**T.C.**

**İSTANBUL AREL ÜNİVERSİTESİ**

**MÜHENDİSLİK MİMARLIK FAKÜLTESİ**

logo içeren bir resim

Açıklama otomatik olarak oluşturuldu**BİLGİSAYAR MÜHENDİSLİĞİ**

**ARABA KİRALAMA SİSTEMİ**

**NAZLI DOLMA**

**200309017**

**İSTANBUL, 2023**

**İÇİNDEKİLER**

**Sayfa**

**KAPAK………………………………………………………………………....…...**1

**İÇİNDEKİLER……………….………………………………………………….....**2

1. **GİRİŞ…………………………………………..………………………………..**3
2. **KULLANILAN SINIFLAR VE NESNE TABANLI PROGRAMLAMA (OOP) PRENSİPLERİ ………………………………………………………………....**4
   1. Araba sınıfı ….………………………………………………………4
   2. Elektrikli Araba ve Benzinli Araba Sınıfları…………………...........4
   3. Kiralama Sınıfı………………………………………………………4
   4. Arac Kiralama Uygulamasi Sınıfı…………………………………...5

Tkinter Kütüphanesi ve GUI Tasarımı

Veritabanı Bağlantısı ve Tabloların Oluşturulması

Araba Kiralama Süreci

Müşteri Girişi ve Araba Seçimi

Kiralama Tipi ve Süre Belirleme

Araç Durumunu Güncelleme ve Kiralama Bilgilerini Ekleyerek Süreci Tamamlama

Veritabanı Örnek Kayıtlar

Müşteri, Araba ve Kiralama Bilgilerinin Eklenmesi

Elektrikli ve Benzinli Arabaların Özellikleri

1. **SONUÇ………………………………………………………………...8**
2. **GİRİŞ**

Teknolojinin hızla gelişmesi ve değişen tüketici talepleri ile birlikte günümüzde araç kiralama sektörü önemli bir konuma sahiptir. Bu sektörde verimli ve etkili bir yönetim sağlamak, müşteri memnuniyetini artırmak ve işletme sahiplerine daha fazla kontrol sağlamak açısından çok önemlidir.

Bu uygulama, Python programlama dili kullanılarak geliştirilen ve grafik kullanıcı arayüzü (GUI) tasarımını bir araya getirerek, araç kiralama süreçlerini yönetebilen bir yazılımın nasıl oluşturulacağını gösterir. Kodun temelinde, nesne tabanlı programlama (OOP) prensipleri ve SQLite veritabanı kullanımı bulunmaktadır.

Bu makale, yazılımın temelini oluşturan sınıf yapısına odaklanarak başlamaktadır. Ardından, kullanıcı arayüzünü yöneten AracKiralamaUygulamasi sınıfının detaylı incelenmesiyle devam eder. Bu class; müşteri girişi, araç listesi görüntüleme ve kiralama işlemleri gibi önemli fonksiyonları içerir.

Makalenin ilerleyen bölümlerinde, uygulamanın genel yapısı, kullanıcı arayüzü, sınıf özellikleri, araç kiralama işlemleri ve veritabanı işlemleri gibi konular detaylı bir şekilde incelenecektir.

1. **KULLANILAN SINIFLAR VE NESNE TABANLI PROGRAMLAMA (OOP) PRENSİPLERİ**

**2.1 Araba Sınıfı:**

* **Nesne Tabanlı Programlama (OOP)** **Prensipleri:**
* **Encapsulation (Kapsülleme):** Araba sınıfı, özelliklerini (marka, model, yıl, kilometre) ve bu özelliklere erişim yöntemlerini içerir. Bu sayede, sınıfın iç yapısını gizleyerek dış dünyadan gelebilecek olası değişikliklere karşı koruma sağlanır.
* **Inheritance (Kalıtım):** Elektrikli Araba ve Benzinli Araba sınıfları, Araba sınıfından türetilmiştir. Bu sayede, temel özellikleri Araba sınıfından miras alarak, kod tekrarı önlenir ve kodun genişletilebilirliği artar.
* **Abstraction (Soyutlama):** Araba sınıfı, soyut bir sınıftır. İçerisinde soyut bir metot olan "bakim" metodu bulunur. Araç tipine özel bakım sürecini soyut olarak tanımlayarak yöntemin alt sınıflara göre uygulanması gerektiğini belirtir.

**2.2 Elektrikli Araba ve Benzinli Araba Sınıfları:**

* **Nesne Tabanlı Programlama (OOP) Prensipleri:**
* **Inheritance (Kalıtım):** Her iki sınıf da Araba sınıfından türetilmiştir. Bu sayede, her iki sınıf da Araba sınıfının özelliklerini ve metotlarını kullanabilir.
* **Polymorphism (Çok Biçimlilik):** Her iki alt sınıf, Araba sınıfının soyut metodu olan "bakim"ı kendi ihtiyaçlarına göre uygular. Aynı isimli metodun farklı davranışlar sergilemesine olanak tanır.

**2.3 Kiralama Sınıfı:**

* Kiralama sınıfı, bir kiralama işlemini temsil eder. Bu sınıf, kiralama işleminde kullanılan özellikleri ve bu işlemin ücretlendirilmesini içerir.
* Constructor (\_\_init\_\_): Bu metod, kiralama işlemi için gerekli olan bilgileri alır. Müşteri adı (musteri), araba bilgisi (araba), kiralama tipi (kiralama\_tipi) ve süre (sure) parametre olarak alınır ve kiralama nesnesi oluşturulur.
* Ücret Hesaplama Metodu (hesapla\_ucret): Bu metot, kiralama süresi ve tipine göre kira ücretini hesaplar. Eğer kiralama tipi "günlük" ise süreyi 1000 ile, "saatlik" ise süreyi 500 ile çarparak ücreti belirler. Eğer geçersiz bir kiralama tipi girilirse ekrana bir uyarı yazdırır ve -1 değerini döndürür.

**2.4 AracKiralamaUygulamasi Sınıfı:**

* **Nesne Tabanlı Programlama (OOP) Prensipleri:**
* **Encapsulation (Kapsülleme):** AracKiralamaUygulamasi sınıfı, GUI (grafik kullanıcı arayüzü) tasarımını ve kullanıcı etkileşimini içerir. Bu, uygulamanın arayüzle olan etkileşimini sınırlar ve iç yapısını korur.
* **Abstraction (Soyutlama):** AracKiralamaUygulamasi sınıfı, kullanıcıya görünen arayüz ile arka planda gerçekleşen işlemleri soyutlayarak, uygulamanın kullanıcıya daha sade bir deneyim sunmasını sağlar.

1. Sistem Yapısı ve Sınıf **Yapısı**

Araç Kiralama Uygulaması, genel olarak birinci sınıf nesne modeline dayalı bir sınıf yapısı üzerine inşa edilmiştir. Bu sınıflar arasındaki ilişkiler, Python programlama dilindeki nesne tabanlı programlama prensiplerine uygun bir şekilde tasarlanmıştır. Başlıca sınıflar arasında Araba, ElektrikliAraba, BenzinliAraba, Kiralama, ve AracKiralamaUygulamasi yer almaktadır.

Araba Sınıfı: Temel araç özelliklerini içerir ve soyut bir sınıf olan Araba sınıfından türetilir. Her aracın marka, model, üretim yılı gibi temel bilgileri bu sınıfta bulunmaktadır. Aynı zamanda aracın kilometre bilgisi ve özel bakım işlemleri de bu sınıf içerisinde yer alır.

ElektrikliAraba ve BenzinliAraba Sınıfları: İki farklı araç tipini temsil eden bu sınıflar, Araba sınıfından türetilir. Elektrikli araçlar için batarya durumu, benzinli araçlar için ise yakıt durumu gibi özellikleri içerirler. Her iki sınıf da soyut bir metot olan bakim metodu ile araç bakım işlemlerini gerçekleştirir.

Kiralama Sınıfı: Kiralama işlemlerini temsil eden bu sınıf, müşteri, araç, kiralama tipi, kiralama süresi gibi bilgileri içerir. Ayrıca kiralama sürecinin başlangıç tarihini kaydeder ve kiralama ücretini hesaplamak için hesapla\_ucret metodu kullanılır.

AracKiralamaUygulamasi Sınıfı: Bu sınıf, GUI tasarımını ve uygulamanın temel akışını yönetir. Müşteri girişi, araç listesi gösterimi, araç kiralaması ve veritabanı işlemleri gibi işlevselliği içerir. Aynı zamanda kullanıcı arayüzü elemanlarını oluşturarak, uygulama ile etkileşimi sağlar.

Bu sınıfların bir araya gelmesiyle, uygulama kullanıcıların araç kiralama süreçlerini düzenli ve anlaşılır bir şekilde yönetebilmekte, aynı zamanda veritabanı üzerinde güncel bilgileri saklayarak takip edebilmektedir.

1. **Veritabanı Tasarımı ve İlk Verilerin Eklenmesi:**
   * SQLite veritabanının nasıl kullanıldığının detaylı açıklanması.
   * Musteri, Arabalar ve Kiralama tablolarının tasarımı ve mantığı.
   * İlk verilerin veritabanına nasıl eklendiği.
2. **AracKiralamaUygulamasi Sınıfı:**
   * Grafik kullanıcı arayüzü (GUI) tasarımının nasıl gerçekleştirildiği.
   * Müşteri girişi, araç listesi gösterimi ve kiralama işlemleri üzerindeki işlevselliğin detaylı açıklanması.
   * Kullanıcı deneyimi ve arayüz tasarımındaki düşünce süreci.
3. **Araç Kiralama İşlemi:**
   * Uygulamanın araç kiralama süreçlerinin adım adım açıklanması.
   * Kiralama ücretinin nasıl hesaplandığının vurgulanması.
   * Veritabanı güncellemelerinin nasıl yapıldığının açıklanması.
4. Veritabanı İşlemleri:

SQLite veritabanı bağlantısının nasıl kurulduğu ve nasıl yönetildiği.

Veritabanı tablolarının oluşturulması ve güncellenmesi.

İlk verilerin veritabanına eklenmesinin etkili bir şekilde nasıl yapıldığı.

1. Sonuç ve Gelecek Adımlar:

Projeye ait başarıları öne çıkarmak.

Geliştirilebilecek noktaların belirtilmesi ve gelecek adımların açıklanması.

Kullanıcı Arayüzü ve Arayüz Kontrolleri

AracKiralamaUygulamasi sınıfı, tkinter kütüphanesi kullanılarak grafik bir kullanıcı arayüzü tasarlar. Bu arayüz, müşteri girişi, araç listesi gösterimi ve araç kiralama işlemleri için kontroller içerir. Kullanıcı, uygulama aracılığıyla araç kiralamak için basit bir kullanıcı deneyimine sahiptir.