

Лабораторно упражнение 8 по ПИК 3

Визуално приложение със Swing и Netbeans

В това упражнение ще се запознаете с разработката на визуални приложения посредством java технологията SWING и средата за програмиране NetBeans IDE.

Ще бъде разработено приложение за конвертиране на температурна стойност от Целзий във Фаренхайт.

Работата е разделена в пет подзадачи, всяка от които има своите стъпки:

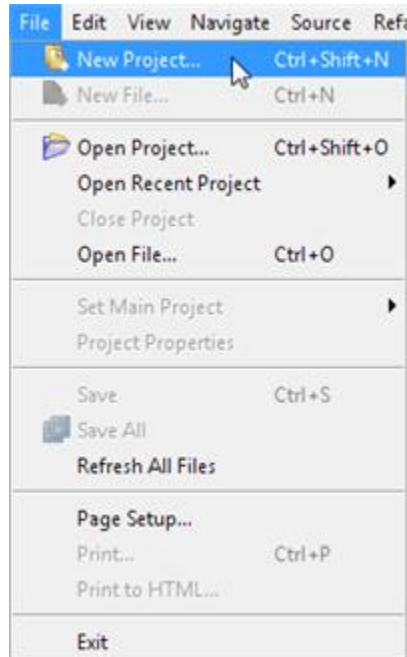
Създаване на проекта CelsiusConverter.....	2
Step 1: Създаване на нов проект	2
Step 2: Избор на вид проект	2
Step 3: Задаване на име на проекта	3
Step 4: Добавяне на JFrame форма.....	4
Step 5: Именуване на GUI класа	4
Основни полета на средата NetBeans IDE	5
Палитрата (Palette)	5
Поле за дизайн (Design Area)	5
Прозорец на свойствата (Property Editor)	6
Inspector.....	7
Създаване на графичния интерфейс (GUI) на CelsiusConverter	7
Step 1: Задаване на заглавие.....	8
Step 2: Добавяне на текстово поле (JTextField).....	8
Step 3: Добавяне на етикет (JLabel).....	9
Step 4: Добавяне на бутон (JButton)	10
Step 5: Добавяне на втори етикет (JLabel).....	10
Настройка на графичния интерфейс на CelsiusConverter	11
Step 1: Настройка на текста на компонентите.....	11
Step 2: Настройка на размера на компонентите	11
Step 3: Елиминиране на излишното пространство	12
Добавяне на логиката на приложението	13
Step 1: Промяна на стандартните име на променливи	13
Step 2: Регистриране на слушатели за събития	14
Step 3: Добавяне на кода за преобразуване на температурата	15
Step 4: Стартиране на приложението	16

I. Създаване на проекта CelsiusConverter

Тук са описани настройките при създаването на проекта, които трябва да следвате точно.

Step 1: Създаване на нов проект

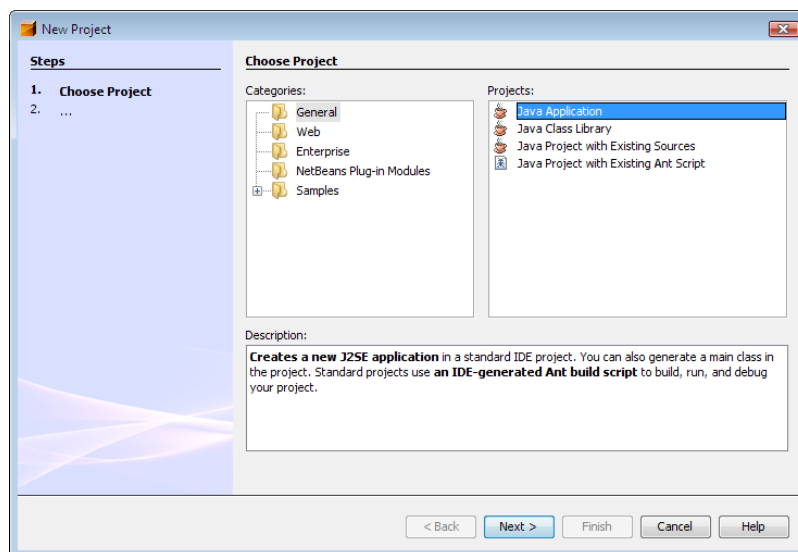
За да създадете нов проект, стартирайте NetBeans IDE и изберете New Project от менюто File:



Изгледът на NetBeans IDE може да се различава при платформите, но функционалността му остава същата.

Step 2: Избор на вид проект

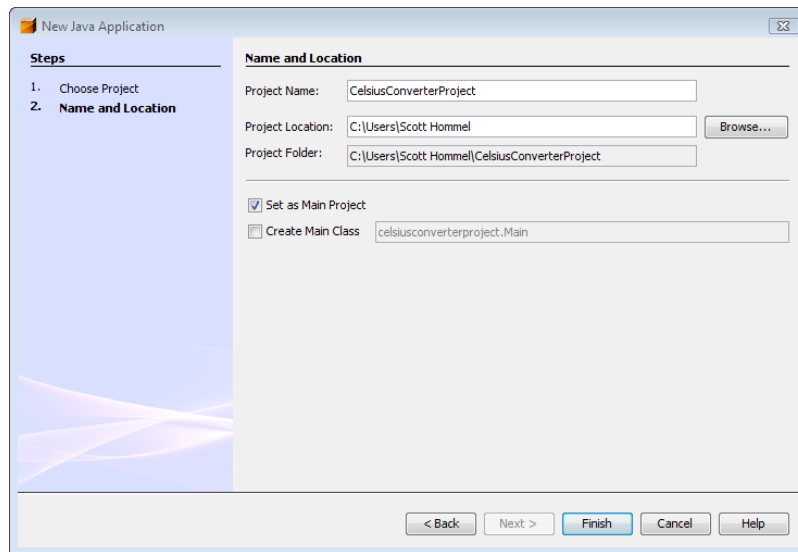
Изберете General -> Java Application



Може да забележите споменаването на "J2SE" в полето за описание; това е старото име на "Java SE" платформата. За да продължите натиснете бутона "Next".

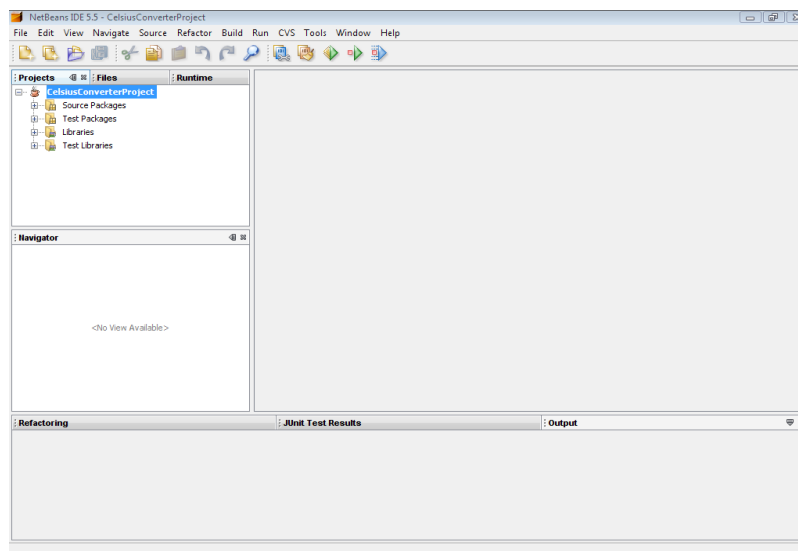
Step 3: Задаване на име на проекта

Въведете "CelsiusConverterProject" за име на проекта. За местоположение на проекта (полетата Project Location и Project Folder) изберете папка с ваше име в D:\Student\PIK3.



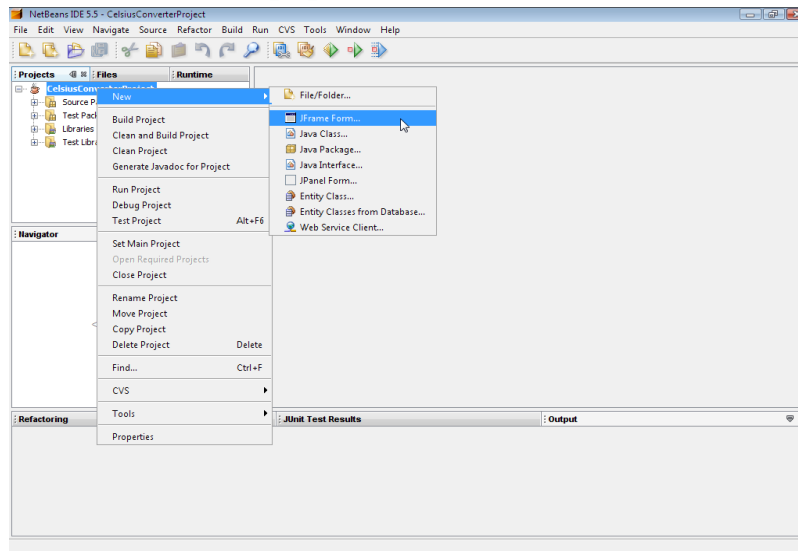
Обърнете внимание, че кутийката **"Create Main Class"** не трябва да е маркирана!

Ако оставим тази опция избрана ще се генерира нов клас с началната main функция на приложението, но нашият графичен прозорец (следващата стъпка) ще изпълнява тази задача. Натиснете бутона "Finish" когато приключите.



Когато средата (IDE) приключи зареждането, ще видите екран подобен на горния. Всички полета ще са празни с изключение на горе вляво, където е показан току що създаденият проект.

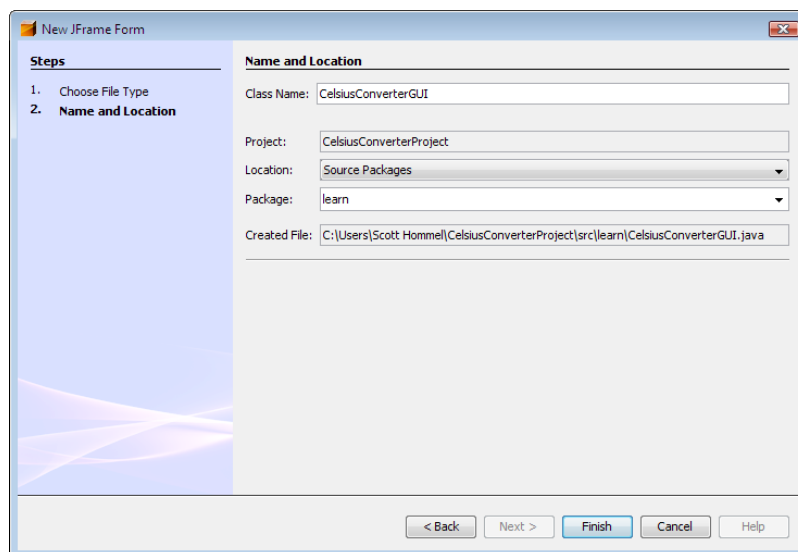
Step 4: Добавяне на JFrame форма



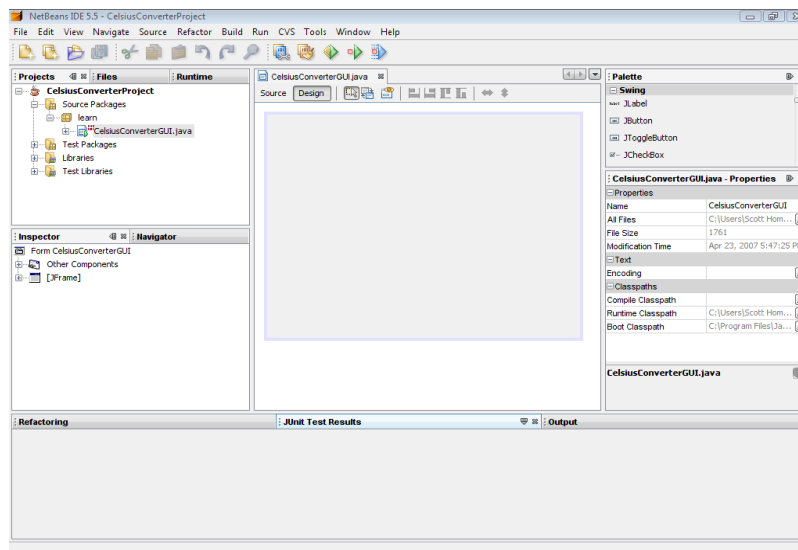
Натиснете десен бутон на мишката върху името CelsiusConverterProject на проекта и изберете New -> JFrame Form (JFrame е Swing класът отговарящ за основния прозорец на приложението).

Step 5: Именуване на GUI класа

В новоотвореният се прозорец напишете CelsiusConverterGUI като име на класа, а learn за име на пакета. По принцип можете да кръстите пакета както си искате, но в случая спазваме конвенцията на tutorиала с уроци.



Останалите полета ще бъдат автоматично попълнение както е показано горе. Натиснете бутона Finish.



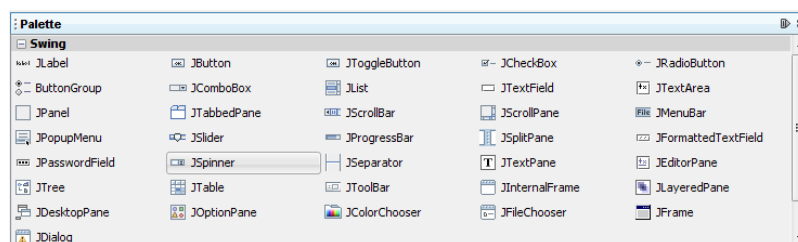
След като средата зареди, средното поле ще изобрази графичен изглед на CelsiusConverterGUI. В този екран (Display) могат да се изпълняват drag & drop и други манипулации с различните Swing компоненти.

II. Основни полета на средата NetBeans IDE

It is not necessary to learn every feature of the NetBeans IDE before exploring its GUI creation capabilities. In fact, the only features that you really need to understand are the *Palette*, the *Design Area*, the *Property Editor*, and the *Inspector*. We will discuss these features below.

Палитрата (Palette)

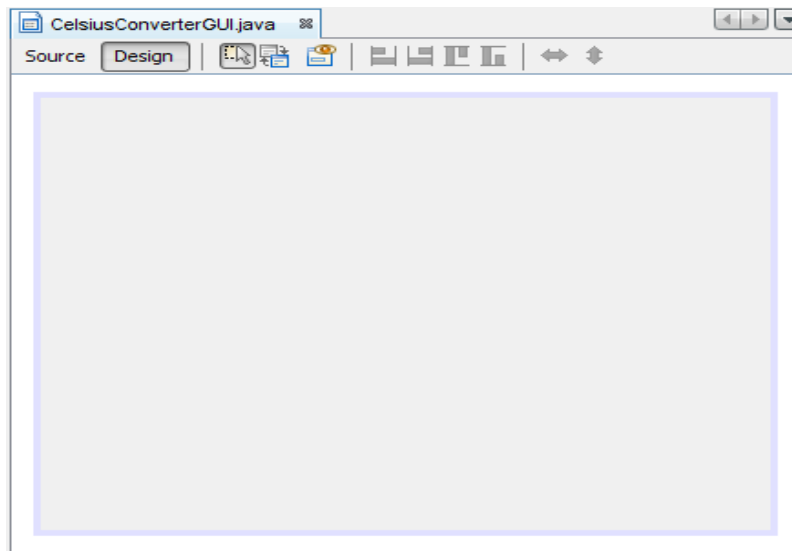
Палитрата съдържа всички компоненти предоставени от Swing API. По иконите и имената на много от тях може да се досетите за какво служат те. (JLabel е текстов етикет, JList списък от типа на падащо меню, и т.н.)



От този списък в приложението ще ползваме три от елементите: JLabel (стандартен текстов етикет), JTextField (текстово поле за въвеждане, в случая на температурата) и JButton (за конвертиране на температурата от Celsius към Fahrenheit.)

Поле за дизайн (Design Area)

В Design полето можете визуално да изградите вашия интерфейс. Той има два изгледа: *source view* и *design view*. Design изгледът е по подразбиране, както е показано по-долу. Може да превключвате между тях по всяко време като натискате на съответния им таб.



Фигурата отгоре показва един JFrame обект, представен като голям сив правоъгълник със син ръб. Често стандартното поведение (като например затваряне при натискане на „X“ бутона) за автоматично генерирани от средата и могат да се видят в изглед на кода в сини непроменими секции от код, познати като *guarded blocks*.

```

public class CelsiusConverterGUI extends javax.swing.JFrame {

    /** Creates new form CelsiusConverterGUI */
    public CelsiusConverterGUI() {
        initComponents();
    }

    /** This method is called from within the constructor to
     * initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is
     * always regenerated by the Form Editor.
     */
    // <editor-fold defaultstate="collapsed" desc="Generated Code ">
    private void initComponents() {

        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
        javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
        getContentPane().setLayout(layout);
        layout.setHorizontalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addGroup(
                    layout.createSequentialGroup()
                        .addGap(0, 400, Short.MAX_VALUE)
                )
        );
        layout.setVerticalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addGroup(
                    layout.createSequentialGroup()
                        .addGap(0, 300, Short.MAX_VALUE)
                )
        );
        pack();
    } // </editor-fold>

    /**
     * @param args the command line arguments
     */
    public static void main(String args[]) {

```

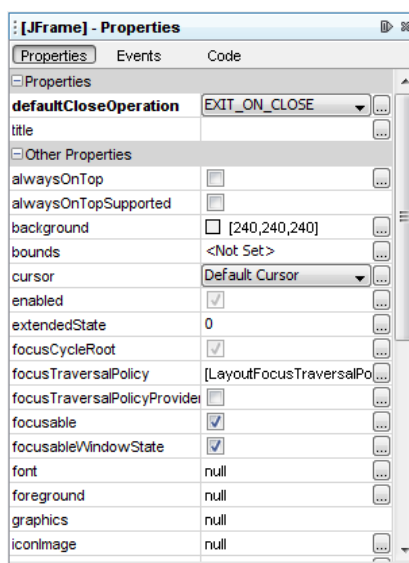
Бърз преглед на кода показва че IDE-то създава private метод с име `initComponents`, който инициализира всички компоненти на графичния интерфейс. Той също казва на приложението да се затваря ("exit on close"), изпълнява някои специфични задачи и след това пакетира компонентите (които скоро ще бъдат добавени) заедно на екрана.

В момента не е нужно да се разбира този код в детайли, разглеждаме го за да ни се изясни структурата на нещата. За повече информация за тези компоненти може да видите: [How to Make Frames \(Main Windows\)](#) и [Laying Out Components Within a Container](#).

Прозорец на свойствата (Property Editor)

Прозорецът на свойствата ни дава възможност да модифицираме свойствата на всеки един компонент. Той е интуитивен за ползване, в него има поредица от редове – всеки

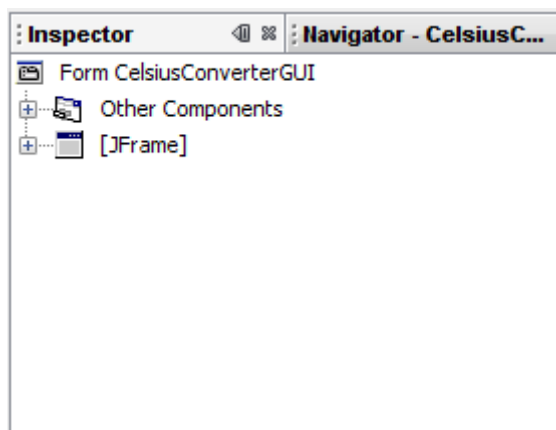
от които представя свойство. Всяко от свойствата може да се избира и променя без да се влиза директно в кода. Следващата фигура показва прозорецът на свойствата с току що зареден `JFrame` обект:



Фигурата показва множество свойства на този обект цвят на фона (`background color`), цвят на текста (`foreground color`), шрифт (`font`) и курсор на мишката (`cursor`).

Inspector

Последното поле от средата на NetBeans IDE, което ще използваме тук е инспекторът (Inspector):



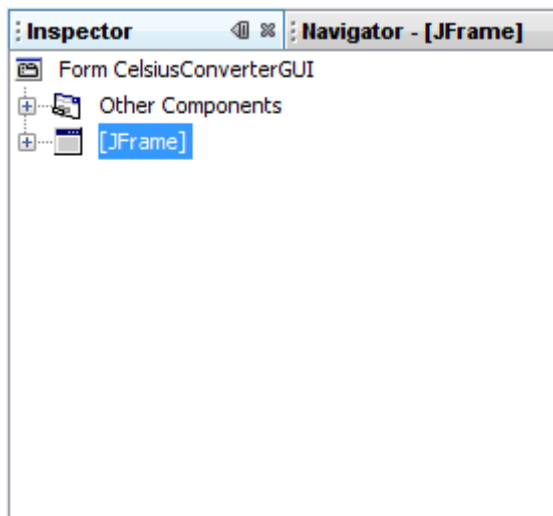
Инспекторът представлява графично представяне на компонентите във вашето приложение. Ние ще използваме инспекторът само веднъж за да променим имената на променливите, на нещо различно от това по подразбиране.

III. Създаване на графичния интерфейс (GUI) на CelsiusConverter

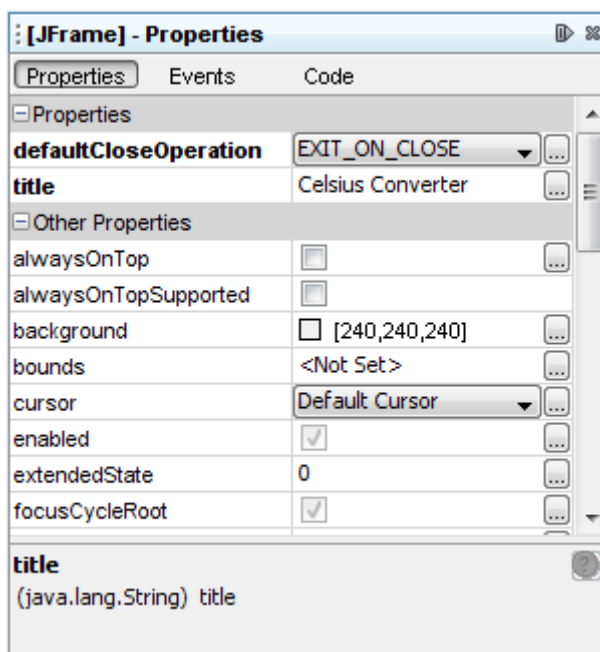
В тази точка ще обясним как се ползва средата на NetBeans IDE за създаването на графичния интерфейс на вашето приложение. Когато провлачвате компоненти от Палитрата до Полето за дизайн, IDE-то автоматично генерира необходимия сорс код.


Step 1: Задаване на заглавие

Първо нека зададем заглавие на наформата `JFrame` - "Celsius Converter", като изберете (маркирате) с единично натискане върху `JFrame` в инспектора:



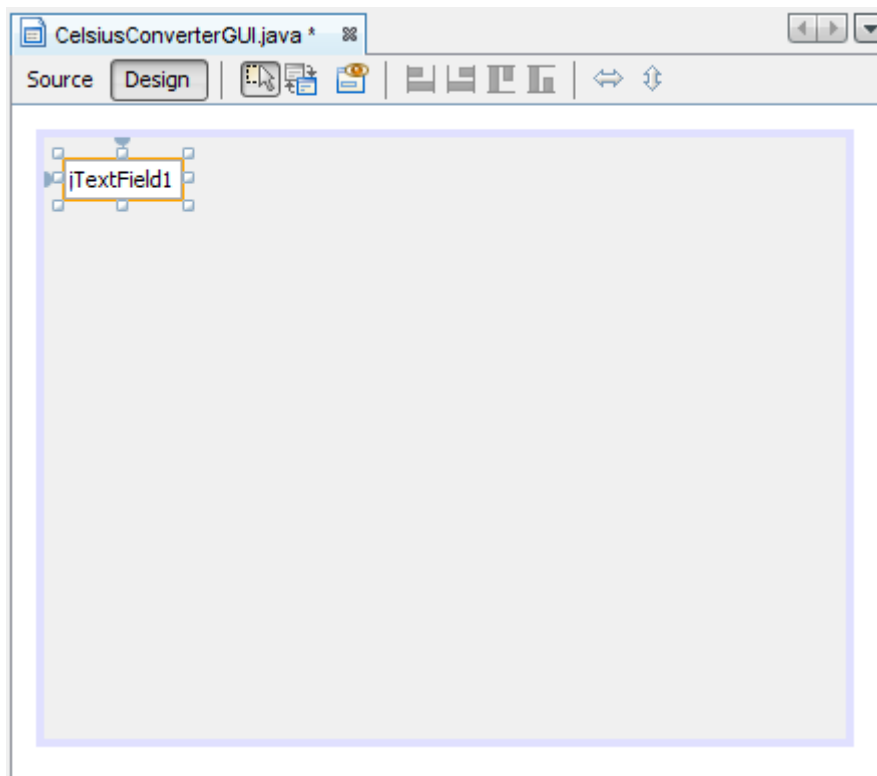
След това в прозореца Property Editor променете свойството `title` на посоченото горе:



Може да промените заглавието чрез двойно натискане и въвеждане на текста директно, или чрез натискане на бутона  и въвеждане на името в предоставеното ви поле. Възможно е да го направите като натиснете веднъж върху `JFrame` в инспектора и въведете текста директно без да ползвате прозореца на свойствата.

Step 2: Добавяне на текстово поле (JTextField)

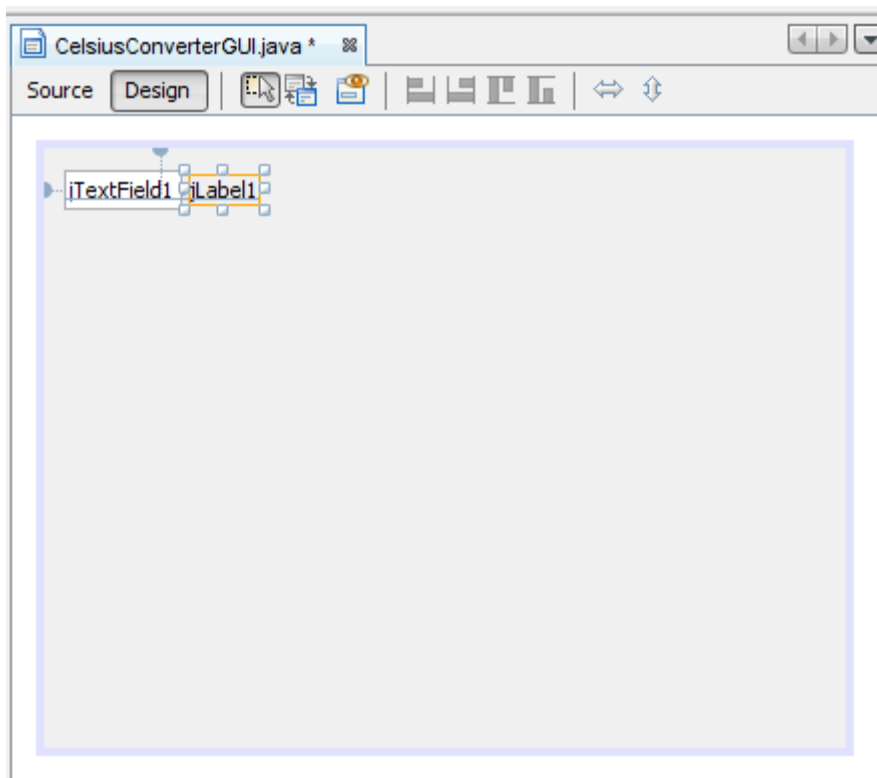
Провлачете `JTextField` от палитрата. Когато доближавате горния ляв ъгъл, средата ви предоставя визуални маркери (пунктирани линии), които подсказват подходящо разстояние. Използвайки линиите като помощ, пуснете `JTextField` в горния ляв ъгъл на прозореца, както е показано на следващата фигура:



Може да се изкушите да изтриете текста "JTextField1", но за сега го оставете на място. Ще го заменим по-нататък, когато правим настройките на всеки компонент. За повече информация за този компонент може да видите [How to Use Text Fields](#).

Step 3: Добавяне на етикет (JLabel)

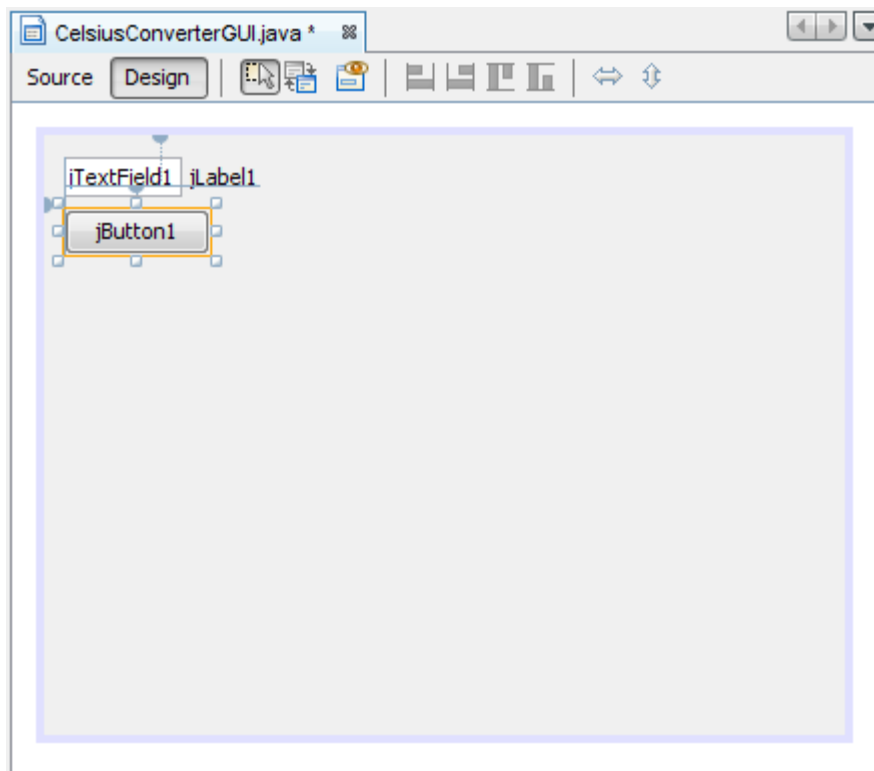
Провлачете JLabel в Design Area. Поставете го отдясно на JTextField. Подравнете долните части на двата елемента.



За повече информация: [How to Use Labels](#).

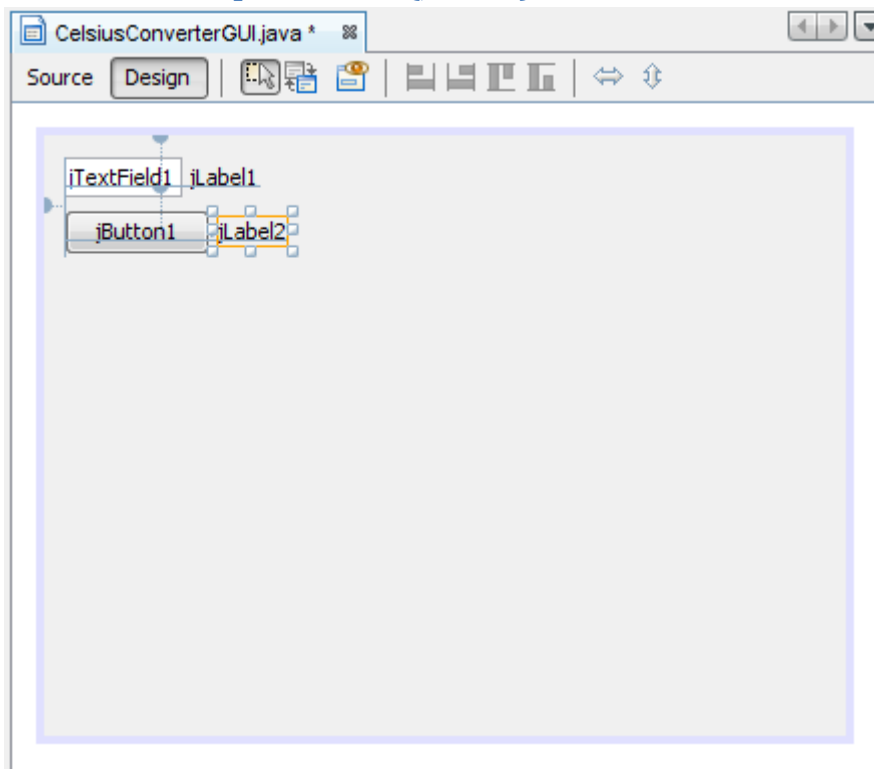
Step 4: Добавяне на бутон (JButton)

Добавете и един JButton от палитрата и го позиционирайте отдолу под JTextField.



За сега не променяйте ръчно големината на контролите. По-нататък в точките ще покажем как е по-правилно това да се прави. Информация за бутоните: [How to Use Buttons](#).

Step 5: Добавяне на втори етикет (JLabel)



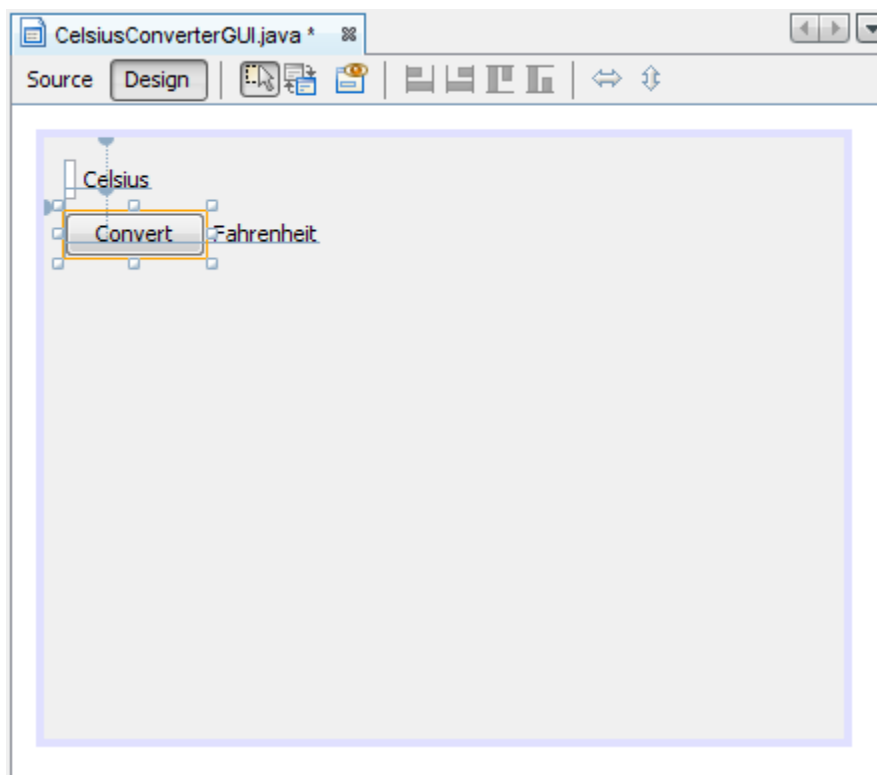
Накрая добавете и втори `JLabel`, повтаряйки стъпка 2. Поставете го отдясно на бутона `JButton`.

IV. Настройка на графичния интерфейс на `CelsiusConverter`

Когато всички компоненти са поставени можем да направим някои финални настройки. Има различни начини да се направи това, а по-долу е един от тях.

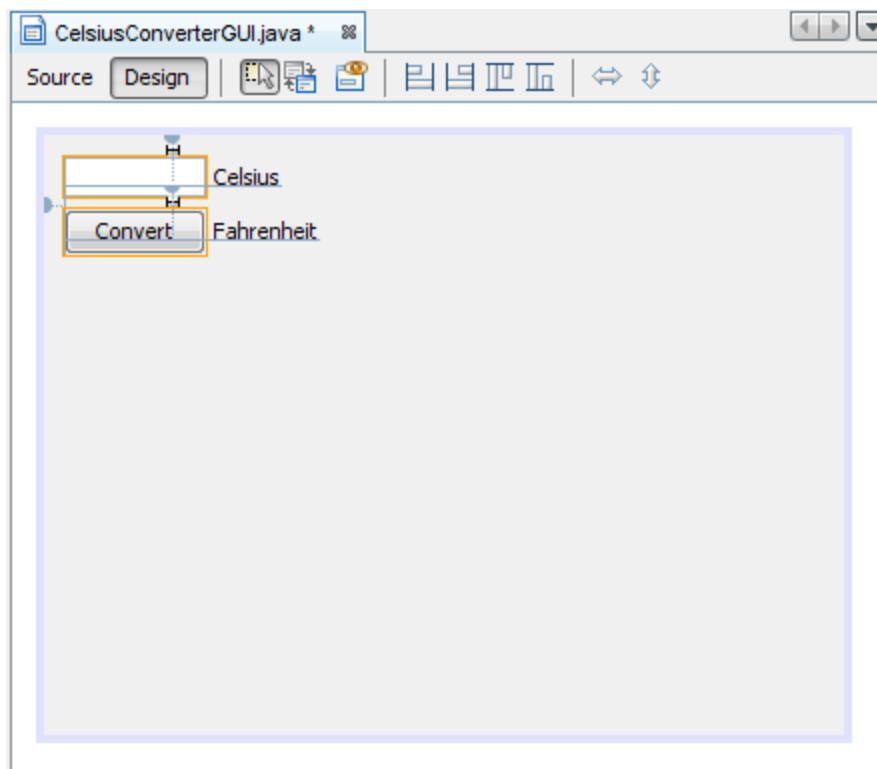
Step 1: Настройка на текста на компонентите

С двойно натискане върху `JTextField` и `JButton` може да смените стандартния им текст по подразбиране. Когато изтриете текста от `JTextField`, полето ще се свие до размера на фигурата по-долу. Може да промените текста на бутона `JButton` от "`JButton1`" на "Convert." Също променете текста на горния `JLabel` на "Celsius", а този на долния на "Fahrenheit".



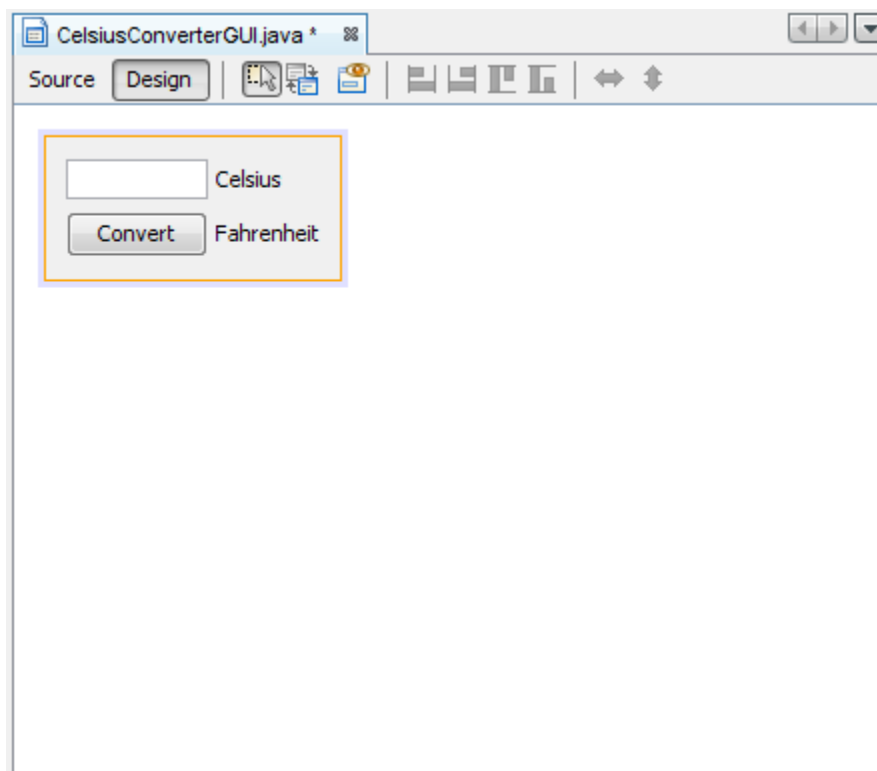
Step 2: Настройка на размера на компонентите

Изберете компонентите `JTextField` и `JButton` с `shift` и натискане на мишката. Това ще маркира и двата показвайки че са избрани. Натиснете с десен бутон и изберете `Same Size -> Same Width`. Компонентите сега ще са с еднаква ширина. Когато изпълнявате тази стъпка се уверете, че формата `JFrame` не е също селектирана. Ако е, `Same Size` менюто няма да е активно.



Step 3: Елиминирание на излишното пространство

Накрая хванете долния десен ъгъл на `JFrame` и променете големината ѝ, така че да премахнете излишното празно пространство. Имайте предвид, че ако намалите прекалено много формата, заглавието (което се вижда само runtime) може да не се вижда изцяло.



Графичната (GUI) част от това приложение е вече готова. Ако NetBeans IDE е свършило правилно своята работа, създаването на подобно GUI ви изглежда лесна задача, но отделете време да разгледате кода, който се е изгенерирал.

```
CelsiusConverterGUI.java
Source Design
private void initComponents() {
    jTextField1 = new javax.swing.JTextField();
    jLabel1 = new javax.swing.JLabel();
    jButton1 = new javax.swing.JButton();
    jLabel2 = new javax.swing.JLabel();

    setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

    jLabel1.setText("Celsius");

    jButton1.setText("Convert");

    jLabel2.setText("Fahrenheit");

    javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
    getContentPane().setLayout(layout);
    layout.setHorizontalGroup(
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addContainerGap()
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addComponent(jTextField1)
                .addComponent(jLabel1)
                .addComponent(jButton1)
                .addComponent(jLabel2))
            .addContainerGap(150, false))
    );
}
```

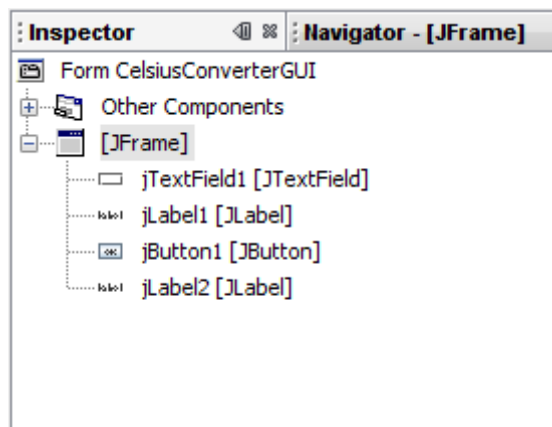
Можете да разширявате и скривате определени блокове от код (като например тяло на метод) като натискате символите + или – от лявата страна на кода.

V. Добавяне на логиката на приложението

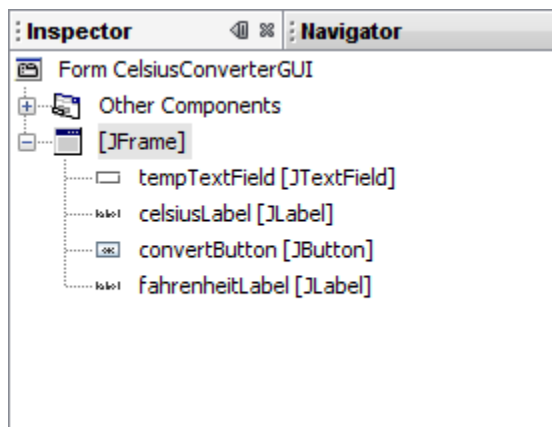
В тази стъпка ще добавим логиката на приложението, или това как то ще работи.

Step 1: Промяна на стандартните име на променливи

Фигурата отдолу показва стандартните имена по подразбиране както се появяват в инспектора. За всеки от компонентите първо е написано неговото име, а след това неговият тип в квадратни скоби. Например jTextField1 [JTextField] означава, че "jTextField1" името на обект от тип "JTextField".



Стандартните имена не отговарят на контекста на приложението, затова е удачно да се сменят от стойностите по подразбиране на нещо, което по-смислено. С натискане на десен бутон върху всяко от тях изберете "Change variable name." След промяната имената трябва да са следните:

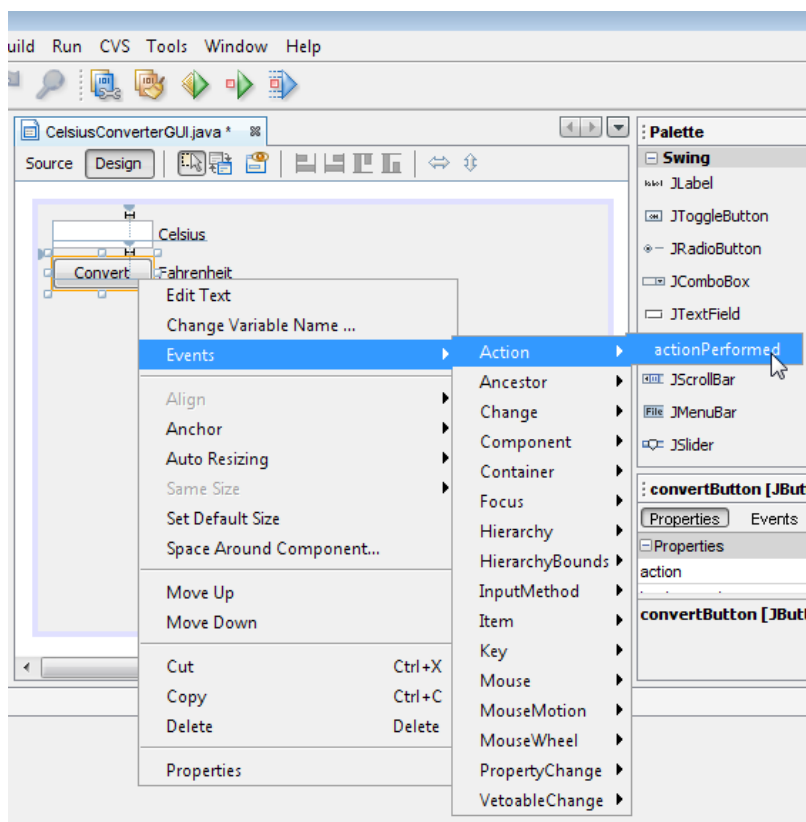


New Variable Names

Имената трябва да са "tempTextField", "celsiusLabel", "convertButton" и "fahrenheitLabel". Всяка промяна направена в инспектора ще се отрази автоматично в кода.

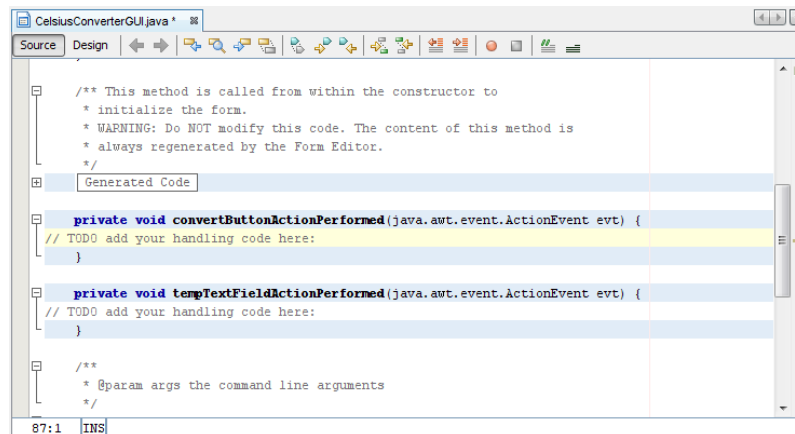
Step 2: Регистриране на слушатели за събития

Когато потребителя си взаимодейства със Swing GUI компонент (като например натискане на бутон), този компонент генерира специален вид обект, наречен обект – събитие (*event object*), който след това се разпространява до други предварително дефинирани „слушатели“ (*listeners*) за това събитие. В средата NetBeans IDE е лесно да се регистрират слушатели:



В полето за дизайн, натиснете върху бутона Convert за да го маркирате. Уверете се, че само той е маркиран (в противен случай тази стъпка няма да се получи). Натиснете десния бутон на мишката върху бутона и изберете от менюто Events -> Action ->

ActionPerformed. Това ще генерира необходимият код за обработка на събитието, оставяйки ви празен метод, в който трябва да напишете своята функционалност:



```
/** This method is called from within the constructor to
 * initialize the form.
 * WARNING: Do NOT modify this code. The content of this method is
 * always regenerated by the Form Editor.
 */
Generated Code
private void convertButtonActionPerformed(java.awt.event.ActionEvent evt) {
// TODO add your handling code here:
}
private void tempTextFieldActionPerformed(java.awt.event.ActionEvent evt) {
// TODO add your handling code here:
}
/**
 * @param args the command line arguments
 */
```

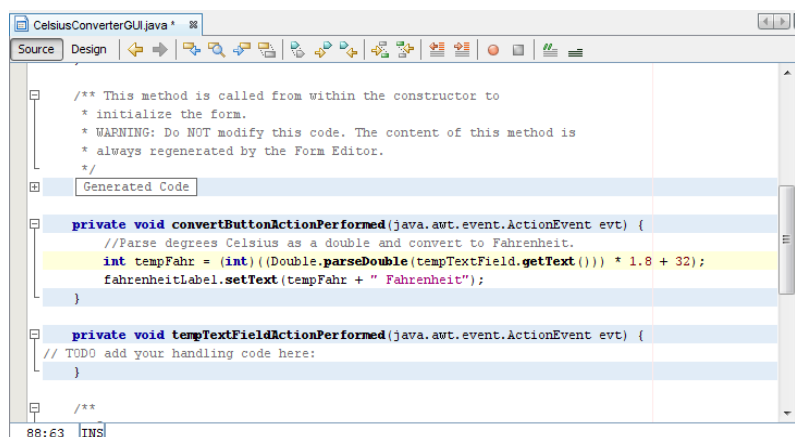
Има много различни видове събития представящи различни начини на действия, които потребителя може да направи (натискане на бутон на мишката, натискане на бутон на клавиатурата, мърдане на мишката и др.). За повече информация може да прочетете [Writing Event Listeners](#).

Step 3: Добавяне на кода за преобразуване на температурата

Последната стъпка трябва да поставите кода за преобразуване в тялото на метода.

```
//Parse degrees Celsius as a double and convert to Fahrenheit.
int tempFahr = (int)((Double.parseDouble(tempTextField.getText()))
    * 1.8 + 32);
fahrenheitLabel.setText(tempFahr + " Fahrenheit");
```

След като копирате кодът трябва да има следния вид:



```
/** This method is called from within the constructor to
 * initialize the form.
 * WARNING: Do NOT modify this code. The content of this method is
 * always regenerated by the Form Editor.
 */
Generated Code
private void convertButtonActionPerformed(java.awt.event.ActionEvent evt) {
//Parse degrees Celsius as a double and convert to Fahrenheit.
int tempFahr = (int)((Double.parseDouble(tempTextField.getText())) * 1.8 + 32);
fahrenheitLabel.setText(tempFahr + " Fahrenheit");
}
private void tempTextFieldActionPerformed(java.awt.event.ActionEvent evt) {
// TODO add your handling code here:
}
/**
```

Така вече приложението е завършено.

Важно: Този пример използва методът `parseDouble`, който не е локализиран (локални настройки за точка/запетая на дробните числа). Този пример е само за демонстрация. По-добре е да се използва [Scanner](#) класа за преобразуване на потребителския вход.

Step 4: Стартиране на приложението

Стартирайте с `Run -> Run Main Project` в средата NetBeans IDE.

Първият път, когато стартирате приложението ще бъдете попитани дали да настроите `CelsiusConverterGUI` като `main` клас на проекта. Натиснете `OK` бутона и след като програмата завърши компилирането ще видите програмата работеща в своя собствен прозорец.

Честито! Сега остават допълнителните задачи...