الجلسة السادسة - برمجة 3

الغاية من الجلسة: البرمجة الموجهة للأحداث Event Driven Programming

في برمجة Java الموجهة للأحداث (Event-Driven Programming) ، تتمثل الأساسيات في مفاهيم الـ Source (المصدر) والـ Event (الحدث) والـ Listener (المستمع):

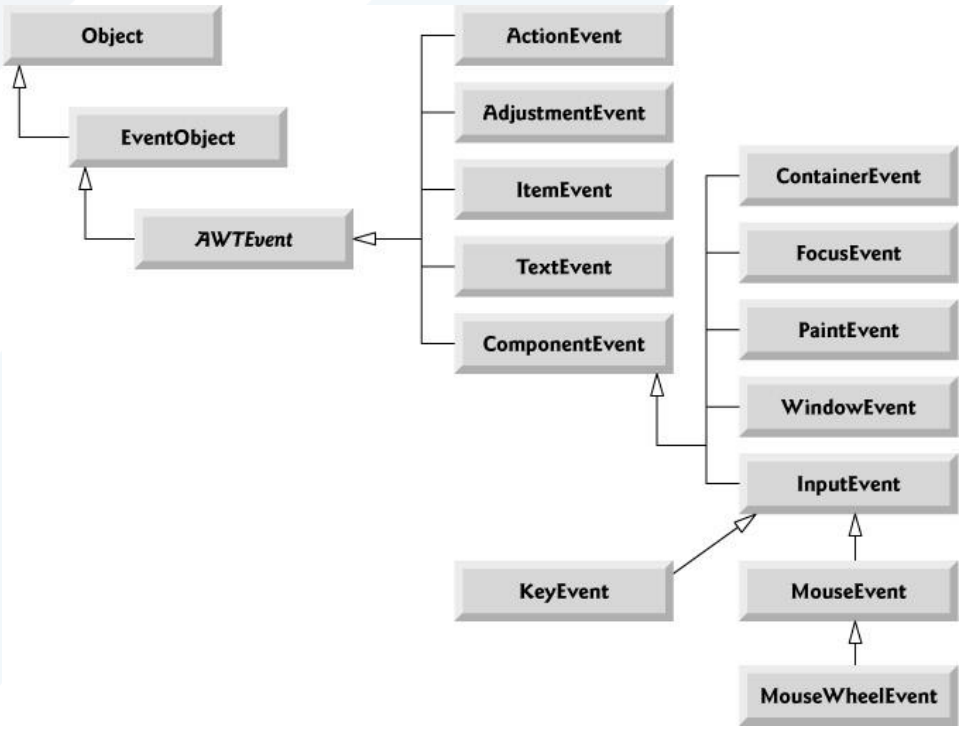
1. **الـ Source:**

- المصدر هو العنصر الذي يُثير الحدث عندما يحدث تفاعل معه، مثل زر الفأرة أو زر على واجهة المستخدم أو أي عنصر آخر يتفاعل مع المستخدم.
- عندما يحدث تفاعل مع المصدر، ينشئ الـ Source (المصدر) كائناً من نوع الحدث المناسب ويُعلن الحدث.

2. **الـ Event:**

- الحدث هو النتيجة التي تحدث نتيجة لتفاعل المصدر، مثل النقر على زر الفأرة أو الضغط على زر على واجهة المستخدم.
- في Java، الحدث هو كائن من نوع معين يمثل الحدث الذي وقع، مثل **ActionEvent** أو **MouseEvent**.
- يحتوي الحدث عادة على معلومات إضافية حول الحدث الذي وقع، مثل الموقع (في حالة الفأرة) أو النص الذي تم كتابته (في حالة حدث النص).

ولدينا أنواع للحدث مثل الـ Mouse Event ويتضمن أي تفاعل يقوم به المستخدم مع الفأرة مثل النقر والتحريك و .. الخ، وكذلك الـ Key Event تنتج من التفاعل مع الـ keyboard والـ Action Event للنقر على الزر، أنواع أخرى من الـ Events واضحة في الشكل الآتي:

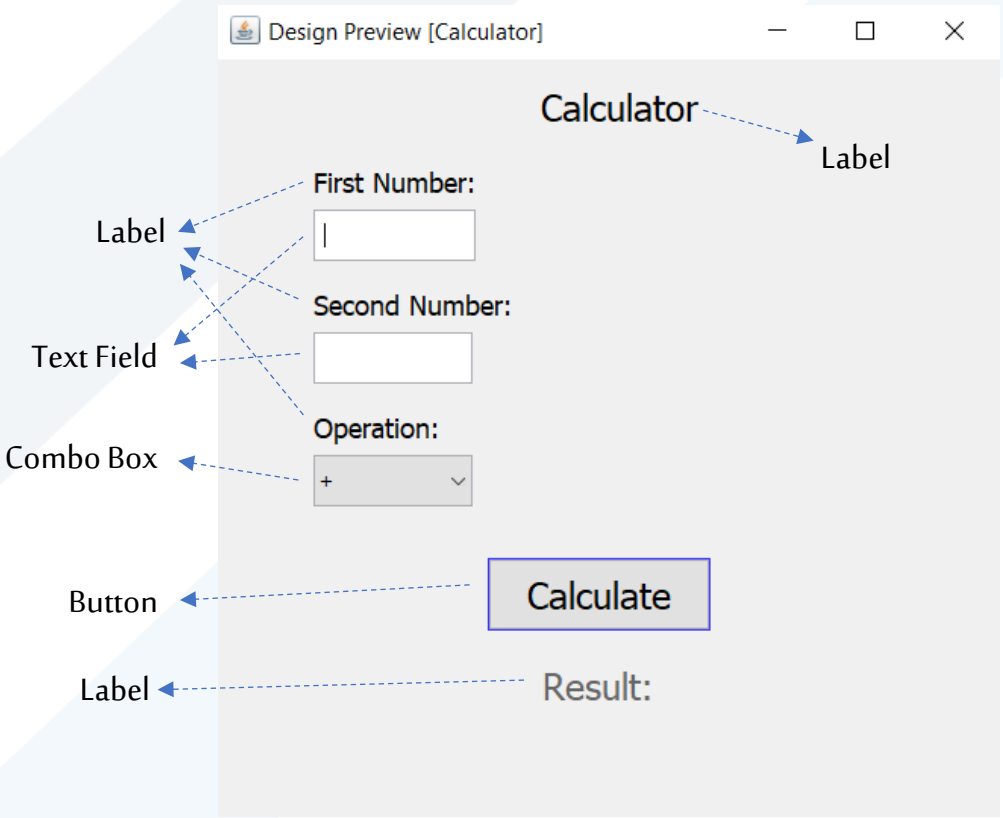


3. الـ Listener:

- المستمع هو الكائن الذي يتم تعليقه على المصدر للاستماع إلى الأحداث التي يثيرها.
- في Java، يحقق المستمع واجهة معينة تحتوي على طريقة (أو أكثر) يتم استدعائها عند حدوث الحدث المرتبط بالمستمع.
- عادةً ما يتم إنشاء مستمع مخصص لكل نوع من الأحداث التي يمكن أن يثيرها المصدر، على سبيل المثال **ActionListener** للاستماع إلى حدث النقر على زر.

هذه الآلية تمكنك من بناء تطبيقات تفاعلية في Java باستخدام الـ Java Swing وهي مكتبة غنية تمكّن المستخدمين من التفاعل مع العناصر على واجهة المستخدم ويستجيب التطبيق بناءً على هذه الإدخالات.

مثال: نريد بناء تطبيق بسيط لآلة حاسبة بحيث تحتوي على حقل لإدخال العدد الأول وحقل لإدخال العدد الثاني وقائمة منسدلة يختار المستخدم منها العملية التي يريدها، وزر calculate، وعبارة من أجل الخرج، الشكل الآتي يوضح شكل الآلة الحاسبة:



ننشئ Java project ثم ننقر بالزر اليميني على الـ package ونختار JFrame Form.

سوف نستخدم مجموعة من العناصر مثل الـ label والـ TextField والـ Button والـ ComboBox.

أين نكتب الكود الخاص بعملية الجمع؟

ننقر نقرتين على زر الـ Calculate فنذهب إلى الطريقة: private void jButton1ActionPerformed(java.awt.event.ActionEvent evt)

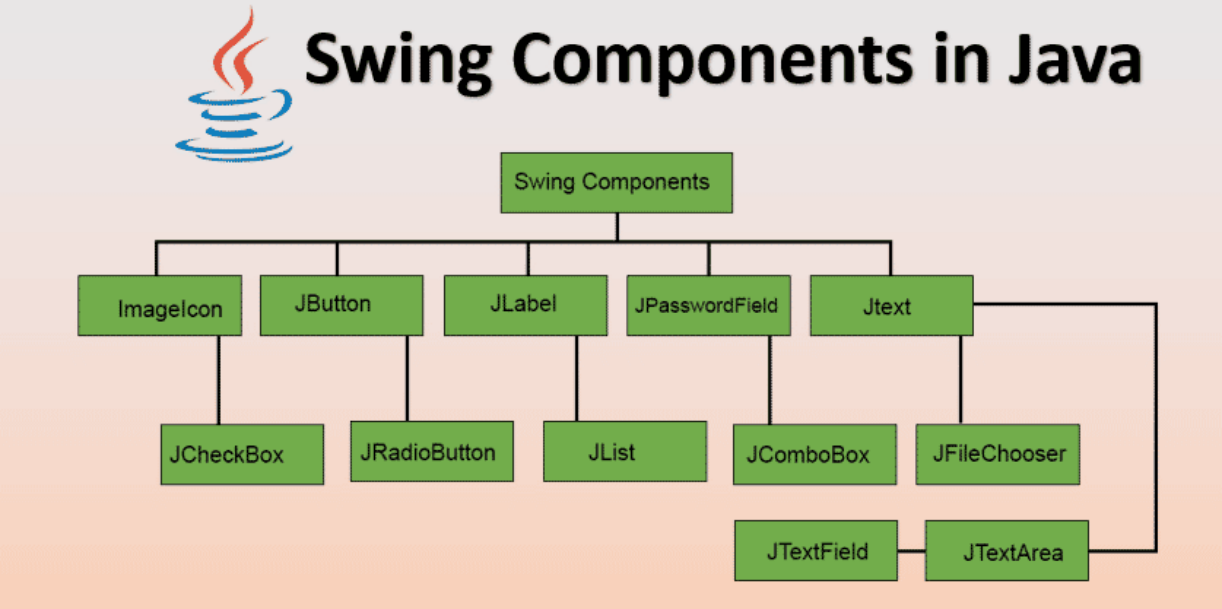
- الاسم **jButton1ActionPerformed** يعبر عن اسم الطريقة، وعادة ما يكون هو نفس اسم العنصر (component) الذي يثير الحدث (في هذه الحالة، الزر Calculate يُسمى jButton1) وبعد ذلك تأتي كلمة "ActionPerformed" لتشير إلى أن هذه الطريقة تتعامل مع حدث ينشأ عندما يتم تنفيذ إجراء (action) معين.

- الوسيط **(java.awt.event.ActionEvent evt)** يحتوي على معلومات حول الحدث الذي حدث مثل المصدر وما إلى ذلك.

هذه الطريقة هي الـ **Listener** لأنه ضمنها تتم معالجة الحدث الذي هو النقر على زر الـ Calculate.

لاحظ أن الوسيط الممرر للطريقة السابقة هو ActionEvent (الذي هو النقر على الزر).

فيما يلي مجموعة من العناصر في Java Swing:



العناصر السابقة هي الـ Sources والتي عندما نتفاعل معها تنقدح أنواع مختلفة من الأحداث Events والتي تنصت إليها الـ Listeners.

في البداية يجب أن نعطي اسم (متحول) لكل عنصر في النافذة من خلال النقر بالزر اليميني على العنصر ثم change variable name ونعطي كل منها اسم معين مثلاً نسمي الحقل الأول x والثاني y وقائمة العمليات الاسم op وعبارة النتيجة الاسم res، ثم نكتب في الطريقة السابقة الكود الآتي:

```
double xx = Double.parseDouble(x.getText());
double yy = Double.parseDouble(y.getText());

String selectedOperator = (String) op.getSelectedItem();
switch (selectedOperator) {
    case "+": res.setText("Result: "+ (xx+yy)); break;
    case "-": res.setText("Result: "+ (xx-yy)); break;
    case "*": res.setText("Result: "+ (xx*yy)); break;
    case "/": res.setText("Result: "+ (xx/yy)); break;
}
```

حيث أن xx هي من أجل تحويل القيمة في الحقل الأول (حقل إدخال العدد الأول) من String إلى double ونفس الأمر بالنسبة لـ yy.

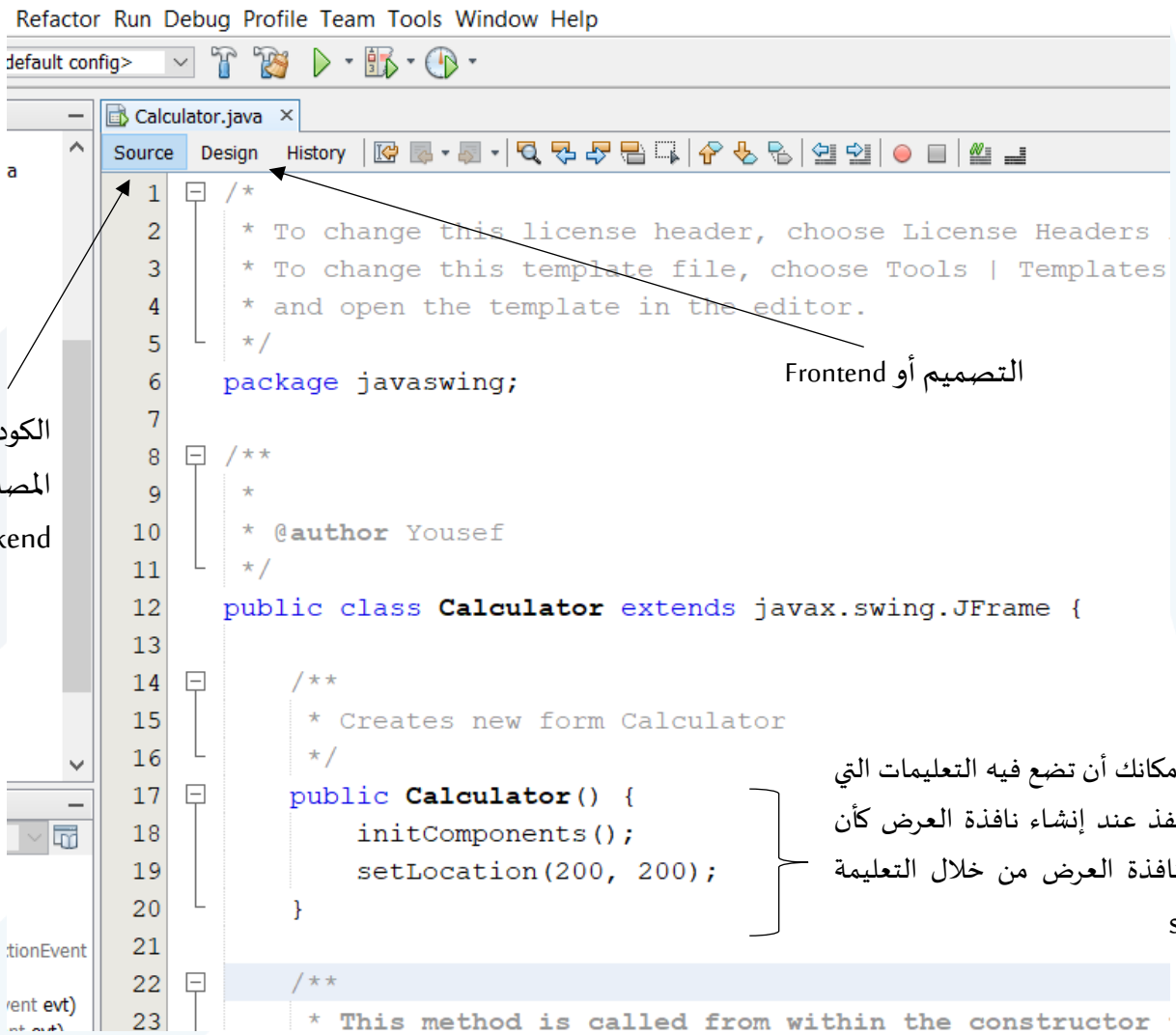
حيث أن parseDouble تحول الـ String المدخل (الوسيط) إلى double.

المتحول selectedOperator هو للحصول على العملية التي تم تحديدها (+ أو – أو * أو /) إلى String.

ثم في عبارة switch نختبر، فإذا كانت العملية هي "+" بالتالي أسند إلى متحول res (متحول الخرج) كلمة result متبوعة بجمع القيمتين المدخلتين xx+yy، ونفس الأمر بالنسبة لباقي العمليات.

من أجل اختبار البرنامج ننقر على run أو بالزر اليميني نختار run File ونختبر الآلة الحاسبة.

لا تنس: أنت تتعامل مع قسم يعبر عن الواجهة أو الـ Frontend وقسم يتعامل مع الكود في الكواليس أو Backend.



أيضاً من المهم أن تعرف أن جميع العناصر التي تعرفها في الواجهة من أزرار وحقول إدخال وقوائم وغيرها هي في الحقيقة أغراض من صفوف يتم تعريفها في الـ Backend بشكل تلقائي بمجرد أن تضعها في الواجهة Frontend:

```
210
211         // Variables declaration - do not modify
212         private javax.swing.JButton jButton1;
213         private javax.swing.JLabel jLabel1;
214         private javax.swing.JLabel jLabel2;
215         private javax.swing.JLabel jLabel3;
216         private javax.swing.JLabel jLabel4;
217         private javax.swing.JComboBox<String> op;
218         private javax.swing.JLabel res;
219         private javax.swing.JTextField x;
220         private javax.swing.JTextField y;
221         // End of variables declaration
222     }
```

أما باقي الطرق الموجودة في الـ Backend هي طرق معالجة الأحداث Listeners لمعالجة جميع التفاعلات مع العناصر التي وضعتها في الواجهة من أزرار وحقول إدخال وقوائم وغيرها، بالإضافة إلى طريقة الـ main التي يبدأ التنفيذ منها حيث يتم فيها أخذ غرض من الواجهة التي عرفتها التي هي الـ Calculator وجعلها مرئية setVisible(true):

```
/**
 * @param args the command line arguments
 */
public static void main(String args[]) {
    /* Set the Nimbus look and feel */
    Look and feel setting code (optional)

    /* Create and display the form */
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new Calculator().setVisible(true);
        }
    });
}
```

انتهت الجلسة