JAVA SWING

JFC (Java Foundation Class) ve Swing

- Java dilinde iki grafik ortamı tanımlanmıştır. Bunlar awt ve swing kütüphaneleridir.
- Bir çok programı hazır olarak java kütüphanelerinden alacağız, bir kısmını da burada temel kalıplar olarak kullanılmak üzere hazırlanmıştır.
- Java grafik sistemini analamak için anlamamız gereken ilk kavram koordinat sistemidir.
- Koordinat birimi pixel (ekran görüntü elemanı) Digital bilgisayarda ekran görüntüsü bir çok pixelin bir arada kullanılmasıyla oluşur.
- Her bir pixel'in rengi, parlaklığı gibi çeşitli fiziksel özellikleri değiştirilerek ekranda veyakullanılan grafik ortamında görüntü oluşur.
- Java'da ve diğer bilgisayar grafik sistemlerinde piksel koordinatları karteziyen koordinat sistemi gibi düzenlenmiştir, ancak y ekseni aşağı doğru yönelmiştir.
- Bunun temel sebebi bilgisayar grafiklerinin ilk defa printerlar kullanılarak oluşturulmuş olması ve satır satır ileriye doğru giden printerlarda geriye doğru hareket imkanı bulunmamasıydı.

- Günümüzde böyle çalışma gereksinimi olmamasına rağmen , bu tarihi sebepten dolayı grafik sistemi bu şekilde standartlaştırılmıştır.

KAYNAK 1. A. Kadir GÜNEYTEPE: Java Swing, **Başlangıçtan İleri Düzeye, Her Yönüyle GUI Geliştirme Klavuzu**

KAYNAK 2. M. TURHAN ÇOBAN: **JAVA BİLGİSAYAR DİLİYLE PROGRAMLAMA**

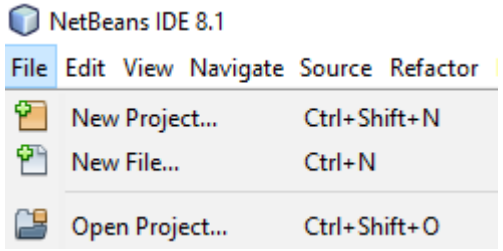
- Swing, yukarıda da belirtildiği gibi, Java için geliştirilmiş ve **JFC**'nın bir parçası olan, görsel bileşenler içeren ve grafiksel arayüze sahip uygulamalar geliştirmeyi sağlayan bir **API**'dir.

- Swing öncülü olan **AWT**'den (**Abstract Window Toolkit**) daha gelişmiş ve daha iyi donatılmış GUI bileşenleri içerir; ayrıca farklı platformlara kolayca uyum sağlayabilmesine izin veren **Eklenebilir Görünüm ve Davranış**'ı destekler.

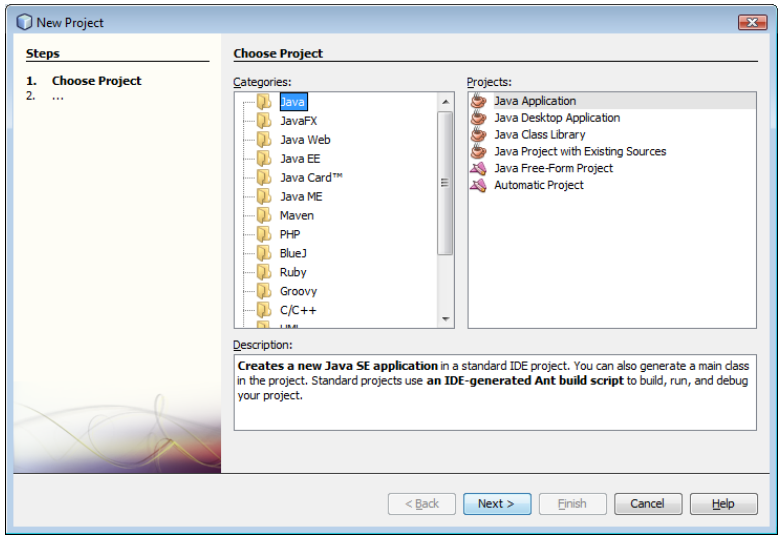
JFC, grafiksel kullanıcı arayüzü (GUI) geliştirmek için gerekli özellikler ve Java uygulamaları için zengin grafiksel işlevsellik ve etkileşim sağlayan sınıflar topluluğudur.

Uygulama Oluşturma:

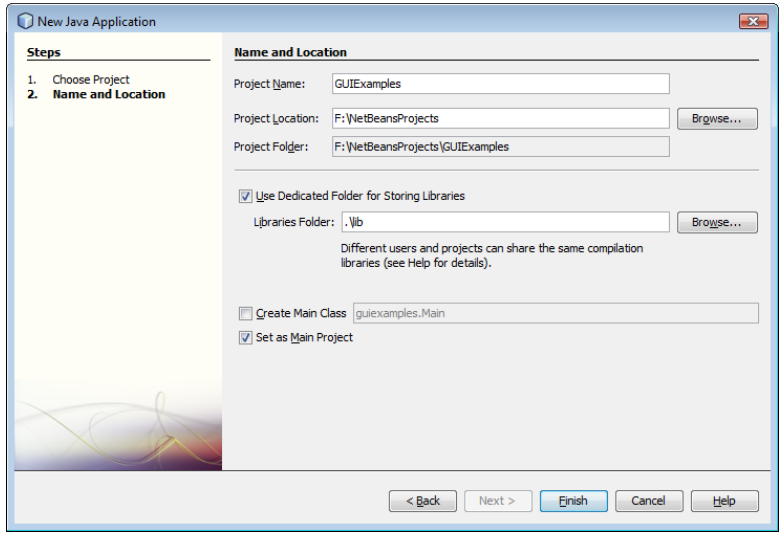
1. **NetBeans** üzerinde **File** menüsünden **New Project** seçilir.



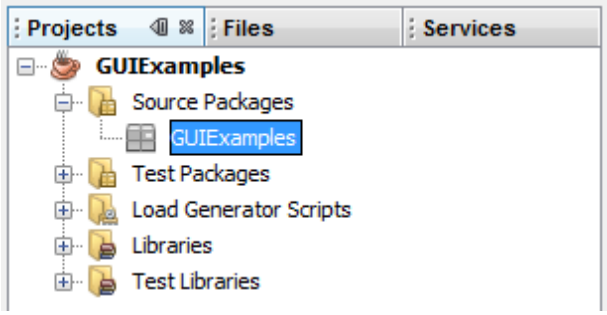
2. **Java Uygulamasını Seçilir**



3. Uygulama Adının Belirlenmesi

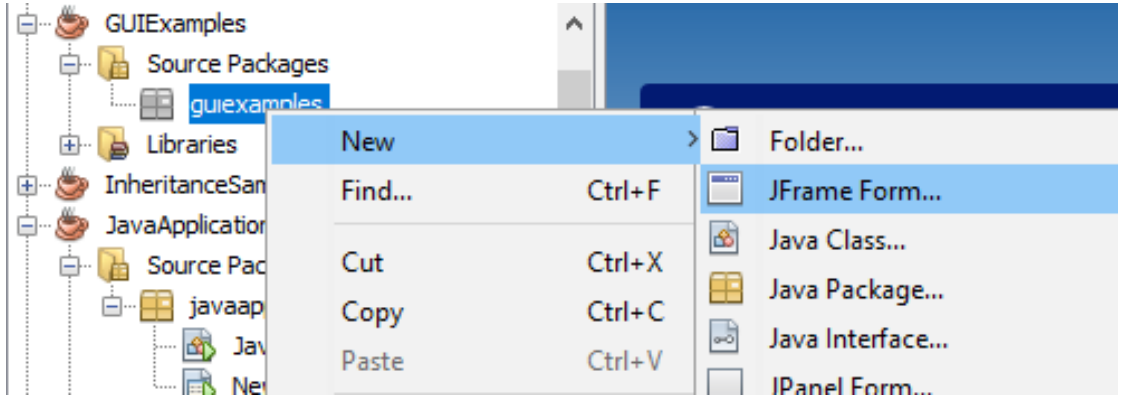


İşlem sonlandırıldığında, **NetBeans Projects** penceresinde aşağıdaki gibi bir görünüm oluşur.



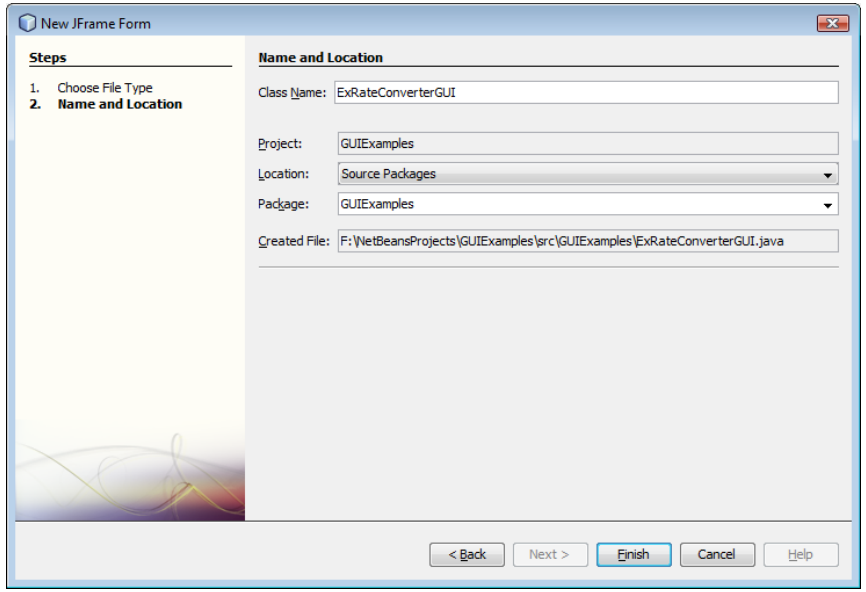
4. JFrame Form Ekleme

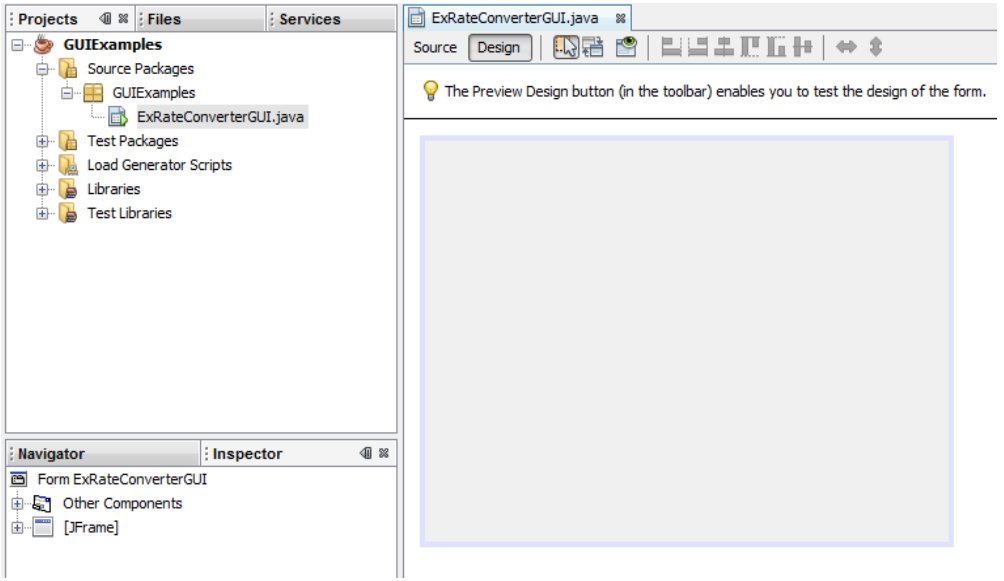
GUIExamples adlı proje üzerinde farenin sağ düğmesine tıklayarak, **New -> JFrame Form** seçilir.



5. GUI Sınıfını Adlandırma

Name and Location alanında, **Class Name** bölümünde sınıf adı girilir. Sonuçta sağdaki gibi bir görünüm ortaya çıkar.

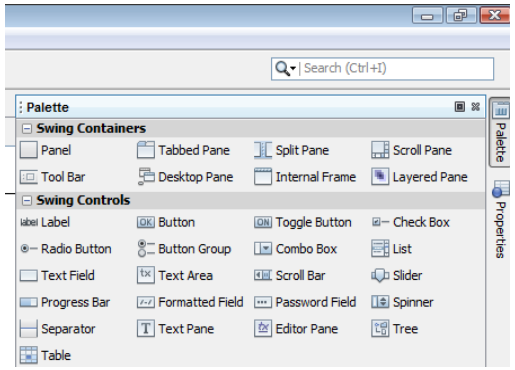




NetBeans Arayüzü

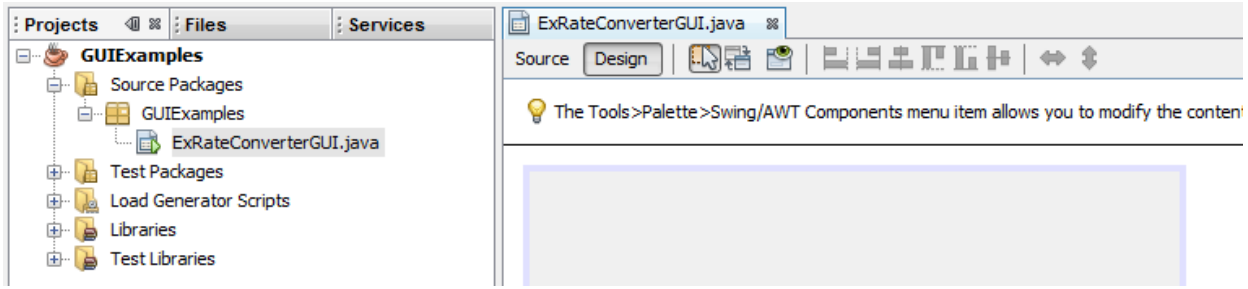
NetBeans geliştirme ortamının temel bazı bölümleri **Palette**, **Design Area**, **Property Editor** ve **Inspector** pencereleridir.

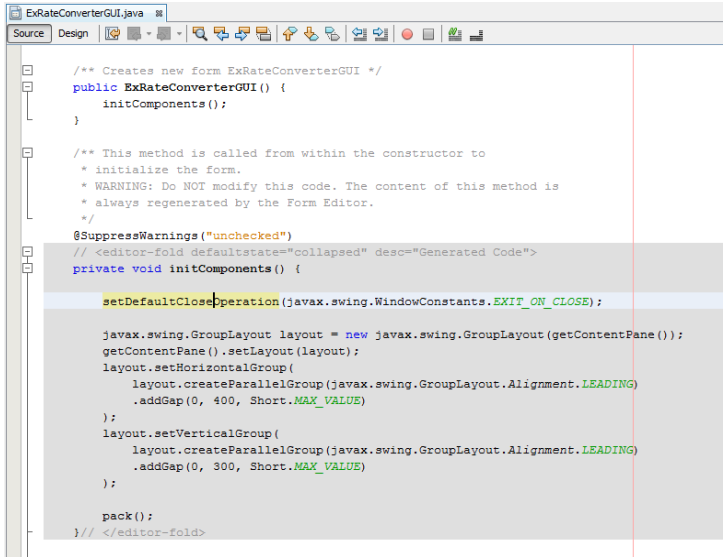
Palette, Swing API tarafından sağlanan bileşenleri içerir. **Palette** penceresinden, formun üzerine sürükle bırak yöntemiyle istenilen bileşen yerleştirilebilir.



Design Alanı

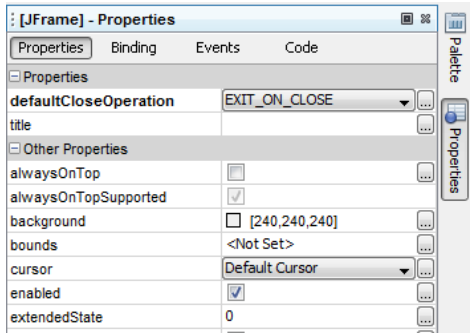
Design Area, uygulamamızın görsel olarak oluşturulacağı alandır. **Source** ve **Design** olmak üzere iki bölümden oluşur. Bu iki bölüm arasında kendi adlarını taşıyan geçişli düğmelere basarak geçiş yapılabilir.





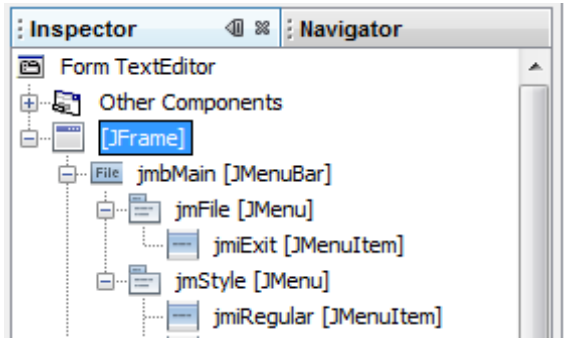
Property Editor

Property Editor, forma eklediğimiz tüm bileşenlerin özelliklerini düzenleyebilmemizi sağlar.
Değiştirmek istenen özelliğe tıklanıp bu özellik rahatlıkla düzenlenebilir.
Böylece kod eklenmeden istenilen bir bileşenin özelliği kolayca değiştirilebilir. Aşağıda JFrame nesnesi seçili iken **Property Editor**'un görünümü verilmiştir.

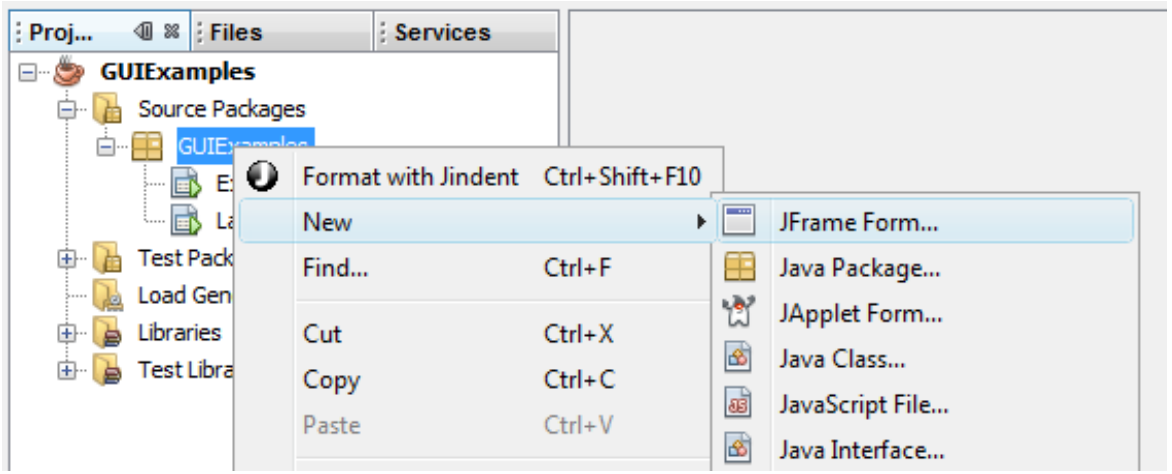


Inspector

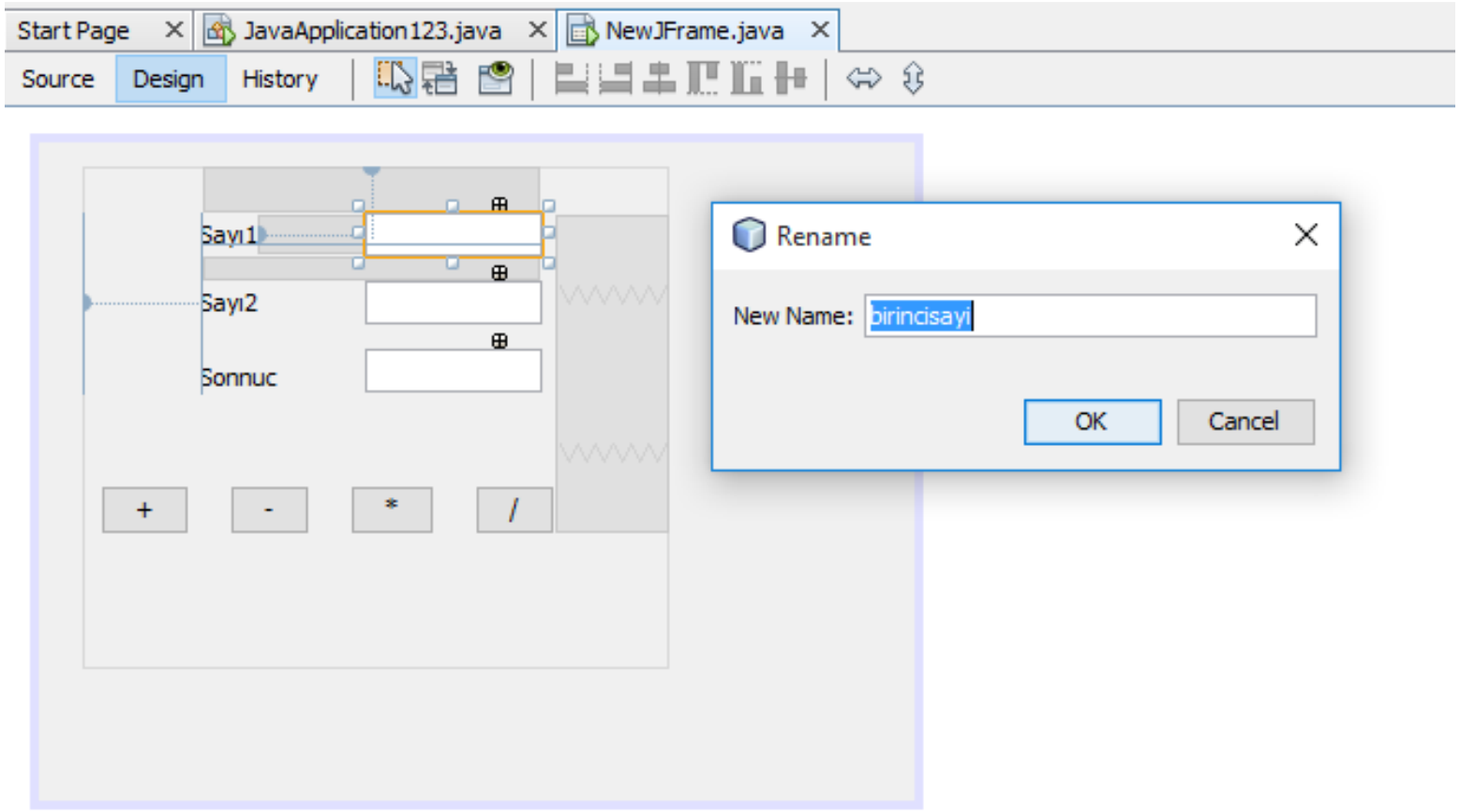
Inspector, uygulamamızda kullandığımız bileşenlerin görsel sunumunu sağlar.
Bileşen sıradüzeni buradan izlenebilir. Bu bölüm aynı zamanda bileşen adlarını değiştirmek için de kullanılabilir.
Aşağıda bir proje bileşen sıradüzenini gösteren bir **Inspector** penceresi görünümü verilmiştir.



Source Packages üzerinde sağ tıkla açılan menüden, **JFrame Form** seçeneği seçebilir.



2. Adlandırma

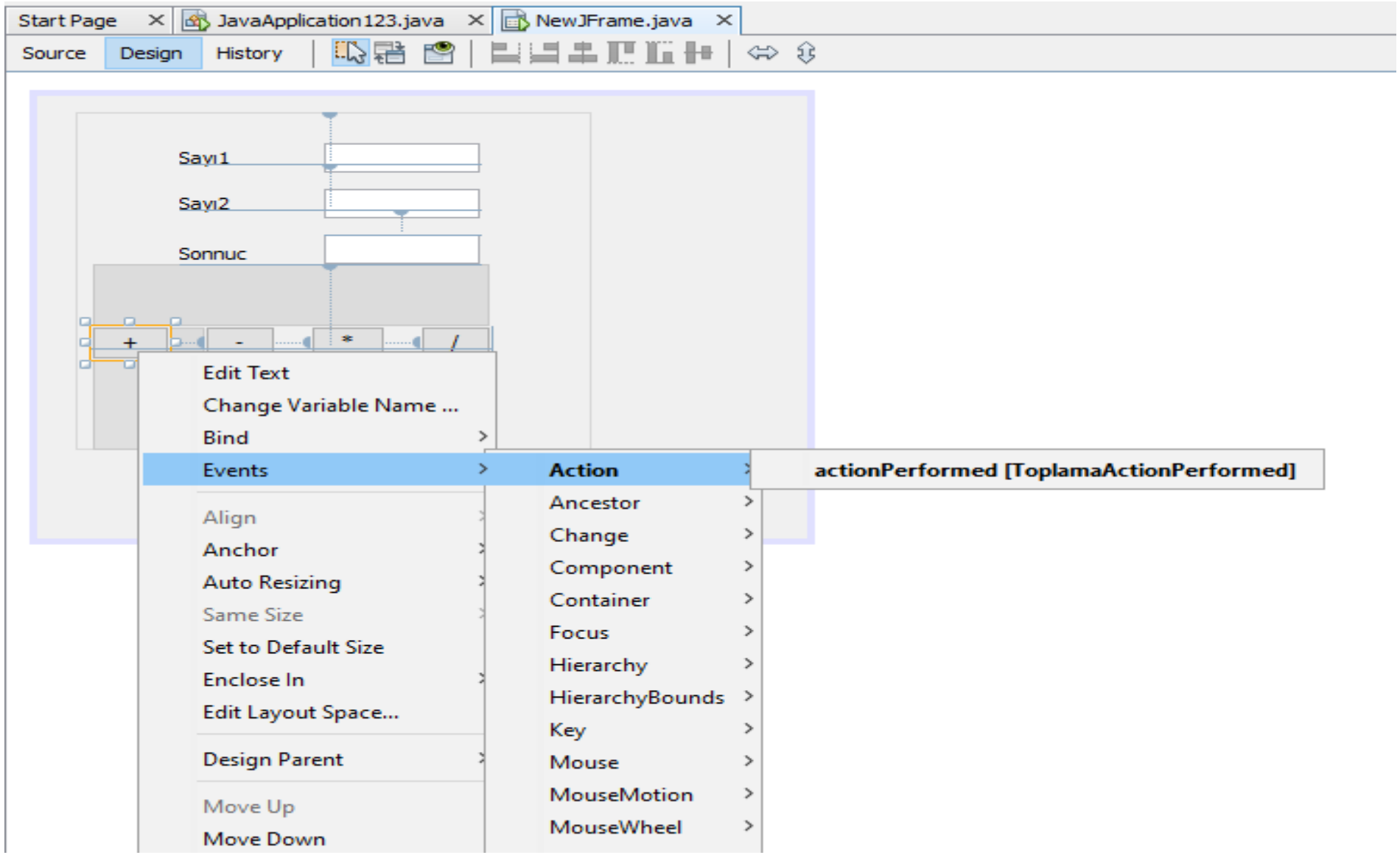


3. Değişken tanımlama

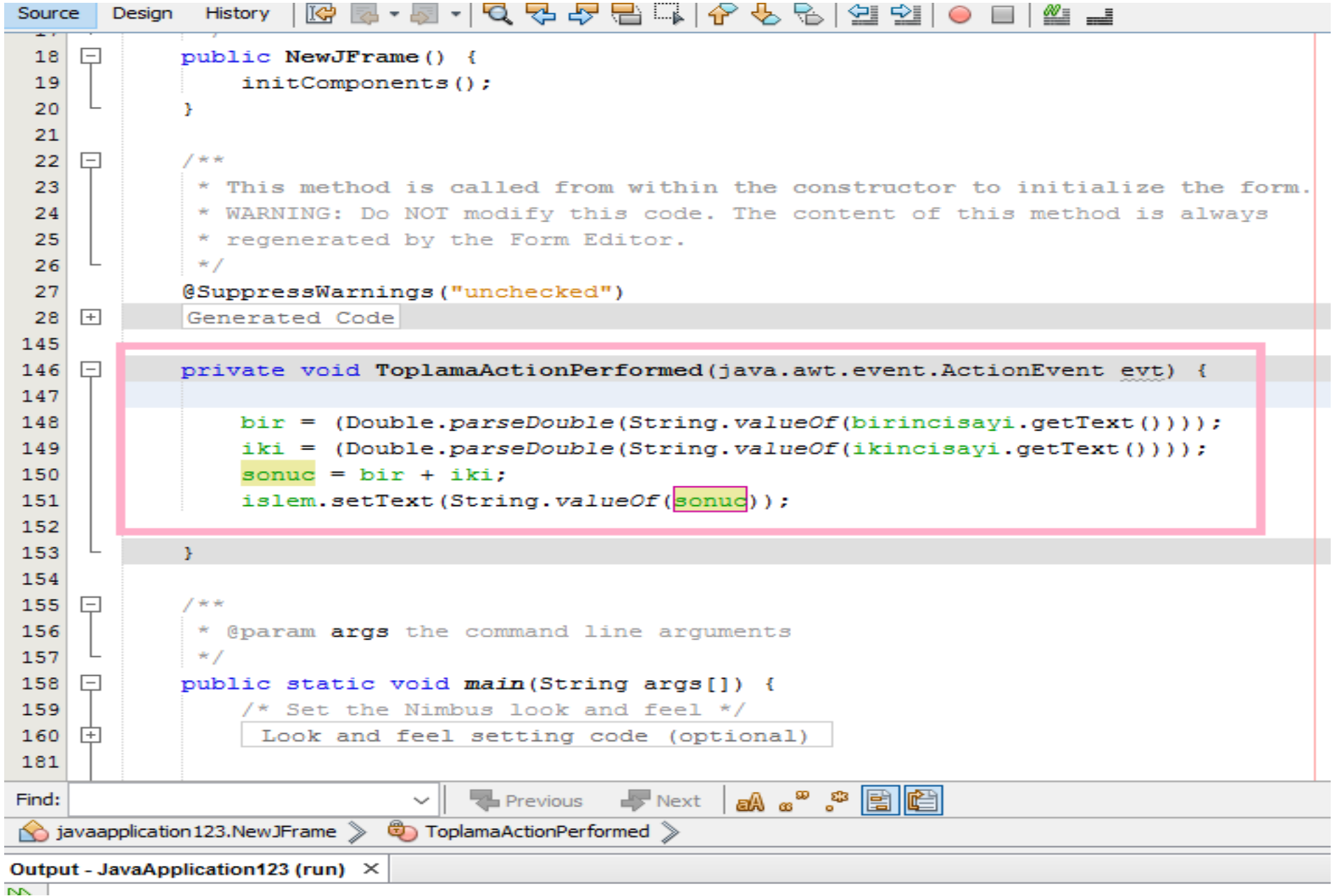
Source Design History

```
1  /*
2   * To change this license header, choose License Headers in Project Properties.
3   * To change this template file, choose Tools | Templates
4   * and open the template in the editor.
5   */
6  package javaapplication123;
7
8  /**
9   *
10  * @author Caner Koç
11  */
12 public class NewJFrame extends javax.swing.JFrame {
13
14     double bir, iki, sonuc;
15     /**
16      * Creates new form NewJFrame
17      */
18     public NewJFrame() {
19         initComponents();
20     }
21
22     /**
23      * This method is called from within the constructor to initialize the form.
24      * WARNING: Do NOT modify this code. The content of this method is always
25      * regenerated by the Form Editor.
26      */
27     @SuppressWarnings("unchecked")
28     Generated Code
```

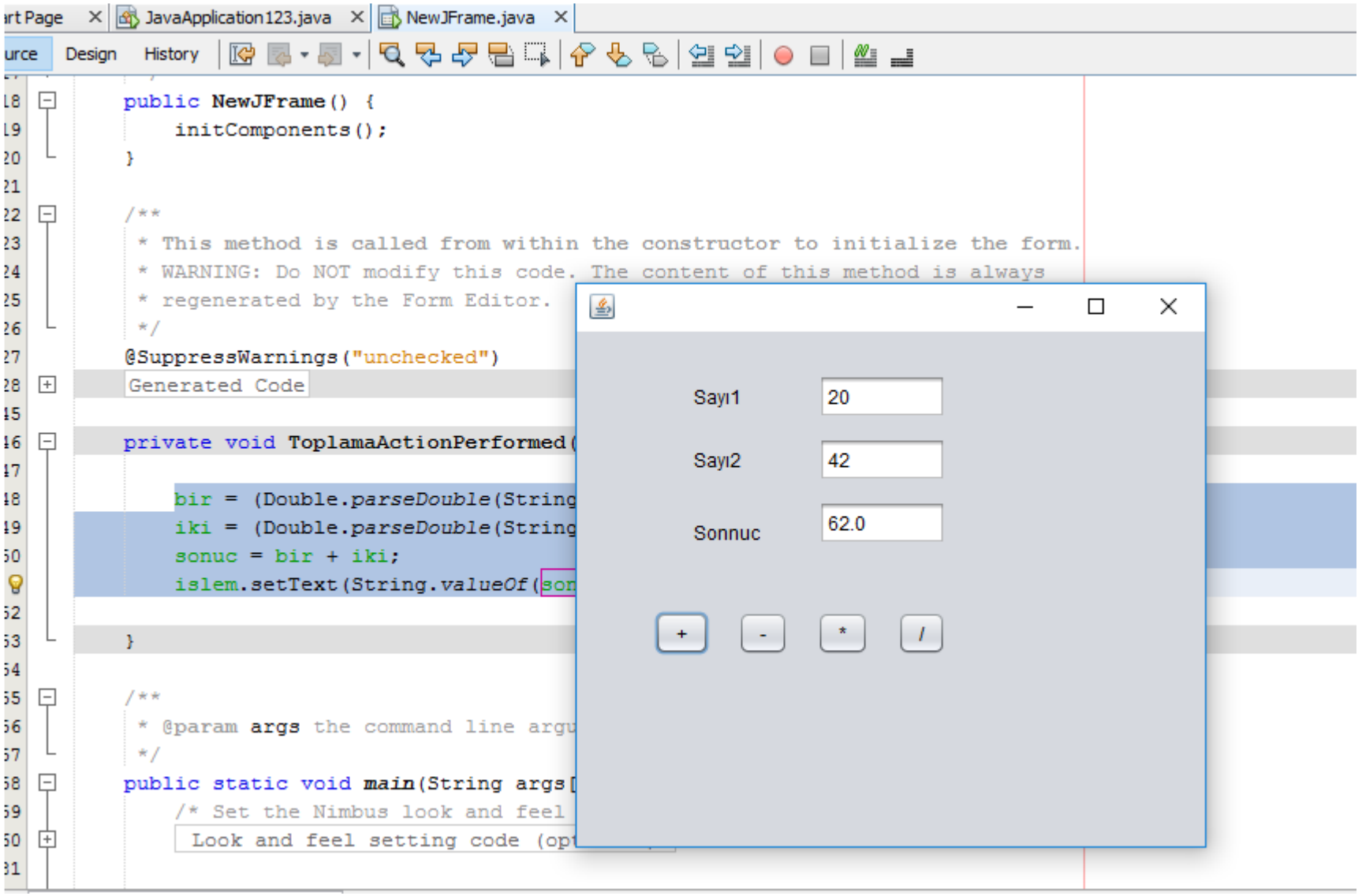
4. Komuta görev atama



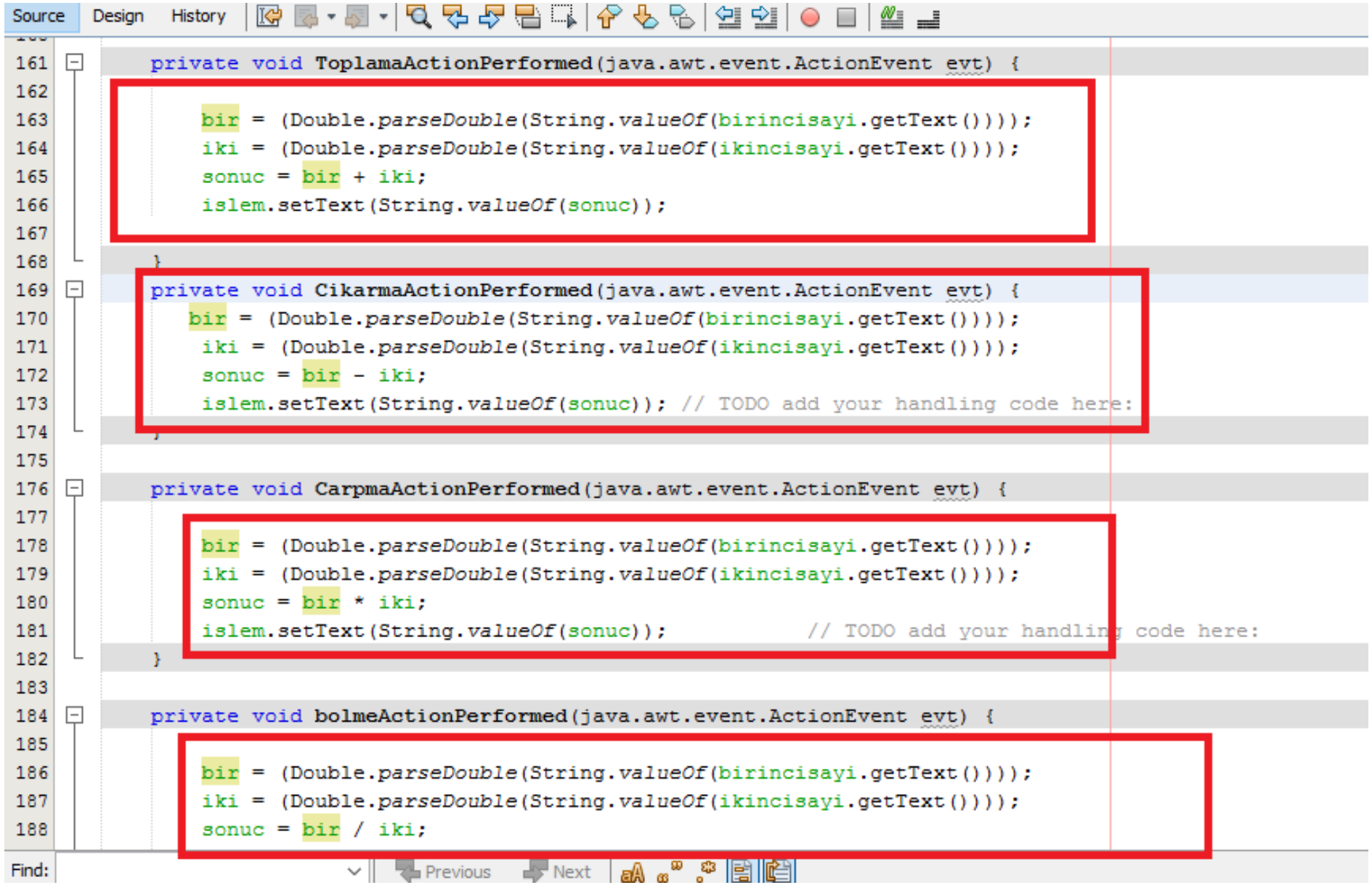
5. Komut yazma



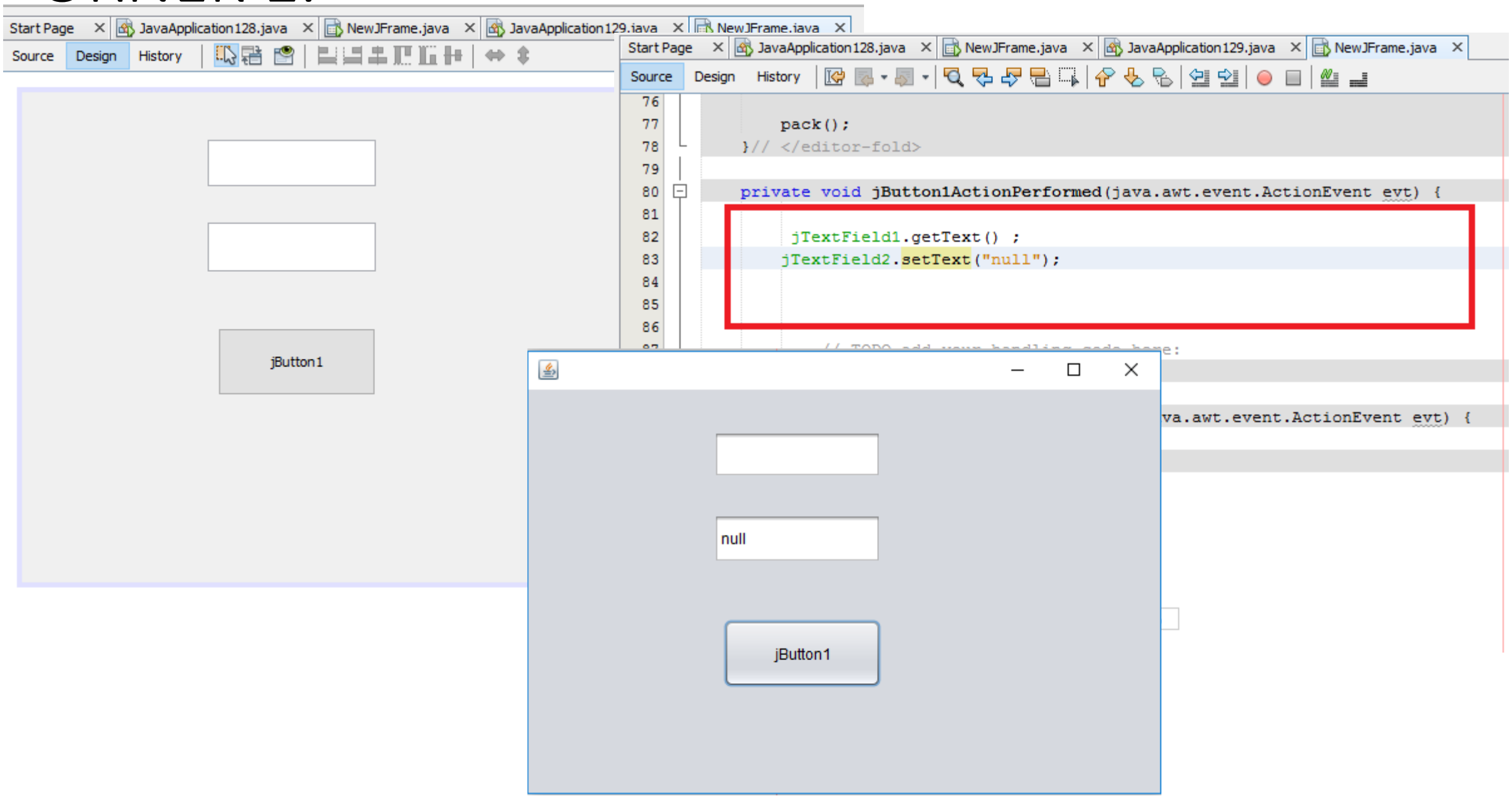
6. İşlem yapma

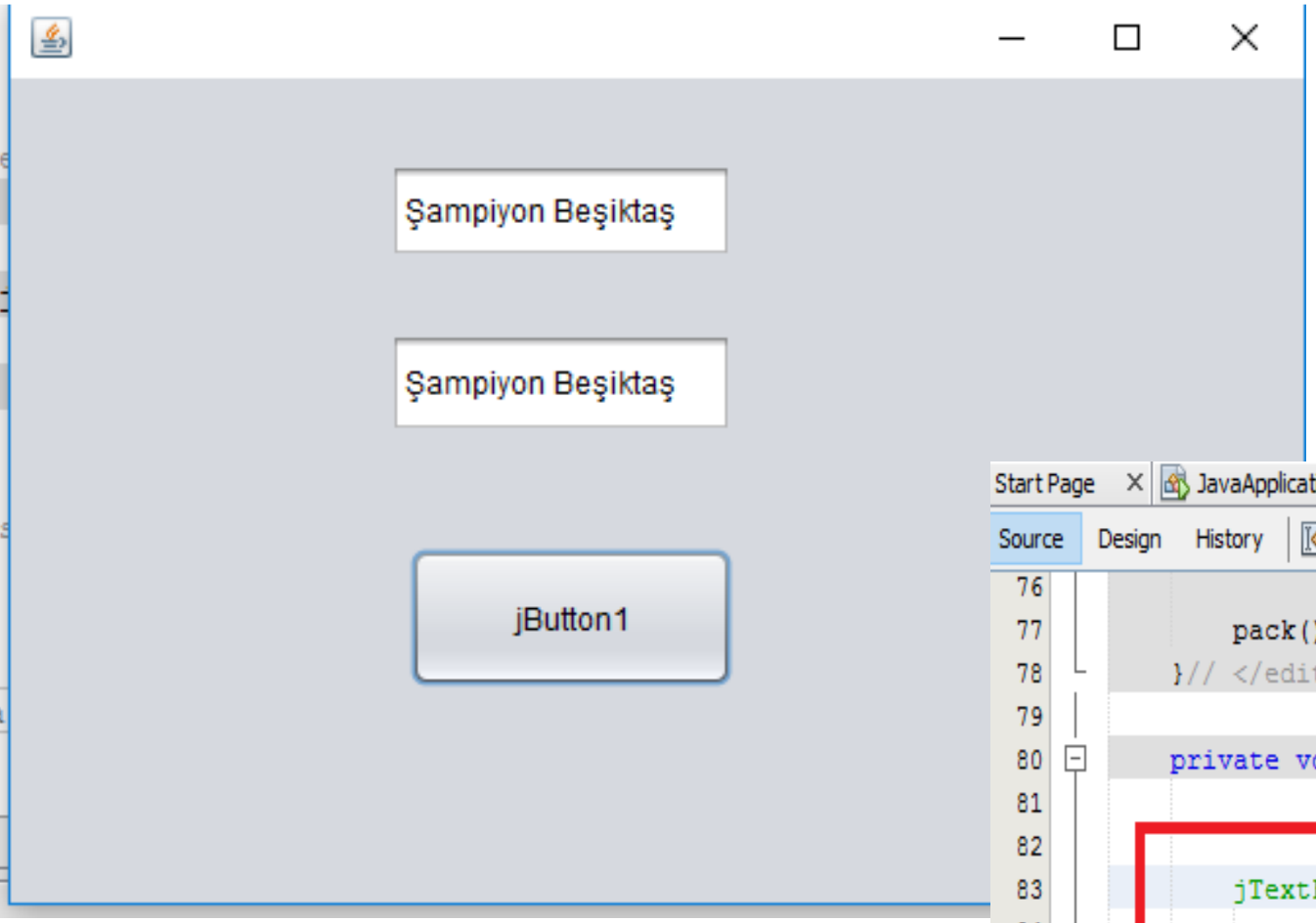


7. İşlemi çoğaltma



ÖRNEK 1.


Start Page
JavaApplication128.java
NewJFrame.java
JavaApplication129.java
NewJFrame.java
Source
Design
History
jButton1
Start Page
JavaApplication128.java
NewJFrame.java
JavaApplication129.java
NewJFrame.java
Source
Design
History
76 pack();
77 }// </editor-fold>
78
79 private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
80
81 jTextField1.getText() ;
82 jTextField2.setText("null");
83
84
85
86
va.awt.event.ActionEvent evt) {
null
jButton1



jTextField1.getText();

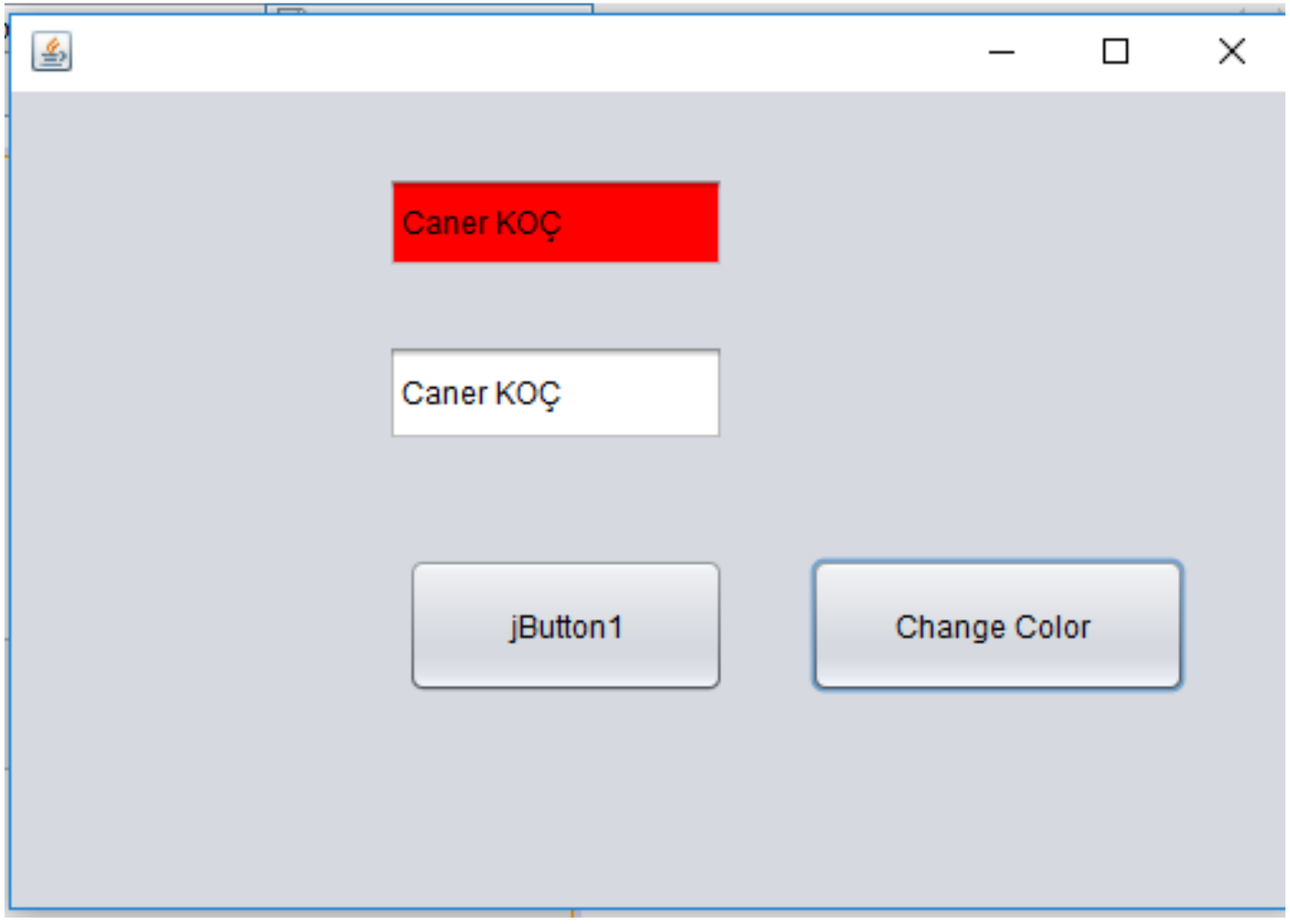
jTextField2.setText();

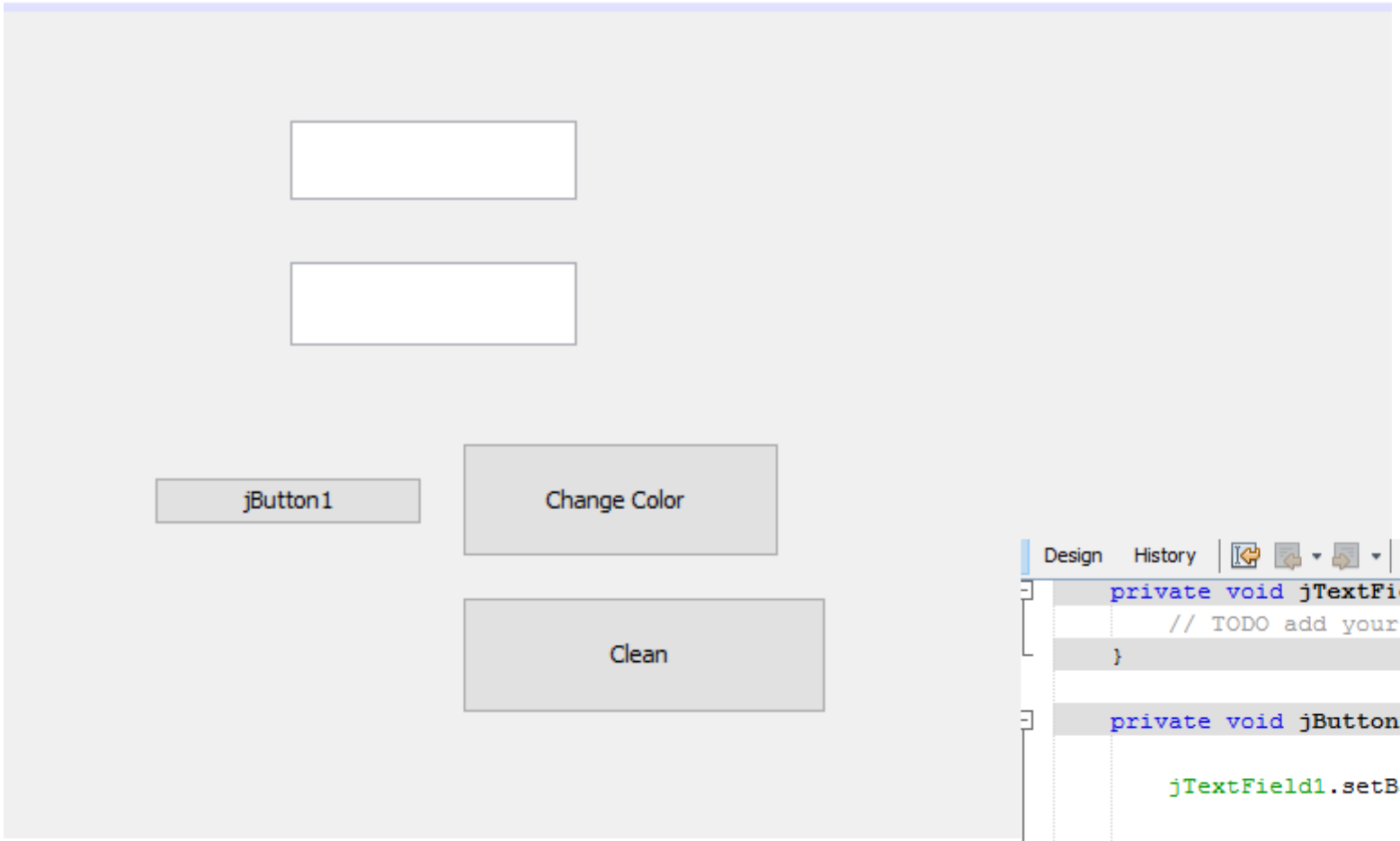
Start Page | JavaApplication128.java | NewJFrame.java | JavaApplication129.java | NewJFrame.java

Source | Design | History

```
76
77          pack();
78      }// </editor-fold>
79
80      private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
81
82
83          jTextField2.setText( jTextField1.getText());
84
85
86
87              // TODO add your handling code here:
88      }
89
90      private void jTextField2ActionPerformed(java.awt.event.ActionEvent evt) {
91          // TODO add your handling code here:
```

```
private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
    jTextField1.setbackground(Color.red);

}

/**
```

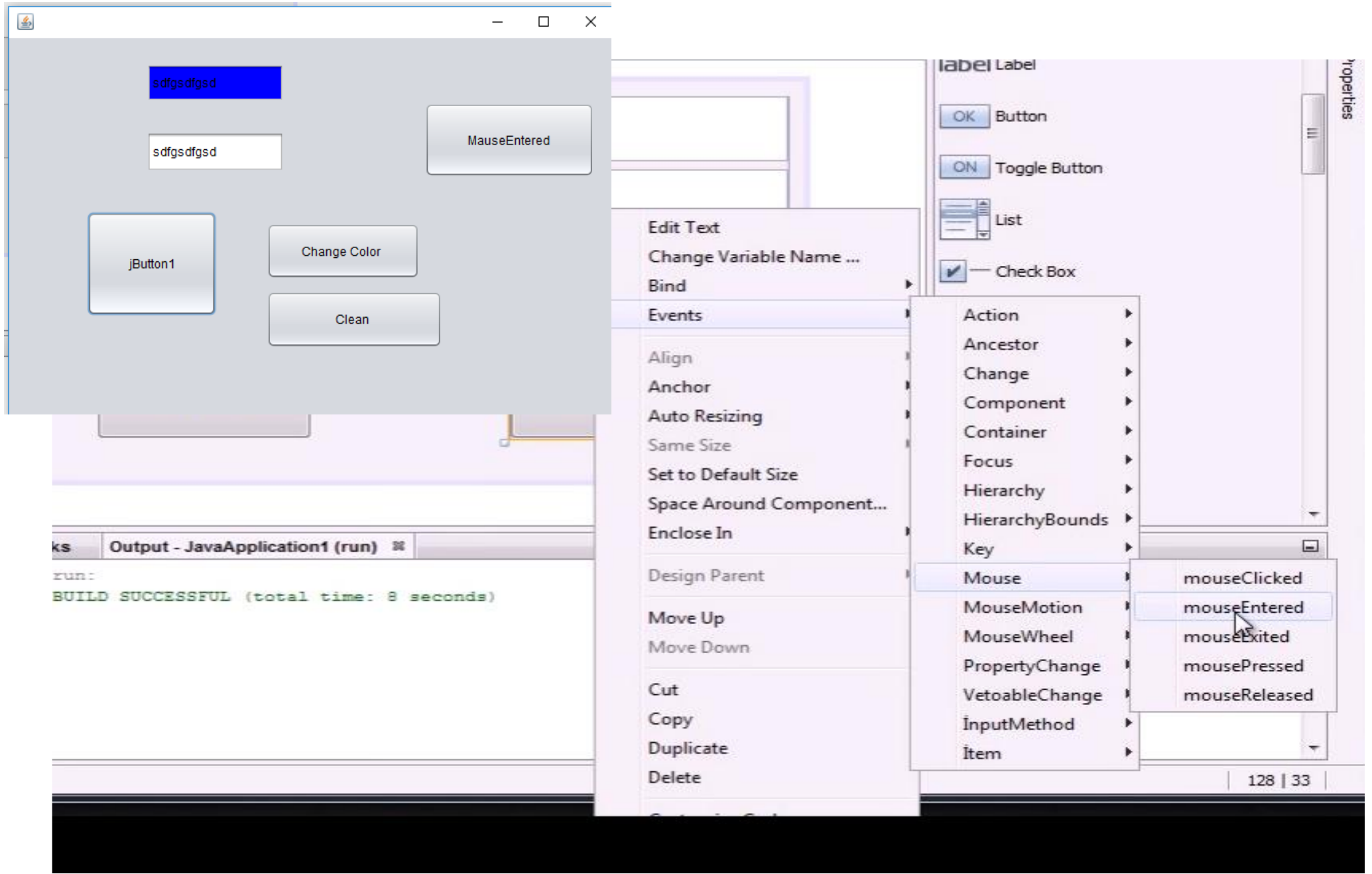




Design History

```
    private void jTextField2ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
    }

    private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {

        jTextField1.setBackground(Color.red);

        // TODO add your handling code here:
    }

    private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {

        jTextField1.setText(null);
        jTextField2.setText(null);

// TODO add your handling code here:
    }
```

Mouse Görevleri



```
    private void jButton4MouseEntered(java.awt.event.MouseEvent evt) {

        jTextField1.setBackground(Color.yellow);

        // TODO add your handling code here:
    }

    private void jButton4MouseExited(java.awt.event.MouseEvent evt) {

jTextField1.setBackground(Color.black);
 jTextField1.setBackground(Color.white);

// TODO add your handling code here:
    }
```