İnönü Üniversitesi
Bilgisayar Mühendisliği Bölümü
2024 Bahar Nesneye Yönelik Yazılım Mühendisliği (376)
Proje Analiz Rapor

Ders Sorumlusu	Dr. Öğr. Üyesi Ahmet Arif AYDIN
Öğrenci(ler)	Metin Ilgar Mutlu - 02225076042
Proje İsmi	Language Translator (LangT)
Aşama Tanımı	3. Aşama: Kodlama (Implementation)

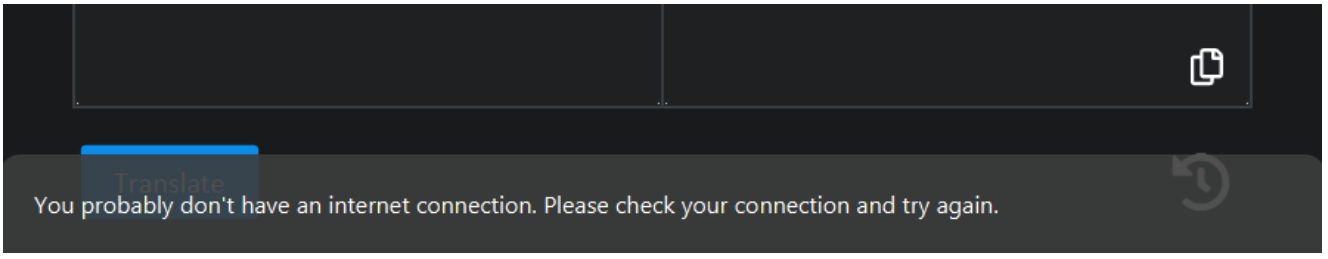
AÇIKLAMALAR

3. Belirtilen Aşamaların Kodlanması ve Kullanılan Tasarım Kalıpları

Creational (Oluşturucu) Tasarım Kalıplarından : Singleton

Singleton, bir sınıfın yalnızca bir örneğe sahip olmasını sağlarken bu örneğe küresel bir erişim noktası sağlamanıza olanak tanıyan oluşturucu bir tasarım modelidir. Projemde Snackbar ve LanguageTranslator sınıflarında Singleton modeli kullanıyorum.

Snackbar sınıfım görevi herhangi olağan dışı bir durumda kullanıcıyı bilgilendirmek için pencerenin alt kısmından açılan bir bilgilendirme mesajıdır.



Snackbar sınıfı uygulamanın farklı yerlerinde kullanıcıya farklı uyarı mesajları göstermek için kullanılmıştır.Snackbar'ın yalnızca bir örneğinin oluşturulduğundan emin olmak için Singleton modelini kullanır. Bu, Constructor'ı private hale getirerek ve yalnızca bir Snackbar örneğinin oluşturulmasını sağlayan getInstance(Pane pane) statik metodunu sağlayarak elde edilir.

```
public class Snackbar {
    // Class ile aynı türde static bir değişken oluşturulur.
    private static Snackbar instance;

    // Constructor private olmalı. Bu yapılan işlem new ile nesne
    // oluşturulmasını engeller.
    private Snackbar(Pane pane) {
        JFXsnackbar = new JFXSnackbar(pane);
    }

   // Static değişkene ulaşmak için static bir metot oluşturulmalıdır.
    public static Snackbar getInstance(Pane pane) {
        if (instance == null) {
            instance = new Snackbar(pane);
        }
        return instance;
    }
```

LanguageTranslator sınıfı DeepL API ile iletişimi sağlayan bir sınıftır. Çeviri işlemleri, kaynak ve hedef dillere erişim buradan yapılır. Benzer şekilde LanguageTranslator sınıfı da Singleton modelini kullanır. Constructor'ı private ve LanguageTranslator örneğini almak için statik bir getInstance() metodu sağlanır.

```
public class LanguageTranslator {
    // Class ile aynı türde static bir değişken oluşturulur.
    private static LanguageTranslator instance;

    // Constructor private olmalı. Bu yapılan işlem new ile nesne
    // oluşturulmasını engeller.
    private LanguageTranslator() throws DeepLException, InterruptedException {
        translator = new Translator("api_key");
        this.sourceLanguages = translator.getSourceLanguages();
        this.targetLanguages = translator.getTargetLanguages();
    }

    // Static değişkene ulaşmak için static bir metot oluşturulmalıdır.
    public static LanguageTranslator getInstance() {
        if (instance == null){
            try {
                instance = new LanguageTranslator();
            } catch (DeepLException | InterruptedException e) {
                throw new RuntimeException(e);
            }
        }
        return instance;
    }
}
```