

Typing Tutor Uygulama Raporu

Hazırlayan: Safia Ashraf Said

Öğrenci Numarası: 23411182

Proje Adı: Typing Tutor

1. Tasarım Süreci

Bu projenin amacı, kullanıcıların klavye üzerindeki harf, rakam ve özel karakterleri daha verimli yazabilmeleri için alıştırma yapabilecekleri bir uygulama geliştirmektir. Proje Java programlama dili kullanılarak, Swing kütüphanesi aracılığıyla grafiksel kullanıcı arayüzü (GUI) oluşturularak geliştirilmiştir.

Ana pencereyi oluşturmak için **TypingTutorFrame** adlı bir sınıf tanımlanmış ve bu sınıf **JFrame** sınıfından kalıtım almıştır.

Arayüzde yer alan başlıca bileşenler şunlardır:

- Kullanıcının metin yazabileceği bir **JTextArea**,
- Harf, rakam ve özel karakterleri temsil eden **JBUTTON**'lar,
- Space, Enter, Backspace, Caps Lock, Shift gibi özel tuşlar,
- Yukarı, aşağı, sağ ve sol yön tuşları.

Klavye düzeni olarak QWERTY dizilimi esas alınmış ve tuşlar yatay satırlara uygun şekilde hizalanmıştır. Tuşların oluşturulması için **createKeyButton** ve **createSpecialButton** gibi yardımcı metotlar kullanılmıştır. Görsellik ve işlevsellik birlikte düşünülerek, kullanıcı dostu bir yapı sağlanmıştır.

2. Sınıflar ve Metot Açıklamaları

Kullanılan Sınıf ve Değişkenler:

- TypingTutorFrame**: Ana uygulama penceresini ve tüm bileşenleri içeren sınıftır.
- JTextArea textArea**: Kullanıcının yazdığı metni gösteren yazı alanıdır.

- `Map<Character, JButton> keyMap`: Her karakteri ilgili `JButton` ile eşleştirmek için kullanılır.
- `boolean capsOn`: Caps Lock tuşunun aktif olup olmadığını belirten durum değişkenidir.

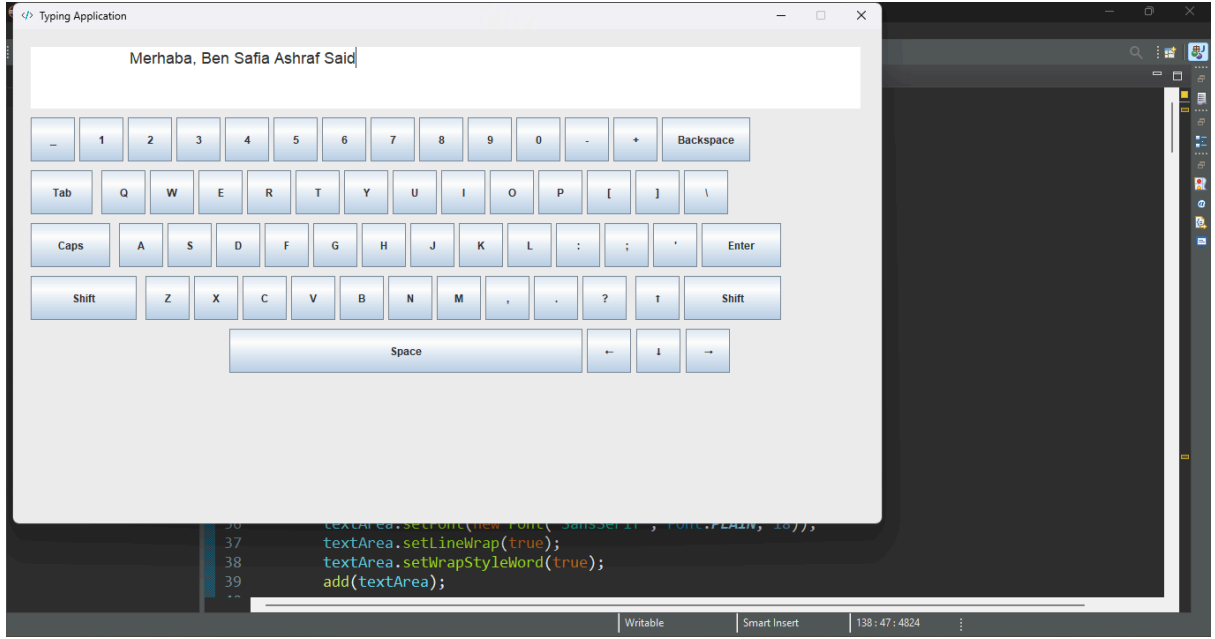
Önemli Metotlar:

- `createKeyButton(char c, int x, int y)`: Belirtilen karakter ve koordinatlar ile standart bir harf tuşu oluşturur.
- `createSpecialButton(String label, int x, int y, int w, int h, char value)`: Özel tuşları (örneğin Space, Enter) oluşturmak için kullanılır.
- `updateLetterKeys()`: Caps Lock durumuna göre harf tuşlarının üzerindeki yazıları günceller.
- `changeButtonColor(JButton btn)`: Bir tuşa basıldığında geçici olarak tuş rengini değiştirir.
- `createArrowButton(...)`: Yön tuşlarını oluşturur ve imlecin (caret) konumunu değiştirir.

3. Karşılaşılan Zorluklar ve Çözümler

- Tuşların Düzensiz Yerleşimi: Başlangıçta tuşlar düzenli görünmüyordu. Her satır için farklı bir X başlangıç noktası verilerek hizalama sorunu çözüldü.
 - Caps Lock İşlevselliği: İlk denemede Caps Lock tuşu sadece tuş üzerindeki harfleri değiştiriyordu ama yazılan metin büyük kalıyordu. Bu durum, her tuşa basıldığında `capsOn` durumunun dikkate alınmasıyla düzeltildi.
 - Yön Tuşlarının Etkisizliği: İlk sürümde yön tuşları sadece simge olarak görünüyordu. Daha sonra `setCaretPosition()` kullanılarak imleç konumu değiştirilebilir hale getirildi.
 - Shift Tuşu: Şu anda yalnızca görsel olarak eklenmiş durumda ve renklendiriliyor. Gerçek işlevi olan sembollerle yazma özelliği sonraki sürümlerde eklenebilir.
-

4. Ekran Görüntüsü



5. Dosya Yapısı

Dosya Adı	Açıklama
<code>TypingTutorFrame.java</code>	Uygulamanın ana kod dosyası
<code>7468307.png</code>	Uygulama penceresinin simge görseli
<code>report.docx/pdf</code>	Bu rapor belgesi

✓ Genel Değerlendirme:

Bu uygulama, klavye kullanımı konusunda pratik yapmak isteyen kullanıcılar için temel bir yazım aracı sunmaktadır. Arayüz sade, anlaşılır ve geliştirilmeye açık olarak tasarlanmıştır. Gelecekte Shift tuşu gibi işlevler aktif hale getirilebilir ve kullanıcıya hız ve doğruluk geri bildirimi verecek özellikler eklenebilir.