İsim: Altay Yücetaş

Üniversite: Orta Doğu Teknik Üniversitesi

Bölüm: Bilgisayar Mühendisliği

İÇİNDEKİLER

TÜBİTAK BİLGEM
YAZILIM PROJESİ

Projenin Ana Teması

Bu projede bir kahve sipariş uygulaması geliştirilmesi istenmiştir. Örnek çıktıda belirtilen 7 adet kahvenin konsola çıktı olarak verilmesi ve kullanıcının bu çıktılardan bir tanesini seçmesi gerekmektedir. Kullanıcı istediği seçeneği seçtikten sonra kahve hazırlanıp içeriği ile birlikte konsola yazdırılmaktadır.

Ben bu verilen bilgilerden yararlanarak projeyi 3 ana sınıfa böldüm. Bunlardan ilki "Coffee" sınıfıdır.

"Coffee" sınıfı, 3 adet değişkenden oluşmaktadır. Bunlardan ilki kahve objesinin adını tutmak için tanımlanmış ve "String" türünde olan "name" değişkenidir. İkincisi, kahve objesinin fiyatını tutmak için tanımlanmış ve "int" türünde olan "price" değişkenidir. Sonuncusu ise kahve objesinin tarifini "ürün : miktar" şeklinde tutmak için tanımlanmış ve "HashMap<String, Integer>" türündeki "recipe" değişkenidir. İlki iki değişken obje oluşturulduktan hemen sonra "Default Constructor" tarafından gerekli değerlere eşitlenirler. "recipe" değişkeni ise birkaç yardımcı işlemden geçer. İlk olarak tarif "1xHot Water, 2xEspresso …" formatında alındığından virgüllere göre ayrılarak bir "String Array"ine eklenir. Bu sayede kahvenin içeriği, türlerine göre ayrılmış olur. Ardından bu array içindeki kahve içerikleri, içerik ve içeriğin miktarını birbirinden ayırmak için bir "foreach" döngüsüne girerler. Her bir döngüdeki "String"in ilk karakteri içeriğin miktarını belirtirken 2. indeksinden "String"in sonuna kadar olan kısım içeriğin türünü belirtir. Bu sayede, içerik ve içeriğin miktarı, key-value ikilisi şeklinde "HashMap"e eklenirler. Ayrıca şunu da belirtmek isterim ki kullanıcının siparişi hazırlandıktan sonra ekrana bastırılacak olan kullanıcının seçtiği kahve türü ve bu kahve türünün içeriği direkt olarak bu "HashMap"ten alınmaktadır. Bundan dolayı kahvenin içeriği İngilizce olarak ekrana bastırılır. Örnek vermek gerekirse, "Mocha seçtiniz. Bu içeceğimiz 1 doz Hot Water, 2 doz Espresso … içermektedir." gibi.

"Coffee" sınıfının devamında ise OOP'de gerekli olan "Abstraction" ilkesi gereğince getter fonksiyonları tanımlanmıştır. Son olarak, "toString" fonksiyonu

"Override" edilerek menüde gösterilmesi planlanan formata göre düzenlenmiştir. Örneğin, "Espresso (20 TL)" gibi.

İkinci sınıf ise "CoffeeMenu" sınıfıdır. Bu sınıf, kahve objelerinin üretiminden ve bunların depolanmasından sorumludur. Yani, aslında bildiğimiz kahve menüsü görevi görür. İki adet değişkene sahiptir. Bunlardan ilki üretilen kahve objelerinin depolanacağı "ArrayList<Coffee>" türündeki "menu" listesi, diğeri ise bu listedeki kahve adetlerinin tutulacağı "int" türündeki "menuSize" değişkenidir. Bu sınıfın "Default Constructor"ı, obje üretildikten sonra görevde belirtilen 7 adet kahve objesini üretir, bunları "menu" listesine ekler ve "menuSize"ı 7'ye eşitler. Obje üretildikten sonra ise objenin kullanabileceği birkaç fonksiyon vardır. Bunlardan bazıları, "Coffee" sınıfında olduğu gibi klasik olarak tanımlanan "getter" fonksiyonlardır.

Bu sınıfta, projede istenilen görevlerden farklı olarak menüye kahve ekleme, menüden kahve çıkarma veya menüyü tamamen silme işlevleri olan fonksiyonlar tanımladım. Asıl amacım, tanımladığım sınıflardaki olayları değiştirerek farklı senaryolar oluşturmak ve bu fonksiyonların güvenlik anlamında yaratabilecekleri olası sorunları incelemekti. Sınıfta 2 adet "addCoffee", 2 adet "deleteCoffee" ve 1 adet "clearMenu" fonksiyonu bulunmakta. "addCoffee" fonksiyonu (kahve_ismi, kahve_fiyatı, tarif) veya (kahve_nesnesi), "deleteCoffee" fonksiyonu ise (kahve_ismi) veya (kahve_nesnesi) parametrelerini alabilmek amacıyla overload edildi. Kod üzerindeki yorum satırlarında da belirtildiği üzere (kahve_nesnesi) alan fonksiyonların amacı sadece test sürecinde kolaylık sağlamalarıdır. Bu fonksiyonlar ayrıca "menuSize" değişkenini de güncellemektedir. "clearMenu" fonksiyonu ise sadece "menu" listesini temizleyip "menuSize"ı 0'a eşitlemektedir.

Son olarak menünün tamamının kullanıcıya gösterilmesi için "toString" fonksiyonu "Override" edilmiştir. Örnek çıktısı, "1. Espresso (20 TL) …" şeklinde aşağıya doğru uzayarak gitmektedir.

Son sınıf ise "CoffeeMachine"dir. Bu sınıf kullanıcı ile etkileşimde olan tek sınıftır. Kullanıcı girdisini alır ve girdiye göre gerekli çıktıları oluşturur. Bu sınıfın tek değişkeni "CoffeeMenu" türündeki "myMenu"dür. Bu sınıfın objesi üretildikten sonra ilk olarak kullanıcıyı bir hoş geldin mesajı ile selamlar ve menüyü kullanıcı için bastırır. Kullanıcı, istediği kahveyi seçtikten sonra kahvenin numarasını girer ve bu sınıf seçilen numaraya göre bir geri dönüş sağlar. Bu sınıftaki ilk fonksiyon var olan menüyü döndüren "printMenu"dür.

"CoffeeMenu" sınıfında da bahsettiğim gibi projede istenilen görevlerden farklı olarak birkaç yapı tanımlamıştım. Bunlardan bir tanesi de "CoffeeMachine" sınıfında yer alan "accessAPI" fonksiyonudur. Eğer kullanıcı hazırlanacak olan kahve numarası için 10.000'den fazla bir değer girer ise bu fonksiyon devreye girer. 10.000'den fazla

kahve sunan bir menünün olması pek mümkün olmadığı için bu sayıyı seçtim. Bu fonksiyon girilen ve 10.000'den büyük olan bu sayıyı alır ve kendi içinde bulunan "612546903" değeri ile karşılaştırır, eğer girilen değer ile "612546903" sayısı eşit ise kullanıcı, kahve makinesinin geliştirici menüsüne erişir. Girilen değer "612546903"ye eşit değil ise bir uyarı alır ve tekrardan kahve seçme ekranına döner. "612546903" sayısı, API'a giriş için gereken ve sadece geliştiricilere verilmesi gereken bir doğrulama mekanizması kodudur. Bu sayede geliştiriciler gerektiğinde gerekli değişiklikleri yapabilirler.

Geliştirici ekranına eriştikten sonra ekrana 3 adet seçenek gelmektedir. Bunlar "Kahve ekleme, kahve çıkarma ve çıkış" seçenekleridir. Geliştirici, bu menü aracılığı ile kahve ekleme ve çıkarma işlemi yapabilir, yanlışlıkla girdiği taktirde ise "çıkış" seçeneğini seçerek menüden çıkabilir. Geliştirici, kahve ekleme ve çıkarma işlemlerinden birini seçtiğinde ilk olarak kaç adet kahve ekleyeceğini/sileceğini belirtir. Ardından, eğer ekleme adımını seçmişse sırasıyla "kahve adı", "kahve fiyatı" ve "kahve tarifi" değerlerini girerek ekleme işlemini yapar. Birden fazla kahve eklemek istiyorsa "for" döngüsü aracılığı ile üstteki sıra ile diğer kahve bilgilerini de girmesi gerekir. Eğer silme adımını seçmişse sadece silmek istediği kahvenin adını girer ve bunu önceden girdiği silme işlem adedi kadar yapar. Eğer silmek istediği kahve menüde değilse kahve isminin bulunamadığına dair bir uyarı mesajı alır.

"CoffeeMachine" sınıfının sahip olduğu diğer bir fonksiyon ise "getCoffeeInfo" fonksiyonudur. Menü ekrana bastırıldıktan sonra kullanıcının girdiği değer eğer 10.000'den küçük ise otomatik olarak bu fonksiyon devreye girer. Kullanıcının girdiği değere göre kahve hazırlanır. Eğer girilen sayı değeri, 0'dan küçük ve menü boyutundan büyük ise kullanıcı yanlış değer girdiğine dair bir hata mesajı alır. Eğer girilen değer geçerli bir değer ise, kullanıcıya kahvenin hazırlandığı bilgisi verilir, kahvenin hazırlanması için 2 saniye beklenir. Ardından kullanıcının seçtiği kahve ve bu kahvenin içeriği hakkındaki bir bilgilendirme mesajı ekrana yazılır. Örnek olarak "Americano seçtiniz. Bu içeceğimiz 4 doz Hot Water ve 1 doz Espresso içermektedir. Afiyet olsun." gibi.

Belirtilen uygulamanın ana iskeleti bu kadardır. Geri kalan ve kullanıcından input alacak olan kısım "CoffeeApp" adı verilen ve "main" fonksiyonunu bulunduran sınıfta yer alır. Burada gerekli objeler tanımlanır ve kullanıcıdan gerekli bilgiler alınır. Alınan bilgiler, "CoffeeMachine" sınıfına gönderilerek işlenir ve bu işlemler sonrasında üretilen sonuçlar kullanıcıya konsol aracılığı ile iletilir.

Olası Riskler Ve Güvenlik Önlemleri

Bir bilgisayar yazılımı geliştirilirken mümkün olduğunca güvenlik açıkları kapatılmaya çalışılır. Fakat, gerek geliştirici tarafından gerekse yazılım geliştirilirken kullanılacak veya uyum sağlanacak teknolojiler tarafından kaynaklanan problemlerden dolayı güvenlik açığı oluşabilir/bırakılabilir. Bu açıklar genellikle büyük projelerde daha sık görünse de bahsedilen kahve makinesi projesi gibi küçük projelerde de görülebilir. Bu projedeki olası güvenlik açıklarından bazılarına bu raporda değinmek istiyorum.

1. **API Güvenliği:** "Projenin Ana Teması" başlığı altında da belirttiğim gibi yazılımda bilerek açık bırakmak ve bunu bu başlık incelemek için "accessAPI" adından bir fonksiyon oluşturdum. Bu fonksiyonun açıklarına gelecek olursak, ilk olarak bu fonksiyon onu kullanacak geliştirici ile iletişimi sırasında herhangi bir şifreleme (TLS gibi) kullanmıyor. Bu da yazılımı MITM (Man In The Middle) saldırılarına karşı savunmasız bırakıyor. Saldırgan, iletişimi dinleyip müdahale ederek yazılımın CIA (Confidentiality, Integrity ve Availability) üçlüsünü tehlikeye atabilir. Olası saldırıların önüne geçmek için yukarıda bahsedildiği üzere TLS gibi iletişimi şifreleyen algoritmalar kullanmak hem yazılımın evrensel standartlar içinde kalmasını sağlar hem de güvenliği daha üst seviyeye taşır. Diğer bir açığa gelecek olursak o da "Access Token"den kaynaklanıyor. Geliştirici, API menüsüne erişmek için "612546903" girmek zorunda fakat bu kod sadece 9 karakter uzunluğunda ve Brute-Force saldırısı ile kolayca kırılabilmekte. Saldırgan, bu saldırı ile eğer "Access Token"ı ele geçirirsek yukarıda bahsedildiği gibi CIA üçlüsünü tehlikeye atabilir. Bu menüye erişimi daha güvenli hale getirmek için daha büyük "Access Token" uzunlukları seçilmelidir. Örneğin, Google Oauth 2.0, 2048 byte uzunluğunda "Access Token"a sahiptir ve Brute-Force saldırısı ile kırılması şu an için pek mümkün değildir. "Access Token"ı üretmek için kullanılan algoritma ve uzunluklar API'ı saldırganlara karşı olan korumayı birkaç adım daha güçlendirir. Bu fonksiyon hakkında bahsedeceğim ve belki bir açık olarak sayılmasa da önemli olduğunu düşündüğüm son bir yorum yapmak istiyorum. Sadece "Access Token"a bağlı kalmak güvenliği ciddi derecede tehlikeye atar çünkü bu token eğer bir şekilde ele geçirilirse saldırganlar belki de farkedilmeden çok uzun süre bu açığı sömürebilirler. Bundan dolayı "Multi-Factor Authentication" adı verilen ve birden fazla doğrulama adımı içeren güvenlik sistemleri, saldırganların işini zorlaştırarak güvenliği arttırır. Ayrıca şüpheli girişlerin kayıt altına alınması saldırganları ve saldırı vektörlerini tespit etmeye yardımcı olur.

2. **Girdi Kontrolü:** Projedeki yazılım kullanıcılardan bir girdi almakta ve o girdiye göre gerekli işlemleri yapmaktadır. Fakat, kullanıcından alınan girdi her zaman sağlıklı olmayabilir. Örneğin, kahve objeleri bir veri tabanında tutuluyor ve "CoffeeMachine" sınıfı kullanıcıdan aldığı girdiyi direkt olarak bu veri tabanından gerekli kahve bilgisini çekmek için kullanıyorsa yazılım tehlike altındadır çünkü unutulmamalıdır ki saldırganlar da aslında kullanıcı sınıfındadırlar. Saldırganlar, "SQL Injection" denilen saldırı türü ile zararlı girdiler kullanarak veri tabanına yetkisiz erişim sağlayabilir ve veri tabanının bütünlüğü ve gizliliğini tehlikeye atabilirler. Bu veri tabanı, şirket veya üzerinde bulunduğu sistem için kritik verileri tutuyor olabilir. Bunların saldırganlar tarafından ele geçirilmesi ve değiştirilmesi şirkete çok büyük zararlar verebilir. Bundan dolayı kullanıcıdan girdi alırken ilk olarak bu girdinin zararlı olup olmadığı kontrol edilmeli ve ancak ondan sonra işleme alınmalıdır.

3. **Güncellik Takibi:** Her ne kadar bu uygulama böyle bir sorun ile pek karşı karşıya olmasa da veri tabanı veya dışarıdan uygulamaya eklenen bir bağlılık yazılımı tehlikeye atabilir. Geliştiriciler, işlerini kolaylaştırmak için yazılımın başka bir yazılım veya kütüphane ile entegrasyonunu sağlayabilir. Fakat, bu entegrasyonların güncelliği her zaman kontrol edilmeli ve gerektiği taktirde güncellenmelidir. Geliştirici her ne kadar kendi yazılımını güvenli bir şekilde geliştirirse geliştirsin eğer başka bağlılığa kendi yazılımında yer verdiyse aslında onun sahip olduğu güvenlik açıklarına da kendi yazılımında yer vermiştir. Örnek vermek gerekirse, Java tabanlı olan ve "Log4j" adı verilen bir bir günlük kitaplığı vardır. Geçtiğimiz yıllarda bunun hakkında bir zero-day açığı keşfedildi. Bu uygulamayı kullanan herkes de o zamanlar tehdit altındaydı çünkü uygulamayı kendi uygulamalarına entegre etmişlerdi. Başka bir örnek olarak, Java platformu için bir framework olan "Spring"te açık keşfedildi ve kullanan herkes saldırı tehditi altındaydı. Bundan dolayı bağlılık bulunan teknolojilerin güvenlik ve güncelleme kontrolü sürekli bir şekilde yapılmalı ve güncelleme geldiği taktirde güncelleme işlemi sağlanmalıdır.

4. **Verilerin Saklanması:** Bu projede üretilen kahve veya menü objeleri sınıf yapıları içinde tutulmaktadır. Bu durum bu proje için pek bir problem oluşturmamakta. Fakat önemli ve kritik bilgiler içeren bir uygulamanın verileri herhangi bir şifreleme kullanmadan ve yazılım kendi içinde tutması CIA üçlüsünü tehdit edebilir ve bazı bilgilerin çalışnmasını/değiştirilmesi tetikleyebilir. Bundan dolayı verileri şifreli şekilde bir veri tabanında tutmak uygulama için bir güvenlik problemini daha ortadan kaldıracaktır.

Son

Yukarıda bahsettiğim Spring Framework zafiyeti hakkında eskiden bir makale yazmıştım. O makaleme aşağıdaki linkten ulaşabilirsiniz.

https://github.com/altayucetas/my-articles/blob/main/Picus-Blog-For-Github.pdf

Bunun dışında hazırladığım diğer projelerime ise GitHub hesabımdan ulaşabilirsiniz.

https://github.com/altayucetas/

Bu projeyi de GitHub hesabıma yükledim fakat herhangi bir kopya durumu ile karşılaşmamak repository'i private olarak tanımladım. Proje tesliminin son gününden sonra public'e çevireceğim. Onun linkini de aşağıya bırakıyorum. Eğer linke 4 Eylül akşamından sonra dahi ulaşılamıyorsa link değişmiş olabilir. Tekrardan yukarıdaki GitHub hesabımdan erişebilirsiniz.

https://github.com/altayucetas/tubitak-project

Herhangi bir sorunuz olursa bana ulaşabilirsiniz. Raporu okuduğunuz ve zaman ayırdığınız için teşekkür ederim.