



# برمجة متقدمة ADVANCE PROGRAMMING IN C#

Java1 - المحاضرة العاشرة

علوم حاسوب وتقنية المعلومات - مستوى ثالث - ترم ثاني

أ/ نجم الدين الدغار [najmuddin.developer@gmail.com](mailto:najmuddin.developer@gmail.com)

## باستخدام لغة جافا

### 1- نموذج لأداء عملية الجمع بين رقمين يتم ادخالهما عن طريق حقول الادخال

#### النموذج



#### الكود

```
package com.mycompany.lab1; // تحديد الحزمة التي ينتمي إليها البرنامج

import javax.swing.*; // استيراد مكتبة Swing لإنشاء الواجهة الرسومية
import java.awt.*; // استيراد مكتبة AWT لتخطيط الواجهة
import java.awt.event.ActionEvent; // استيراد مكتبة لمعالجة الأحداث
import java.awt.event.ActionListener; // استيراد ActionListener للاستماع للأحداث

public class SumCalculator extends JFrame { // إنشاء JFrame تعريف كلاس يرث من نافذة التطبيق

    private JTextField textField1, textField2; // حقول إدخال للأرقام
    private JLabel resultLabel; // تسمية لعرض النتيجة

    public SumCalculator() { // الخاص بالكلاس (Constructor) المنشئ

        setTitle("جمع رقمين"); // تحديد عنوان النافذة
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE); // إغلاق التطبيق عند النقر على زر الإغلاق
        setSize(300, 200); // تحديد حجم النافذة بعرض 300 وارتفاع 200
        setLayout(new GridLayout(4, 2, 5, 10)); // تعيين التخطيط ليكون مع 4 صفوف و2 عمود GridLayout

        // مكونات الواجهة
        JLabel label1 = new JLabel("الرقم الأول:"); // تسمية لحقول الإدخال
        textField1 = new JTextField(); // حقل إدخال للرقم الأول
        JLabel label2 = new JLabel("الرقم الثاني:"); // تسمية لحقول الإدخال
        textField2 = new JTextField(); // حقل إدخال للرقم الثاني
        JButton sumButton = new JButton("جمع"); // زر لتنفيذ عملية الجمع
        resultLabel = new JLabel("النتيجة: "); // تسمية لعرض النتيجة

        // إضافة العناصر إلى النافذة
```

```

add(label1);
add(textField1);
add(label2);
add(textField2);
add(sumButton);
add(resultLabel);

// لاستدعاء الحدث عند النقر على الزر myaction إنشاء كائن من
myaction click = new myaction(textField1, textField2,
resultLabel); // تعريف الكائن الذي سينفذ الحدث
sumButton.addActionListener(click); // ربط حدث النقر بالزر مع الكائن click

/*
sumButton.addActionListener(new ActionListener() { // تعليق طريقة
ActionListener إضافة الحدث باستخدام
@Override
public void actionPerformed(ActionEvent e) {
    calculateSum(); // استدعاء دالة حساب المجموع
}
});
*/

// طريقة لإضافة الحدث بشكل مختصر Lambda
/*sumButton.addActionListener(e -> {
    calculateSum(); // استدعاء دالة حساب المجموع باستخدام لامبدا
});*/

setVisible(true); // هنا يتم عرض النموذج بمجرد تعريف كائن من هذا الكلاس
}

private void calculateSum() { // دالة لحساب المجموع
    try {
        // قراءة الأرقام من الحقول وتحويلها إلى أرقام عشرية
        double num1 =
Double.parseDouble(textField1.getText().trim());
        double num2 =
Double.parseDouble(textField2.getText().trim());

        // حساب المجموع
        double sum = num1 + num2;

        // عرض النتيجة في التسمية
        resultLabel.setText("النتيجة: " + sum);
    } catch (NumberFormatException ex) { // التعامل مع الأخطاء إذا تم إدخال قيمة
        // غير رقمية
        if (textField1.getText().trim().isEmpty() ||
!isNumeric(textField1.getText().trim())) { // التحقق من صحة الحقل الأول

```

```

        JOptionPane.showMessageDialog(this, "الرجاء إدخال رقم صحيح في "
            + "الحقل الأول", "خطأ", JOptionPane.ERROR_MESSAGE); // رسالة خطأ عند إدخال قيمة غير صحيحة
        textField1.requestFocus(); // التركيز على الحقل الأول
    } else if (textField2.getText().trim().isEmpty() ||
        !isNumeric(textField2.getText().trim())) { // التحقق من صحة الحقل الثاني
        JOptionPane.showMessageDialog(this, "الرجاء إدخال رقم صحيح في "
            + "الحقل الثاني", "خطأ", JOptionPane.ERROR_MESSAGE); // رسالة خطأ عند إدخال قيمة غير صحيحة
        textField2.requestFocus(); // التركيز على الحقل الثاني
    }
}

private boolean isNumeric(String str) { // دالة للتحقق إذا كانت النصوص أرقامًا
    try {
        Double.parseDouble(str); // محاولة تحويل النص إلى رقم
        return true; // إذا نجحت عملية التحويل
    } catch (NumberFormatException e) { // إذا فشلت عملية التحويل
        return false; // إرجاع false
    }
}

public static void main(String[] args) { // دالة main لتشغيل التطبيق
    System.out.println("kkkkk"); // طباعة نص للمراجعة
    new SumCalculator(); // إنشاء كائن من الكلاس
}

// كلاس لتنفيذ حدث الزر
class myaction implements ActionListener { // عند ActionListener كلاس ينفذ
    // النقر على الزر
    private JTextField textField1, textField2; // حقول الإدخال
    private JLabel resultLabel; // تسمية لعرض النتيجة

    public myaction(JTextField textField1, JTextField textField2,
        JLabel resultLabel) { // مُنشئ الكلاس
        this.textField1 = textField1;
        this.textField2 = textField2;
        this.resultLabel = resultLabel;
    }

    // الدالة التي يتم تنفيذها عند النقر على الزر
    @Override
    public void actionPerformed(ActionEvent e) {
        try {
            // قراءة الأرقام من الحقول
            double num1 = Double.parseDouble(textField1.getText());
            double num2 = Double.parseDouble(textField2.getText());

```



```

        // حساب المجموع
        double sum = num1 + num2;

        // عرض النتيجة في التسمية
        resultLabel.setText("النتيجة: " + sum);
    } catch (NumberFormatException ex) { // التعامل مع الأخطاء إذا تم إدخال قيمة
        // غير رقمية
        JOptionPane.showMessageDialog(null, "الرجاء إدخال أرقام صحيحة",
            "خطأ", JOptionPane.ERROR_MESSAGE); // عرض رسالة خطأ
    }
}
}

```

## تشغيل الكود

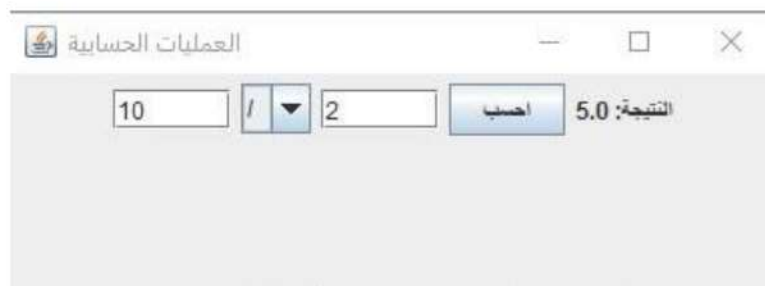
```

public class Lab1{
    public static void main(String[] args) { // لتشغيل التطبيق
        new SumCalculator(); // لتشغيل التطبيق إنشاء كائن من الكلاس
    }
}

```

## 2- نموذج لأداء العمليات الحسابية (+ - \* /) بين رقمين يتم ادخالهما عن طريق حقول الادخال بعد اختيار العملية من أداة [JComboBox]

### النموذج



### الكود

```

package com.mycompany.lab1to2;
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
public class CalculatorApp extends JFrame {
    private JTextField num1Field, num2Field; // حقول إدخال للأرقام
    private JComboBox<String> operations; // قائمة منسدلة لاختيار العملية الحسابية
    private JLabel resultLabel; // تسمية لعرض النتيجة

    public CalculatorApp() { // الخاص بالكلاس (Constructor) المنشئ
        setTitle("العمليات الحسابية"); // تحديد عنوان النافذة
        setSize(400, 150); // تحديد حجم النافذة بعرض 400 وارتفاع 150
    }
}

```

```

        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE); // إغلاق التطبيق عند
النقر على زر الإغلاق
        setLayout(new FlowLayout()); // تعيين التخطيط ليكون
العناصر بشكل تلقائي

        // مكونات الواجهة
        num1Field = new JTextField(5); // إنشاء حقل إدخال للأول مع عرض 5 خانات
        num2Field = new JTextField(5); // إنشاء حقل إدخال للثاني مع عرض 5 خانات
        String[] ops = { "+", "-", "*", "/" }; // مصفوفة تحتوي على العمليات الحسابية
        operations = new JComboBox<>(ops); // إنشاء قائمة منسدلة تحتوي على العمليات الحسابية
        JButton calcButton = new JButton("احسب"); // زر لتنفيذ العملية الحسابية
        resultLabel = new JLabel("النتيجة: "); // تسمية لعرض نتيجة العملية الحسابية

        // زر التنفيذ
        calcButton.addActionListener(e -> { // إضافة حدث عند النقر على الزر
            try {
                double num1 = Double.parseDouble(num1Field.getText());
                // double قراءة الرقم الأول وتحويله إلى
                double num2 = Double.parseDouble(num2Field.getText());
                // double قراءة الرقم الثاني وتحويله إلى
                String op = (String) operations.getSelectedItem(); // الحصول على العملية المحددة من القائمة المنسدلة
                double result = switch (op) { // تنفيذ العملية الحسابية باستخدام
                    switch
                        case "+" -> num1 + num2; // الجمع
                        case "-" -> num1 - num2; // الطرح
                        case "*" -> num1 * num2; // الضرب
                        case "/" -> (num2 != 0) ? num1 / num2 : Double.NaN;
                        // القسمة مع التأكد من عدم القسمة على الصفر
                        default -> 0; // قيمة افتراضية
                };
                resultLabel.setText("النتيجة: " + result); // عرض النتيجة في
            } catch (NumberFormatException ex) { // التعامل مع الأخطاء عند إدخال قيم
                JOptionPane.showMessageDialog(this, "أدخل أرقام صحيحة",
                    "خطأ", JOptionPane.ERROR_MESSAGE); // إظهار رسالة خطأ
            }
        });

        // إضافة المكونات إلى النافذة
        add(num1Field); // إضافة حقل إدخال الرقم الأول
        add(operations); // إضافة القائمة المنسدلة للعمليات
        add(num2Field); // إضافة حقل إدخال الرقم الثاني
        add(calcButton); // إضافة زر التنفيذ
        add(resultLabel); // إضافة تسمية النتيجة

```

```

        // setVisible(true); // جعل النافذة مرئية عند تشغيل التطبيق
    }}

```

تشغيل الكود

```

public class Lab1 {
    public static void main(String[] args) {
        CalculatorApp f=new CalculatorApp();
        f.setVisible(true);
    }
}

```

3- نموذج التعامل مع JList <= ListBox حيث ان عند الضغط على الزر يتم اخذ النص من حقل الادخال الى JList وتأخذ 5 ارقام عشوائية عند بدأ التنفيذ

النموذج



الكود

```

package com.mycompany.lab1to2; // تحديد الحزمة التي ينتمي إليها البرنامج

import javax.swing.*;
import java.awt.*;
import java.util.Random; // لإنشاء أرقام عشوائية Random استيراد مكتبة

public class ListBoxApp extends JFrame {
    private JTextField inputField;
    private DefaultListModel<String> listModel; // نموذج القائمة لتخزين العناصر
    private JList<String> listBox; // عنصر القائمة لعرض البيانات

    public ListBoxApp() { // الخاص بالكلاس (Constructor) المنشئ
        setTitle("إضافة إلى القائمة"); // تحديد عنوان النافذة
        setSize(300, 300); // تحديد حجم النافذة بعرض 300 وارتفاع 300
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE); // إغلاق التطبيق عند
        // النقر على زر الإغلاق
        setLayout(new FlowLayout()); // تعيين التخطيط ليكون
        // العناصر بشكل تلقائي
    }
}

```



```

inputField = new JTextField(10); // إنشاء حقل إدخال النص بعرض 10 خانات
JButton addButton = new JButton("إضافة"); // زر لإضافة النصوص إلى القائمة
ListModel = new DefaultListModel<>(); // إنشاء نموذج القائمة لتخزين البيانات

listBox = new JList<>(listModel); // إنشاء قائمة وإسناد النموذج إليها
listBox.setPreferredSize(new Dimension(150, 100)); // تحديد حجم
القائمة بدون إضافة شريط تمرير

addButton.addActionListener(e -> { // إضافة حدث عند النقر على زر الإضافة
    String text = inputField.getText(); // الحصول على النص من حقل
الإدخال

    if (!text.isEmpty()) { // التحقق من أن الإدخال ليس فارغًا
        listModel.addElement(text); // إضافة النص إلى القائمة
        inputField.setText(""); // مسح حقل الإدخال بعد الإضافة
    }
});

// إضافة 5 أرقام عشوائية عند بدء التشغيل
Random rand = new Random(); // إنشاء كائن لتوليد الأرقام العشوائية
for (int i = 0; i < 5; i++) { // تكرار العملية 5 مرات
    listModel.addElement(rand.nextInt(100) + ""); // توليد رقم عشوائي
بين 0 و 99 وإضافته إلى القائمة
}

add(inputField); // إضافة حقل الإدخال إلى النافذة
add(addButton); // إضافة زر الإضافة إلى النافذة
add(new JScrollPane(listBox)); // إضافة القائمة داخل شريط تمرير لجعلها قابلة
للتمرير عند الحاجة

//setVisible(true);
}
}

```

تشغيل الكود

```

public class Lab1 {
    public static void main(String[] args) {
        ListBoxApp fl=new ListBoxApp();

        fl.setVisible(true);
    }
}

```

#### 4- نموذج لا اختيار صورة ووضعها في JLabel

في Java Swing ، لا توجد أداة PictureBox مثل C# لذلك يتم استخدام JLabel لعرض الصور لأن: JLabel يدعم ImageIcon مباشرة.



طريقة وضع الصورة على JLabel

```
JLabel imageLabel = new JLabel(new ImageIcon("path/to/image.jpg"));
```

النموذج



الكود

```
package com.mycompany.lab1to2;

import javax.swing.*;
import java.awt.*;
import java.io.File; // استيراد مكتبة File للتعامل مع الملفات

public class ImageChooserApp extends JFrame {
    private JLabel imageLabel; // تعريف متغير لتسمية لعرض الصورة

    public ImageChooserApp() { // الخاص بالكلاس (Constructor) المنشئ
        setTitle("اختيار صورة"); // تحديد عنوان النافذة
        setSize(500, 400); // تحديد حجم النافذة بعرض 500 وارتفاع 400
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE); // إغلاق التطبيق عند
        // النقر على زر الإغلاق
        setLayout(new FlowLayout()); // تعيين التخطيط ليكون
        // لترتيب FlowLayout العناصر بشكل تلقائي

        JButton chooseButton = new JButton("اختر صورة"); // إنشاء زر لاختيار
        // الصورة
        imageLabel = new JLabel(); // إنشاء تسمية لعرض الصورة

        chooseButton.addActionListener(e -> { // إضافة حدث عند النقر على زر اختيار
            // الصورة
            JFileChooser fileChooser = new JFileChooser(); // إنشاء مربع
            // حوار لاختيار ملف
```

```

        int result = fileChooser.showOpenDialog(this); // إظهار مربع
        الحوار وحفظ النتيجة

        if (result == JFileChooser.APPROVE_OPTION) { // التحقق مما إذا كان
        المستخدم قد اختار ملفًا أم لا
            File selectedFile = fileChooser.getSelectedFile(); //
            الحصول على الملف المحدد
            ImageIcon imageIcon = new
            ImageIcon(selectedFile.getAbsolutePath()); // إنشاء أيقونة للصورة المختارة
            imageLabel.setIcon(new
            ImageIcon(imageIcon.getImage().getScaledInstance(300, 300,
            Image.SCALE_SMOOTH)));
            // بكسل مع الحفاظ على جودتها 300x300 ضبط الصورة داخل JLabel
        }
    });
    add(chooseButton); // إضافة زر اختيار الصورة إلى النافذة
    add(imageLabel); // إضافة التسمية لعرض الصورة إلى النافذة
}
}

```

### تشغيل الكود

```

public class Lab1 {
    public static void main(String[] args) {
        ImageChooserApp fimage=new ImageChooserApp();
        fimage.setVisible(true);
    }
}

```

### 5- انشاء زر يعمل على غلق النموذج الذي يوضع فيه

#### الكود

```

package com.mycompany.lab1to2;
import java.awt.FlowLayout;
import javax.swing.*;
import java.awt.event.ActionEvent;

class CloseButton extends JButton {
    public CloseButton(JFrame frame) {
        super("إغلاق النافذة");
        addActionListener((ActionEvent e) -> frame.dispose());
    }
}

```

## 6- إنشاء JTextField يقبل العربية أو الإنجليزية فقط

### الكود

```
package com.mycompany.lab1to2;
import java.awt.FlowLayout;
import javax.swing.*;
import java.awt.event.KeyAdapter;
import java.awt.event.KeyEvent;
class FilteredTextField extends JTextField {
    public FilteredTextField(boolean arabicOnly) {
        addKeyListener(new KeyAdapter() {
            @Override
            public void keyTyped(KeyEvent e) {
                char c = e.getKeyChar();
                if (arabicOnly && !(c >= 0x0600 && c <= 0x06FF)) {
                    e.consume();
                } else if (!arabicOnly && !Character.isLetter(c)) {
                    e.consume();
                }
            }
        });
    }
}
```

## 7- إنشاء زر [JButton] دائري

### الكود

```
package com.mycompany.lab1to2;
import javax.swing.*;
import java.awt.*;

class RoundButton extends JButton {
    public RoundButton(String text) {
        super(text);
        setContentAreaFilled(false);
    }

    @Override
    protected void paintComponent(Graphics g) {
        g.setColor(getBackground());
        g.fillOval(0, 0, getWidth(), getHeight());
        super.paintComponent(g);
    }
}
```



8- إنشاء أداة مخصصة مربع نص [LanguageSpecificTextField] يقبل فقط إدخال الحرف الانجليزية اوالعربية وذلك بعد اختيار اللغة من الخاصية المضافة له

#### الكود

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.KeyAdapter;
import java.awt.event.KeyEvent;

public class LanguageSpecificTextField extends JTextField {
    private boolean isArabic; // خاصية لتحديد اللغة (عربي أو إنجليزي)

    // مُنشئ الكلاس
    public LanguageSpecificTextField(int columns) {
        super(columns);
        this.isArabic = false; // اللغة الافتراضية هي الإنجليزية

        // إضافة مستمع للوحة المفاتيح للتحقق من الحروف المدخلة
        this.addKeyListener(new KeyAdapter() {
            @Override
            public void keyTyped(KeyEvent e) {
                char enteredChar = e.getKeyChar();
                if (isArabic) {
                    // إذا كانت اللغة عربية، نسمح فقط بالأحرف العربية
                    if (!Character.toString(enteredChar).matches("[\u0621-\u064A]+")) {
                        e.consume(); // منع إدخال الحرف غير العربي
                    }
                } else {
                    // إذا كانت اللغة إنجليزية، نسمح فقط بالأحرف الإنجليزية
                    if (!Character.toString(enteredChar).matches("[a-zA-Z]+")) {
                        e.consume(); // منع إدخال الحرف غير الإنجليزي
                    }
                }
            }
        });

        // طريقة لتغيير اللغة (عربي أو إنجليزي)
        public void setLanguage(boolean isArabic) {
            this.isArabic = isArabic;
        }
    }
}
```

ملاحظة/ التركيز على المكاتب التي يتم تضمينها.