

Nesne Tabanlı Proje Analiz Raporu

Giriş

Bu rapor, sağlanan nesne tabanlı projenin (BTPuygulaması) nesne yapısını, kullanılan sınıfları, nesneleri ve bunların uygulama içindeki rollerini ve kullanım amaçlarını detaylı bir şekilde analiz etmek amacıyla hazırlanmıştır. Proje, bir bütçe takip uygulamasını temsil etmekte olup, C# dili ve Windows Forms (WinForms) teknolojisi kullanılarak geliştirilmiştir. Rapor, projenin temel bileşenlerini, aralarındaki ilişkileri ve nesne tabanlı programlama prensiplerinin uygulamadaki yansımalarını açıklamaktadır.

Proje Dosyalarının Analizi

Program.cs Analizi

Program.cs dosyası, uygulamanın ana giriş noktasıdır. Program adında static bir sınıf içerir ve bu sınıfın içinde Main metodu bulunur. Main metodu, Windows Forms uygulamasını başlatır ve Form1 sınıfının yeni bir örneğini çalıştırır.

Sınıflar: - Program : Uygulamanın başlangıç noktasını içeren static sınıf.

Nesneler: - Form1 : Uygulama başladığında oluşturulan ilk form nesnesi.

Kullanım Amacı: - Uygulamanın yürütülmesini başlatmak ve ana kullanıcı arayüzü formunu (Form1) yüklemek.

Form1.cs Analizi

Form1.cs dosyası, uygulamanın giriş/kayıt ekranını temsil eden bir Windows Form sınıfıdır. Kullanıcı adı ve şifre ile giriş yapma veya yeni bir kullanıcı kaydetme işlevselliğini sağlar. MySQL veritabanı ile etkileşim kurarak kullanıcı bilgilerini doğrular ve kaydeder.

Sınıflar: - Form1 : Kullanıcı giriş ve kayıt işlemlerini yöneten ana form sınıfı.

Nesneler: - MySqlConnection : Veritabanı bağlantısını yönetmek için kullanılan nesne.

- MySqlCommand : Veritabanı sorgularını yürütmek için kullanılan nesne. -

guna2TextBox1 , guna2TextBox2 : Kullanıcı adı ve şifre girişi için kullanılan GUI

bileşenleri. - guna2Button1 , guna2Button2 , button1 : Giriş, kayıt ve çıkış işlemleri

için kullanılan GUI düğmeleri. - `icerik` : Başarılı giriş sonrası gösterilen ana içerik formunun nesnesi.

Kullanım Amacı: - Kullanıcıların sisteme giriş yapmasını veya yeni hesap oluşturmalarını sağlamak. - Kullanıcı kimlik bilgilerini doğrulamak ve veritabanına kaydetmek. - Başarılı giriş durumunda `icerik` formuna geçişi sağlamak.

icerik.cs Analizi

`icerik.cs` dosyası, uygulamanın ana işlevselliğini içeren ve bütçe girişlerini yöneten bir Windows Form sınıfıdır. Kullanıcıların gelir ve giderlerini kaydetmelerine, kategorilere ayırmalarına, filtrelemelerine ve görselleştirmelerine olanak tanır. Ayrıca aylık özet ve öneriler sunar.

Sınıflar: - `icerik` : Bütçe yönetimi işlevselliğini sağlayan ana içerik formu sınıfı.

Nesneler: - `MySqlConnection` , `MySqlCommand` , `MySqlDataReader` : Veritabanı işlemleri için kullanılan nesneler. - `guna2cmbCategory` , `guna2cmbCategory2` , `cmbFilterCategory` , `cmbTypeFilter` , `cmbMonth` , `cmbYear` : Kategori, filtreleme ve tarih seçimi için kullanılan açılır liste bileşenleri. - `guna2txtAmount` , `guna2txtDescription` : Tutar ve açıklama girişi için kullanılan metin kutuları. - `guna2dtpDate` : Tarih seçimi için kullanılan tarih seçici bileşeni. - `guna2btnSave` , `guna2btnToggleIncomeExpense` , `btnApplyFilter` : Kaydetme, gelir/gider geçişi ve filtreleme için kullanılan düğmeler. - `chartSummary` : Bütçe verilerini pasta grafiği olarak görselleştirmek için kullanılan grafik bileşeni. - `MonthlyAdvisor` : Aylık öneriler oluşturmak için kullanılan yardımcı sınıfın nesnesi. - `Guna.UI2.WinForms.Guna2GroupBox` , `Guna.UI2.WinForms.Guna2TextBox` , `Guna.UI2.WinForms.Guna2Button` : Dinamik olarak oluşturulan bütçe girişlerini görüntülemek ve silmek için kullanılan GUI bileşenleri.

Kullanım Amacı: - Kullanıcıların gelir ve giderlerini kaydetmelerini sağlamak. - Bütçe girişlerini kategorilere ayırmak ve açıklama eklemek. - Bütçe girişlerini tarihe, kategoriye ve türe göre filtrelemek. - Bütçe verilerini pasta grafiği ile görselleştirmek. - Toplam bakiyeyi güncellemek ve aylık özet/öneriler sunmak. - Kayıtları silme işlevselliği sağlamak.

MonthlyAdvisor.cs Analizi

`MonthlyAdvisor.cs` dosyası, kullanıcının aylık bütçe verilerine dayanarak finansal öneriler ve uyarılar oluşturan bir yardımcı sınıftır. Bu sınıf, belirli bir ay ve yıl için kullanıcının gelir ve giderlerini analiz eder ve çeşitli senaryolara göre tavsiyelerde bulunur.

Sınıflar: - `MonthlyAdvisor` : Aylık finansal öneriler oluşturan yardımcı sınıf.

Nesneler: - `MySqlConnection` , `MySqlCommand` , `MySqlDataReader` : Veritabanı işlemleri için kullanılan nesneler.

Kullanım Amacı: - Kullanıcının aylık gelir ve giderlerini analiz etmek. - Harcamaların gelirleri aşp aşmadığı, tasarruf oranları gibi konularda uyarılar ve tavsiyeler sunmak. - En çok harcama yapılan kategoriyi belirlemek. - Kullanıcıya finansal durumu hakkında bilgilendirici geri bildirim sağlamak.

Nesne Yapısı ve Sınıf Hiyerarşisi Özeti

Bu proje, bir bütçe takip uygulamasını temsil eden nesne tabanlı bir yapıyı kullanmaktadır. Temel olarak, uygulama Windows Forms (WinForms) tabanlı bir masaüstü uygulamasıdır ve C# dilinde geliştirilmiştir. Uygulama, kullanıcı arayüzü (UI) bileşenlerini ve veritabanı etkileşimini yönetmek için çeşitli sınıflar ve nesneler kullanır.

Ana Sınıflar ve İlişkileri:

1. `Program` Sınıfı:

- Uygulamanın başlangıç noktasıdır. `Main` metodu aracılığıyla `Form1` sınıfının bir örneğini oluşturur ve çalıştırır.
- Diğer sınıflarla doğrudan bir bağımlılığı yoktur, sadece uygulamanın yaşam döngüsünü başlatır.

2. `Form1` Sınıfı:

- Uygulamanın ilk kullanıcı arayüzü formudur (Giriş/Kayıt ekranı).
- Kullanıcı kimlik doğrulama ve yeni kullanıcı kaydı işlemlerini yönetir.
- `MySqlConnection` ve `MySqlCommand` nesnelerini kullanarak MySQL veritabanı ile etkileşime girer.
- Başarılı bir girişin ardından `icerik` sınıfının bir örneğini oluşturur ve gösterir.
- `icerik` sınıfına bağımlıdır.

3. `icerik` Sınıfı:

- Uygulamanın ana işlevselliğini içeren kullanıcı arayüzü formudur (Bütçe Takip ekranı).
- Kullanıcıların gelir ve giderlerini kaydetmelerini, filtrelemelerini ve görselleştirmelerini sağlar.

- `MySqlConnection` , `SqlCommand` ve `MySqlDataReader` nesnelerini kullanarak veritabanı ile etkileşime girer.
- `MonthlyAdvisor` sınıfının bir örneğini oluşturur ve aylık finansal öneriler almak için kullanır.
- `MonthlyAdvisor` sınıfına bağımlıdır.
- Guna.UI2 kütüphanesinden çeşitli GUI bileşenlerini (`Guna2GroupBox` , `Guna2TextBox` , `Guna2Button` , vb.) kullanır.
- `System.Windows.Forms.DataVisualization.Charting` kütüphanesinden `Chart` nesnesini kullanarak pasta grafikleri oluşturur.

4. `MonthlyAdvisor` Sınıfı:

- `icerik` sınıfı tarafından kullanılan bir yardımcı sınıftır.
- Kullanıcının aylık bütçe verilerini analiz eder ve finansal öneriler (`List<string>`) döndürür.
- `MySqlConnection` , `SqlCommand` ve `MySqlDataReader` nesnelerini kullanarak veritabanından bütçe verilerini çeker.
- Herhangi bir kullanıcı arayüzü bileşeni içermez; tamamen iş mantığına odaklanmıştır.

Veritabanı Yapısı (`butce_takip.sql`):

- **users Tablosu:** Kullanıcı kimlik bilgilerini (`username` , `password`) saklar.
- **budget_entries Tablosu:** Bütçe girişlerini (`username` , `category` , `amount` , `description` , `date`) saklar.

Nesneler Arasındaki İlişkiler:

- **Program -> Form1:** Program sınıfı, uygulamanın başlangıcında `Form1` nesnesini oluşturur.
- **Form1 -> icerik:** `Form1` sınıfı, başarılı girişin ardından `icerik` nesnesini oluşturur ve gösterir.
- **icerik -> MonthlyAdvisor:** `icerik` sınıfı, aylık finansal öneriler almak için `MonthlyAdvisor` nesnesini kullanır.
- **Tüm UI Formları (Form1 , icerik) ve MonthlyAdvisor -> MySQL Veritabanı:** Bu sınıflar, `MySqlConnection` ve `SqlCommand` nesneleri aracılığıyla `butce_takip` veritabanı ile etkileşime girer.

Genel Nesne Yapısı:

Uygulama, kullanıcı arayüzü katmanı (Formlar) ve iş mantığı katmanı (`MonthlyAdvisor`) arasında bir ayırım yapmaya çalışır. Veritabanı etkileşimleri her iki katmanda da

doğrudan gerçekleştirilir. Bu yapı, modülerliği ve yeniden kullanılabilirliği artırır, ancak veritabanı erişiminin doğrudan UI katmanında yapılması, daha büyük projelerde bakım zorluklarına yol açabilir. Proje, WinForms'un olay tabanlı programlama modelini yoğun bir şekilde kullanır.

Sınıfların ve Nesnelerin Genel Kullanım Amaçları

Bu nesne tabanlı projede sınıflar ve nesneler, uygulamanın farklı sorumluluklarını ve işlevselliklerini düzenli ve yönetilebilir bir şekilde ayırmak için kullanılmıştır. Her sınıf belirli bir amaca hizmet eder ve bu amaç doğrultusunda kendi içinde veri (özellikler) ve davranışları (metotlar) barındırır.

Genel Kullanım Amaçları:

1. Modülerlik ve Yeniden Kullanılabilirlik:

- Her bir sınıf (örneğin `Form1`, `icerik`, `MonthlyAdvisor`), uygulamanın belirli bir modülünü veya işlevselliğini temsil eder. Bu, kodun daha küçük, bağımsız parçalara ayrılmasını sağlar. Bu modüller, uygulamanın farklı yerlerinde veya gelecekteki projelerde yeniden kullanılabilir hale gelir.
- Örneğin, `MonthlyAdvisor` sınıfı, finansal öneri mantığını UI'dan ayırarak, bu mantığın başka bir bağlamda da kullanılabilmesine olanak tanır.

2. Sorumlulukların Ayrılması (Separation of Concerns):

- Her sınıfın belirli bir sorumluluğu vardır. `Form1` kullanıcı kimlik doğrulama ile ilgilenirken, `icerik` bütçe girişlerini yönetir ve `MonthlyAdvisor` finansal analiz yapar. Bu ayırım, kodun okunabilirliğini, anlaşılabilirliğini ve bakımını kolaylaştırır.
- Bir hata oluştuğunda veya yeni bir özellik eklendiğinde, ilgili sınıfın kolayca bulunup değiştirilebilmesini sağlar.

3. Veri Kapsülleme (Encapsulation):

- Sınıflar, verileri (özellikleri) ve bu veriler üzerinde çalışan metotları bir araya getirerek kapsülleme sağlar. Bu, verilerin dışarıdan doğrudan erişimini kısıtlayarak, verilerin bütünlüğünü korur ve sınıfın iç işleyişinin dış dünyaya karşı gizlenmesini sağlar.
- Örneğin, `icerik` sınıfındaki `connectionString` gibi hassas veriler, sınıfın içinde tutulur ve sadece sınıfın kendi metotları tarafından kullanılır.

4. Soyutlama (Abstraction):

- Sınıflar, karmaşık sistemlerin basitleştirilmiş bir görünümünü sunar. Kullanıcı veya diğer sınıflar, bir sınıfın iç detaylarını bilmek zorunda kalmadan onunla etkileşime girebilir.
- Örneğin, `icerik` sınıfı `MonthlyAdvisor` sınıfını kullandığında, `MonthlyAdvisor` 'ın önerileri nasıl hesapladığını bilmesine gerek yoktur; sadece `GenerateSuggestions()` metodunu çağırarak sonuçları alır.

5. Kalıtım (Inheritance) ve Polimorfizm (Polymorphism) (WinForms Bağlamında):

- WinForms uygulamalarında, `Form` sınıfından kalıtım almak yaygın bir kullanımdır (`Form1` ve `icerik` sınıfları `Form` sınıfından türetilmiştir). Bu, temel form işlevselliğini yeniden kullanmayı ve özelleştirmeyi sağlar.
- Her ne kadar bu projede açıkça polimorfik davranışlar sergilenmese de, nesne tabanlı programlamanın bu prensipleri, uygulamanın genişletilebilirliğini ve esnekliğini artırır.

6. Veritabanı Etkileşimi:

- `MySqlConnection` , `MySqlCommand` gibi nesneler, uygulamanın veritabanı ile güvenli ve yapılandırılmış bir şekilde iletişim kurmasını sağlar. Bu nesneler, SQL sorgularını yürütmek, parametreleri güvenli bir şekilde bağlamak ve sonuçları okumak için standart bir arayüz sunar.

Sonuç olarak, bu projedeki sınıflar ve nesneler, uygulamanın işlevselliğini mantıksal birimlere ayırarak, kodun daha düzenli, bakımı kolay ve anlaşılır olmasını sağlamıştır. Bu, nesne tabanlı programlamanın temel prensiplerinin pratik bir uygulamasını göstermektedir.