ücret bilgisini hesaplamak ve otoparktan çıkışını yapmaktır.
   1. Kullanıcıdan ücret kesimi yapılacak aracın plaka bilgisi alınmalıdır.
   2. Ücret hesaplama işlemi sırasında uygulanacak algoritma aşağıdaki gibi olmalıdır:
      1. 0 ile 3 saat arasında otoparkı kullanan araçlar için saatlik **3 TL** ücret uygulanmalıdır.
      2. 3 ile 6 saat arasında otoparkı kullanan araçlar için üçüncü saatten itibaren saatlik **2 TL** ücret uygulanmalıdır.
      3. 6 saatten uzun süre otoparkı kullanan araçlar için altıncı saatten itibaren her üç saat için **5 TL** ücret uygulanmalıdır.

*Örnek ücretlendirme*

|  |  |  |  |
| --- | --- | --- | --- |
| **Otopark giriş saati** | **Ücret Hesaplama Saati** | **Arada Geçen Süre** | **Hesaplanan Ücret** |
| **07:15** | 07:55 | 40 dk | 3 TL |
| **07:15** | 09:45 | 2 saat 30 dk | 9 TL |
| **07:15** | 11:20 | 4 saat 5 dk | 13 TL |
| **07:15** | 17:00 | 9 saat 45 dk | 25 TL |
| **07:15** | 23:10 | 15 saat 55 dk | 35 TL |

* 1. Hesaplama işlemi yapıldıktan ve kullanıcıya görüntülendikten sonra aracın çıkış yapıp yapmayacağına dair kullanıcıdan Yes/No sorusu sorularak bilgi alınır. **Eğer kullanıcı aracın çıkış yaptığını belirtir ise araç otoparkın listesinden çıkartılması** **ve araç listesinde yeni eklenen aracın gelmesi beklenmektedir.**
  2. Sistem “*Devam etmek için bir tuşa basın*” ifadesi ile kullanıcının yeni bir işlem yapmasına izin vermelidir.

1. Son işlem olarak sistem kullanıcıya otoparka bir araç girişi yapma izni vermelidir. Kullanıcı araç girişi yapmak istediğini bir input ile belirtmeli ve sistem bunun ardından aşağıdaki akışı gerçekleştirerek otoparka bir araç girişi yapılmasını sağlamalıdır.
   1. Eğer otopark kapasitesi dolu ise, yani mevcut araç sayısı kapasites sayısına eşit veya fazla ise akış başlatılmadan son kullanıcıya kapasitenin dolu olduğuna ve araç girişi yapılamayacağına dair mesaj gösterilmelidir.

*Not: Kapasite bilgisi app.config dosyasından okunmalıdır, sabit bir değer olmamalıdır.*

* 1. Kullanıcıdan araca ait plaka bilgisi alınır. Plaka bilgisi tam 7 karakter uzunluğunda olmalıdır.
  2. Sistem aracı mevcut araç listesine ekler. **Araç ekleme işlemi sonrasında araç listeleme işleminde doluluk oranının güncellenmesi ve araç listesinde yeni eklenen aracın gelmesi beklenmektedir.**
  3. Sistem “*Devam etmek için bir tuşa basın*” ifadesi ile kullanıcının yeni bir işlem yapmasına izin vermelidir.