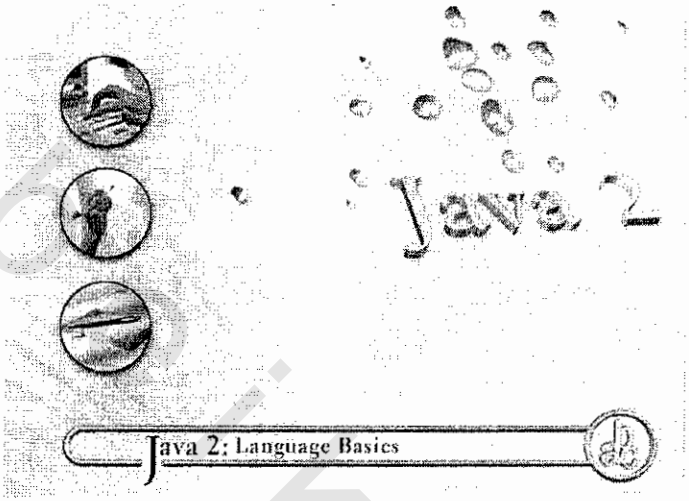
الفصل الثامن عشر



تطبيقات مبادئ اللغة

Language Fundamentals Applications

في هذا الفصل نتعرف علي العديد من التطبيقات الهامة للغة Java حيث نتعرف علي العديد من تطبيقات مبادئ اللغة.

obeikan.com

obeikan.com

مقدمة:

- تعلمنا في الفصول السابقة العديد من الموضوعات في لغة Java.
- ولكن لا يكفي لأي شخص مجرد تعلم موضوعات وأوامر اللغة ، حيث يحتاج دائماً لوجود العديد من التطبيقات حتي يمكنه تطبيق ما تعلم.
- في هذا الفصل نتعرف علي العديد من التطبيقات التي يمكن تنفيذها في لغة Java.
- حاول دائماً أن تقوم بحل التمرين قبل رؤية الحل ، ولا تقلق إذا وجدت صعوبة في البداية ، فذلك طبيعي عند تعلم لغة برمجة قوية مثل لغة Java.
- لاحظ أننا لن نقوم بشرح المثال ، لأن جميع الأمثلة في هذا الفصل تعتمد علي ما سبق لنا شرحه في الفصول السابقة ، ولكننا سنوضح فقط غرض المثال وكود البرمجة بالإضافة إلي نتيجة تنفيذ البرنامج.
- سوف نقوم بتقسيم التمرينات إلي الأقسام التالية:

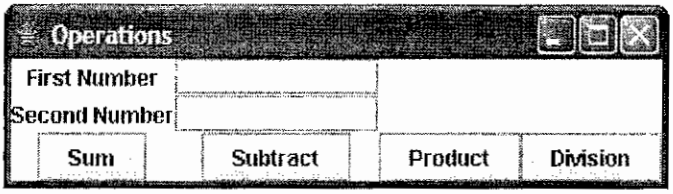
1. مشاريع مبادئ اللغة.
2. مشاريع البرمجة موجهة الهدف OOP.
3. مشاريع النصوص والتواريخ والمجموعات Collections.
4. مشاريع مكتبة Swing ومدير التخطيط Layout Manager ومعالجة الأحداث Event Handling.

- تم تقسيم هذه التمارين علي أربعة فصول ، وسنبدأ في هذا الفصل بمشاريع مبادئ اللغة.

مثال (1): العمليات الحسابية:

أولاً: هدف المثال:

اكتب برنامجاً يقوم بإدخال رقمين في أداة إطار JFrame كما هو واضح في الشكل التالي ثم يقوم بحساب مجموع وناتج الطرح والقسمة والضرب للرقمين وذلك عن طريق الضغط علي زر JButton لكل وظيفة من وظائف الجمع والطرح والقسمة والضرب. يجب أن تكون الرسائل الناتجة للمستخدم توضيحية وتدله علي نوعية الأرقام المعروضة عليه ، كما يجب أن يتعامل البرنامج مع احتمال أن يقوم المستخدم بإدخال حروف بدلاً من أرقام.



الشكل (18-1) تنفيذ البرنامج

ثانياً: كود البرمجة:

```
1:  import javax.swing.*;
2:  import java.awt.*;
3:  import java.awt.event.*;
4:
5:  public class MyOperations extends JFrame
6:  {
7:    JLabel labelNum1,labelNum2;
8:    JTextField textNum1,textNum2;
9:    JButton sum,subtract,product,division;
10:  GridBagLayout grid;
11:  GridBagConstraints gbc;
12:
13:  MyOperations()
14:  {
15:       addWindowListener(new WindowAdapter()
16:            {
17:                 public   void   windowClosing(WindowEvent
     we)
18:                 {
19:                      dispose();
20:                      System.exit(0);
21:                 }
22:            }
23:       );
24:
25:       Container c = getContentPane();
26:
```

```
27:     grid = new GridBagLayout();
28:     gbc = new GridBagConstraints();
29:     c.setLayout(grid);
30:
31:     labelNum1 = new JLabel("First Number");
32:     labelNum2 = new JLabel("Second Number");
33:
34:     textNum1 = new JTextField(10);
35:     textNum2 = new JTextField(10);
36:
37:     ButtonListener b = new ButtonListener();
38:
39:     sum = new JButton("Sum");
40:     sum.addActionListener(b);
41:
42:     subtract = new JButton("Subtract");
43:     subtract.addActionListener(b);
44:
45:     product = new JButton("Product");
46:     product.addActionListener(b);
47:
48:     division = new JButton("Division");
49:     division.addActionListener(b);
50:
51:     gbc.gridx = 1;
52:     gbc.gridy = 1;
53:     grid.setConstraints(labelNum1,gbc);
54:     c.add(labelNum1);
55:
56:     gbc.gridx = 2;
57:     gbc.gridy = 1;
58:     grid.setConstraints(textNum1,gbc);
59:     c.add(textNum1);
60:
61:     gbc.gridx = 1;
```

```
62:         gbc.gridy = 2;
63:         grid.setConstraints(labelNum2,gbc);
64:         c.add(labelNum2);
65:
66:         gbc.gridx = 2;
67:         gbc.gridy = 2;
68:         grid.setConstraints(textNum2,gbc);
69:         c.add(textNum2);
70:
71:         gbc.gridx = 1;
72:         gbc.gridy = 3;
73:         grid.setConstraints(sum,gbc);
74:         c.add(sum);
75:
76:         gbc.gridx = 2;
77:         gbc.gridy = 3;
78:         grid.setConstraints(subtract,gbc);
79:         c.add(subtract);
80:
81:         gbc.gridx = 3;
82:         gbc.gridy = 3;
83:         grid.setConstraints(product,gbc);
84:         c.add(product);
85:
86:         gbc.gridx = 4;
87:         gbc.gridy = 3;
88:         grid.setConstraints(division,gbc);
89:         c.add(division);
90:
91:         setTitle("Operations");
92:         pack();
93:         setVisible(true);
94:   }
95:
96:  public static void main(String args[])
```

obeikan.com

```
97:  {
98:      MyOperations myOperations = new MyOperations();
99:  }
100:
101:     class ButtonListener implements ActionListener
102:     {
103:         public void actionPerformed(ActionEvent e)
104:         {
105:             Object obj = e.getSource();
106:
107:             double num1 = 0;
108:
109:                 try
110:                 {
111:                     num1                                    =
    Double.parseDouble(textNum1.getText());
112:                 }
113:
114:                 catch(Exception ex)
115:                 {
116:
    JOptionPane.showMessageDialog(MyOperations.this,"Ple
    ase Enter a Number");
117:                     textNum1.selectAll();
118:                     textNum1.requestFocus();
119:                     return;
120:                 }
121:
122:                 double num2 = 0;
123:
124:                 try
125:                 {
126:                     num2                                    =
    Double.parseDouble(textNum2.getText());
127:                 }
```

```
128:
129:                    catch(Exception ex)
130:                    {
131:
    JOptionPane.showMessageDialog(MyOperations.this,"Ple
    ase Enter a Number");
132:                        textNum2.selectAll();
133:                        textNum2.requestFocus();
134:                        return;
135:                    }
136:
137:                if ( obj == sum )
138:                {
139:                    double result = num1 +  num2;
140:                    String message = "Sum of " + num1 + "
    and " + num2 + " = " + result;
141:
    JOptionPane.showMessageDialog(MyOperations.this,mess
    age);
142:                }
143:
144:                else if ( obj == subtract )
145:                {
146:                    double result = num1 -  num2;
147:                    String message = "Subraction of " + num1
    + " and " + num2 + " = " + result;
148:
    JOptionPane.showMessageDialog(MyOperations.this,mess
    age);
149:                }
150:
151:                else if ( obj == product )
152:                {
153:                    double result = num1 *  num2;
154:                    String message = "Product of " + num1 +
    " and " + num2 + " = " + result;
```

```
155:
    JOptionPane.showMessageDialog(MyOperations.this,mess
    age);
156:            }
157:
158:            else if ( obj == division )
159:            {
160:                double result = num1 /  num2;
161:                String message = "Division of " + num1 +
    " and " + num2 + " = " + result;
162:
    JOptionPane.showMessageDialog(MyOperations.this,mess
    age);
163:            }
164:        }
165:    }
166:    }
```

ملحوظات:

- تم استخدام الدالة pack() في هذا المثال وهي تستخدم حينما نريد ضبط عرض وطول نافذة البرنامج بحيث تحتوي جميع الأدوات فيها بالضبط ، وبالتالي توفر علينا هذه الدالة Method عملية اختيار أبعاد لنافذة البرنامج ودخول مرحلة تجارب لاختيار أنسب أبعاد.
- تم استخدام الفصيلة Double والدالة parseDouble() وهي تستخدم لتحويل النص إلي رقم.
- عند استخدام أداة صناديق الرسائل JOptionPane ، فتجد أننا استخدمنا المعامل Parameter التالي فيها MyOperations.this مثلما فعلنا في السطر رقم 116 ، وهذا الأمر في غاية الأهمية حيث أننا نريد أن يكون المعامل Parameter عبارة عن النافذة الأم لأداة صناديق الرسائل JOptionPane ، ولكن للأسف لا يمكننا استخدام this مباشرة وذلك بسبب وجودنا داخل الفصيلة ButtonListener ، وبالتالي فإن this ستقوم بالإشارة إلي الهدف Object الحالي والذي سيكون في

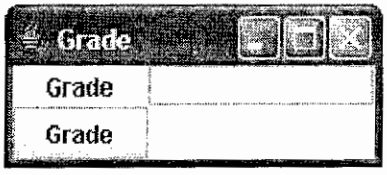
obeikan.com

هذه الحالة من نوع ButtonListener ، وبالتالي لا يتناسب معنا. لاحظ أن سبب هذه المشكلة أساساً هو أن الفصيلة ButtonListener تعتبر فصيلة داخلية Inner Class ، ولذلك توفر لنا لغة Java إمكانية الإشارة إلي الهدف object الحالي من الفصيلة الخارجية MyOperations عن طريق استخدام اسم الفصيلة MyOperations ويليها كلمة this ، أي أنه في حالة وجود فصائل داخلية Inner Classes ووجود حاجة إلينا أن نشير إلي الهدف object الحالي من الفصيلة الخارجية ، فإننا نستخدم اسم الفصيلة Class ويليها كلمة this.

مثال (2): حساب التقديرات:

أولاً: هدف المثال:

اكتب برنامجاً يقوم بإدخال درجة الطالب في أداة إطار JFrame كما هو واضح في الشكل التالي ثم يقوم بطباعة تقدير الطالب. يجب أن يقوم البرنامج بإظهار رسالة خطأ في حالة إدخال رقم أصغر من الصفر أو أكبر من 100. يكون تقدير الطالب "راسب" إذا كانت درجته أقل من 50 ، ويكون تقديره "مقبول" إذا كانت درجته أكبر من أو تساوي 50 وأقل من 65 ، ويكون تقديره "جيد" إذا كانت درجته أكبر من أو تساوي 65 وأقل من 75 ، ويكون تقديره "جيد جداً" إذا كانت درجته أكبر من أو تساوي 75 وأقل من 85 ، ويكون تقديره "ممتاز" إذا كانت درجته أكبر من أو تساوي 85 وأقل من أو تساوي 100 ، كما يجب أن يتعامل البرنامج مع احتمال أن يقوم المستخدم بإدخال حروف بدلاً من أرقام. إذا ضغط المستخدم علي زر الإدخال Enter فيجب أن يقوم البرنامج بطباعة تقدير الطالب كأن المستخدم قام بالضغط علي الزر Grade بالضبط.



الشكل (18-2) تنفيذ البرنامج

ثانياً: كود البرمجة:

```
1:  import javax.swing.*;
2:  import java.awt.*;
3:  import java.awt.event.*;
4:
5:  public class MyGrade extends JFrame
6:  {
7:    JLabel labelGrade;
8:    JTextField textGrade;
9:    JButton grade;
10:  GridBagLayout grid;
11:  GridBagConstraints gbc;
12:
13:  MyGrade()
14:  {
15:       addWindowListener(new WindowAdapter()
16:            {
17:                 public void windowClosing(WindowEvent
     we)
18:                 {
19:                      dispose();
20:                      System.exit(0);
21:                 }
22:            }
23:       );
24:
25:       Container c = getContentPane();
26:
27:       grid = new GridBagLayout();
28:       gbc = new GridBagConstraints();
29:       c.setLayout(grid);
30:
31:       labelGrade = new JLabel("Grade");
32:
33:       textGrade = new JTextField(10);
```

obeikan.com

```
34:
35:     textGrade.addKeyListener(new KeyAdapter()
36:         {
37:             public void keyReleased(KeyEvent e)
38:             {
39:                 char c = e.getKeyChar();
40:
41:                 if ( c == '\n' )
42:                 {
43:                     grade.doClick();
44:                 }
45:             }
46:         }
47:     );
48:
49:     ButtonListener b = new ButtonListener();
50:
51:     grade = new JButton("Grade");
52:     grade.addActionListener(b);
53:
54:     gbc.gridx = 1;
55:     gbc.gridy = 1;
56:     grid.setConstraints(labelGrade,gbc);
57:     c.add(labelGrade);
58:
59:     gbc.gridx = 2;
60:     gbc.gridy = 1;
61:     grid.setConstraints(textGrade,gbc);
62:     c.add(textGrade);
63:
64:     gbc.gridx = 1;
65:     gbc.gridy = 2;
66:     grid.setConstraints(grade,gbc);
67:     c.add(grade);
68:
```

obeikan.com

```
69:         setTitle("Grade");
70:         pack();
71:         setVisible(true);
72:     }
73:
74:     public static void main(String args[])
75:     {
76:         MyGrade myGrade = new MyGrade();
77:     }
78:
79:     class ButtonListener implements ActionListener
80:     {
81:         public void actionPerformed(ActionEvent e)
82:         {
83:             Object obj = e.getSource();
84:
85:             double mark = 0;
86:
87:                 try
88:                 {
89:                     mark                                    =
    Double.parseDouble(textGrade.getText());
90:                 }
91:
92:                 catch(Exception ex)
93:                 {
94:
    JOptionPane.showMessageDialog(MyGrade.this,"Pleas
    e Enter a Number");
95:                     textGrade.selectAll();
96:                     textGrade.requestFocus();
97:                     return;
98:                 }
99:
100:                if ( obj == grade )
101:                {
```


obeikan.com

```
102:                String studentGrade = "";
103:
104:                if ( mark < 0 )
105:                {
106:
    JOptionPane.showMessageDialog(MyGrade.this,"Pleas
    e enter a positive number");
107:                    return;
108:                }
109:
110:                if ( mark > 100 )
111:                {
112:
    JOptionPane.showMessageDialog(MyGrade.this,"Pleas
    e enter a number less than 100");
113:                    return;
114:                }
115:
116:                if ( mark < 50 )
117:                {
118:                    studentGrade = "Fail";
119:                }
120:
121:                else if ( mark < 65 )
122:                {
123:                    studentGrade = "Pass";
124:                }
125:
126:                else if ( mark < 75 )
127:                {
128:                    studentGrade = "Good";
129:                }
130:
131:                else if ( mark < 85 )
132:                {
133:                    studentGrade = "Very Good";
```

```
134:                }
135:
136:                else
137:                {
138:                    studentGrade = "Excellent";
139:                }
140:
141:                String message = "Grade is " +
    studentGrade;
142:
    JOptionPane.showMessageDialog(MyGrade.this,mess
    age);
143:            }
144:        }
145:    }
146:    }
```

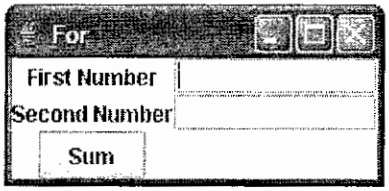
ملحوظات:

- تم استخدام الدالة doClick() في هذا المثال وهي تستخدم حينما نريد محاكاة الضغط علي الزر بلغة البرمجة ، بمعني أنه عندما نستدعي الدالة doClick() فذلك يكافئ أننا ضغطنا علي الزر ، وبالتالي يتم تنفيذ الأوامر الخاصة بالضغط علي هذا الزر ، وبهذا الشكل يتم توفير إعادة كتابة الكود لأننا نريد تنفيذ نفس وظيفة الزر عند الضغط علي زر الإدخال Enter.

مثال (3): مجموع الأرقام باستخدام for:

أولاً: هدف المثال:

اكتب برنامجاً يقوم بإدخال رقمين في أداة إطار JFrame كما هو واضح في الشكل التالي ثم يقوم بحساب مجموع الأرقام بداية من الرقم الأول حتي الرقم الثاني ، كما يجب أن يتعامل البرنامج مع احتمال أن يقوم المستخدم بإدخال حروف بدلاً من أرقام. إذا ضغط المستخدم علي زر الإدخال Enter فيجب أن يقوم البرنامج بطباعة مجموع الأرقام كأن المستخدم قام بالضغط علي الزر Sum بالضبط. قم بتنفيذ البرنامج باستخدام for.



الشكل (18-3) تنفيذ البرنامج

ثانياً: كود البرمجة:

```
1:  import javax.swing.*;
2:  import java.awt.*;
3:  import java.awt.event.*;
4:
5:  public class MyFor extends JFrame
6:  {
7:    JLabel labelNum1,labelNum2;
8:    JTextField textNum1,textNum2;
9:    JButton sum;
10:  GridBagLayout grid;
11:  GridBagConstraints gbc;
12:
13:  MyFor()
14:  {
15:       addWindowListener(new WindowAdapter()
16:            {
17:                 public void windowClosing(WindowEvent
     we)
18:                 {
19:                      dispose();
20:                      System.exit(0);
21:                 }
22:            }
23:       );
24:
25:       Container c = getContentPane();
26:
```

```
27:     grid = new GridBagLayout();
28:     gbc = new GridBagConstraints();
29:     c.setLayout(grid);
30:
31:     labelNum1 = new JLabel("First Number");
32:     labelNum2 = new JLabel("Second Number");
33:
34:     textNum1 = new JTextField(10);
35:     textNum2 = new JTextField(10);
36:
37:     MyKeyListener k = new MyKeyListener();
38:     textNum1.addKeyListener(k);
39:     textNum2.addKeyListener(k);
40:
41:     ButtonListener b = new ButtonListener();
42:
43:     sum = new JButton("Sum");
44:     sum.addActionListener(b);
45:
46:     gbc.gridx = 1;
47:     gbc.gridy = 1;
48:     grid.setConstraints(labelNum1,gbc);
49:     c.add(labelNum1);
50:
51:     gbc.gridx = 2;
52:     gbc.gridy = 1;
53:     grid.setConstraints(textNum1,gbc);
54:     c.add(textNum1);
55:
56:     gbc.gridx = 1;
57:     gbc.gridy = 2;
58:     grid.setConstraints(labelNum2,gbc);
59:     c.add(labelNum2);
60:
```

obeikan.com

```
61:         gbc.gridx = 2;
62:         gbc.gridy = 2;
63:         grid.setConstraints(textNum2,gbc);
64:         c.add(textNum2);
65:
66:         gbc.gridx = 1;
67:         gbc.gridy = 3;
68:         grid.setConstraints(sum,gbc);
69:         c.add(sum);
70:
71:         setTitle("For");
72:         pack();
73:         setVisible(true);
74:  }
75:
76:  public static void main(String args[])
77:  {
78:         MyFor myFor = new MyFor();
79:  }
80:
81:  class ButtonListener implements ActionListener
82:  {
83:         public void actionPerformed(ActionEvent e)
84:         {
85:                Object obj = e.getSource();
86:
87:                double num1 = 0;
88:
89:                       try
90:                       {
91:                              num1                                    =
    Double.parseDouble(textNum1.getText());
92:                       }
93:
94:                       catch(Exception ex)
```

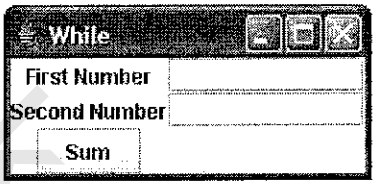
```
95:             {
96:
    JOptionPane.showMessageDialog(MyFor.this,"Please
  Enter a Number");
97:                 textNum1.selectAll();
98:                 textNum1.requestFocus();
99:                 return;
100:                }
101:
102:                double num2 = 0;
103:
104:                try
105:                {
106:                    num2                          =
  Double.parseDouble(textNum2.getText());
107:                }
108:
109:                catch(Exception ex)
110:                {
111:
    JOptionPane.showMessageDialog(MyFor.this,"Please
  Enter a Number");
112:                    textNum2.selectAll();
113:                    textNum2.requestFocus();
114:                    return;
115:                }
116:
117:                if ( num1 > num2 )
118:                {
119:
    JOptionPane.showMessageDialog(MyFor.this,"First
  number must be less than or equal second number");
120:                    textNum1.selectAll();
121:                    textNum1.requestFocus();
122:                    return;
```

```
123:                }
124:
125:            if ( obj == sum )
126:            {
127:                double result = 0;
128:
129:                for ( double i = num1 ; i <= num2 ;
    i++ )
130:                {
131:                    result += i;
132:                }
133:
134:                String message = "Sum of numbers
    between " + num1 + " and " + num2 + " = " + result;
135:
     JOptionPane.showMessageDialog(MyFor.this,messag
    e);
136:            }
137:        }
138:    }
139:
140:    class MyKeyListener extends KeyAdapter
141:    {
142:        public void keyReleased(KeyEvent e)
143:        {
144:            char c = e.getKeyChar();
145:
146:            if ( c == '\n' )
147:            {
148:                sum.doClick();
149:            }
150:        }
151:    }
152:    }
```

مثال (4): مجموع الأرقام باستخدام while:

أولاً: هدف المثال:

اكتب برنامجاً يقوم بإدخال رقمين في أداة إطار JFrame كما هو واضح في الشكل التالي ثم يقوم بحساب مجموع الأرقام بداية من الرقم الأول حتي الرقم الثاني ، كما يجب أن يتعامل البرنامج مع احتمال أن يقوم المستخدم بإدخال حروف بدلاً من أرقام. إذا ضغط المستخدم علي زر الإدخال Enter فيجب أن يقوم البرنامج بطباعة مجموع الأرقام كأن المستخدم قام بالضغط علي الزر Sum بالضبط. قم بتنفيذ البرنامج باستخدام while.



الشكل (18-4) تنفيذ البرنامج

ثانياً: كود البرمجة:

```
1:  import javax.swing.*;
2:  import java.awt.*;
3:  import java.awt.event.*;
4:
5:  public class MyWhile extends JFrame
6:  {
7:    JLabel labelNum1,labelNum2;
8:    JTextField textNum1,textNum2;
9:    JButton sum;
10:  GridBagLayout grid;
11:  GridBagConstraints gbc;
12:
13:  MyWhile()
14:  {
15:       addWindowListener(new WindowAdapter()
16:            {
17:                 public void windowClosing(WindowEvent
     we)
```

```
18:                 {
19:                     dispose();
20:                     System.exit(0);
21:                 }
22:             }
23:         );
24:
25:         Container c = getContentPane();
26:
27:         grid = new GridBagLayout();
28:         gbc = new GridBagConstraints();
29:         c.setLayout(grid);
30:
31:         labelNum1 = new JLabel("First Number");
32:         labelNum2 = new JLabel("Second Number");
33:
34:         textNum1 = new JTextField(10);
35:         textNum2 = new JTextField(10);
36:
37:         MyKeyListener k = new MyKeyListener();
38:         textNum1.addKeyListener(k);
39:         textNum2.addKeyListener(k);
40:
41:         ButtonListener b = new ButtonListener();
42:
43:         sum = new JButton("Sum");
44:         sum.addActionListener(b);
45:
46:         gbc.gridx = 1;
47:         gbc.gridy = 1;
48:         grid.setConstraints(labelNum1,gbc);
49:         c.add(labelNum1);
50:
51:         gbc.gridx = 2;
52:         gbc.gridy = 1;
```

```
53:         grid.setConstraints(textNum1,gbc);
54:         c.add(textNum1);
55:
56:         gbc.gridx = 1;
57:         gbc.gridy = 2;
58:         grid.setConstraints(labelNum2,gbc);
59:         c.add(labelNum2);
60:
61:         gbc.gridx = 2;
62:         gbc.gridy = 2;
63:         grid.setConstraints(textNum2,gbc);
64:         c.add(textNum2);
65:
66:         gbc.gridx = 1;
67:         gbc.gridy = 3;
68:         grid.setConstraints(sum,gbc);
69:         c.add(sum);
70:
71:         setTitle("While");
72:         pack();
73:         setVisible(true);
74:     }
75:
76:     public static void main(String args[])
77:     {
78:         MyWhile myWhile = new MyWhile();
79:     }
80:
81:     class ButtonListener implements ActionListener
82:     {
83:         public void actionPerformed(ActionEvent e)
84:         {
85:             Object obj = e.getSource();
86:
87:             double num1 = 0;
```

```
88:
89:             try
90:             {
91:                 num1                                    =
    Double.parseDouble(textNum1.getText());
92:             }
93:
94:             catch(Exception ex)
95:             {
96:
     JOptionPane.showMessageDialog(MyWhile.this,"Plea
    se Enter a Number");
97:                 textNum1.selectAll();
98:                 textNum1.requestFocus();
99:                 return;
100:                }
101:
102:                double num2 = 0;
103:
104:                try
105:                {
106:                    num2                                =
    Double.parseDouble(textNum2.getText());
107:                }
108:
109:                catch(Exception ex)
110:                {
111:
     JOptionPane.showMessageDialog(MyWhile.this,"Plea
    se Enter a Number");
112:                    textNum2.selectAll();
113:                    textNum2.requestFocus();
114:                    return;
115:                }
116:
```

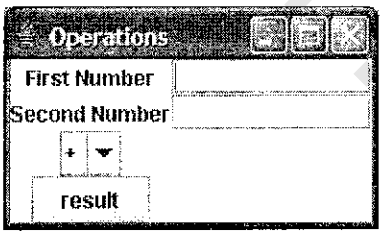
```
117:                    if ( num1 > num2 )
118:                    {
119:
    JOptionPane.showMessageDialog(MyWhile.this,"First
    number must be less than or equal second number");
120:                        textNum1.selectAll();
121:                        textNum1.requestFocus();
122:                        return;
123:                    }
124:
125:                if ( obj == sum )
126:                {
127:                    double result = 0;
128:
129:                    double start = num1;
130:
131:                    while ( start <= num2 )
132:                    {
133:                        result += start;
134:                        start++;
135:                    }
136:
137:                    String message = "Sum of numbers
    between " + num1 + " and " + num2 + " = " + result;
138:
    JOptionPane.showMessageDialog(MyWhile.this,messa
    ge);
139:                }
140:            }
141:        }
142:
143:        class MyKeyListener extends KeyAdapter
144:        {
145:            public void keyReleased(KeyEvent e)
146:            {
147:                char c = e.getKeyChar();
```

```
148:
149:            if ( c == '\n' )
150:            {
151:                sum.doClick();
152:            }
153:        }
154:    }
155:}
```

مثال (5): مجموع الأرقام باستخدام switch:

أولاً: هدف المثال:

اكتب برنامجاً يقوم بإدخال رقمين في أداة إطار JFrame كما هو واضح في الشكل التالي ثم يقوم بحساب مجموع وناتج الطرح والقسمة والضرب للرقمين وذلك عن طريق قائمة اختيار تحتوي علي العمليات المطلوبة ثم الضغط علي زر JButton واحد لجميع هذه الوظائف من وظائف الجمع والطرح والقسمة والضرب. يجب أن تكون الرسائل الناتجة للمستخدم توضيحية وتدله علي نوعية الأرقام المعروضة عليه ، كما يجب أن يتعامل البرنامج مع احتمال أن يقوم المستخدم بإدخال حروف بدلاً من أرقام.



الشكل (18-5) تنفيذ البرنامج

ثانياً: كود البرمجة:

```
1: import javax.swing.*;
2: import java.awt.*;
3: import java.awt.event.*;
4:
5: public class MyOperations extends JFrame
```

```
6:  {
7:    JLabel labelNum1,labelNum2;
8:    JTextField textNum1,textNum2;
9:    JComboBox operation;
10:  JButton result;
11:  GridBagLayout grid;
12:  GridBagConstraints gbc;
13:
14:  MyOperations()
15:  {
16:      addWindowListener(new WindowAdapter()
17:          {
18:              public void windowClosing(WindowEvent
     we)
19:              {
20:                  dispose();
21:                  System.exit(0);
22:              }
23:          }
24:      );
25:
26:      Container c = getContentPane();
27:
28:      grid = new GridBagLayout();
29:      gbc = new GridBagConstraints();
30:      c.setLayout(grid);
31:
32:      labelNum1 = new JLabel("First Number");
33:      labelNum2 = new JLabel("Second Number");
34:
35:      textNum1 = new JTextField(10);
36:      textNum2 = new JTextField(10);
37:
38:      MyKeyListener k = new MyKeyListener();
39:      textNum1.addKeyListener(k);
```

obeikan.com

```
40:     textNum2.addKeyListener(k);
41:
42:     String operationOptions[] = {"+","-","*","/"};
43:     operation = new JComboBox(operationOptions);
44:
45:     ButtonListener b = new ButtonListener();
46:
47:     result = new JButton("result");
48:     result.addActionListener(b);
49:
50:     gbc.gridx = 1;
51:     gbc.gridy = 1;
52:     grid.setConstraints(labelNum1,gbc);
53:     c.add(labelNum1);
54:
55:     gbc.gridx = 2;
56:     gbc.gridy = 1;
57:     grid.setConstraints(textNum1,gbc);
58:     c.add(textNum1);
59:
60:     gbc.gridx = 1;
61:     gbc.gridy = 2;
62:     grid.setConstraints(labelNum2,gbc);
63:     c.add(labelNum2);
64:
65:     gbc.gridx = 2;
66:     gbc.gridy = 2;
67:     grid.setConstraints(textNum2,gbc);
68:     c.add(textNum2);
69:
70:     gbc.gridx = 1;
71:     gbc.gridy = 3;
72:     grid.setConstraints(operation,gbc);
73:     c.add(operation);
74:
```

obeikan.com

```
75:       gbc.gridx = 1;
76:       gbc.gridy = 4;
77:       grid.setConstraints(result,gbc);
78:       c.add(result);
79:
80:       setTitle("Operations");
81:       pack();
82:       setVisible(true);
83:  }
84:
85:  public static void main(String args[])
86:  {
87:       MyOperations myOperations = new
     MyOperations();
88:  }
89:
90:  class ButtonListener implements ActionListener
91:  {
92:       public void actionPerformed(ActionEvent e)
93:       {
94:            Object obj = e.getSource();
95:
96:            double num1 = 0;
97:
98:                 try
99:                 {
100:                      num1 =
     Double.parseDouble(textNum1.getText());
101:                 }
102:
103:                 catch(Exception ex)
104:                 {
105:
      JOptionPane.showMessageDialog(MyOperations.this,
     "Please Enter a Number");
```

```
106:                    textNum1.selectAll();
107:                    textNum1.requestFocus();
108:                    return;
109:                }
110:
111:                double num2 = 0;
112:
113:                try
114:                {
115:                    num2 =
    Double.parseDouble(textNum2.getText());
116:                }
117:
118:                catch(Exception ex)
119:                {
120:
     JOptionPane.showMessageDialog(MyOperations.this,
    "Please Enter a Number");
121:                    textNum2.selectAll();
122:                    textNum2.requestFocus();
123:                    return;
124:                }
125:
126:                String selectedOperation =
    (String)operation.getSelectedItem();
127:                int switchOperation = 0;
128:
129:                if ( selectedOperation.equals("+") )
130:                {
131:                    switchOperation = 1;
132:                }
133:
134:                else if ( selectedOperation.equals("-") )
135:                {
136:                    switchOperation = 2;
```

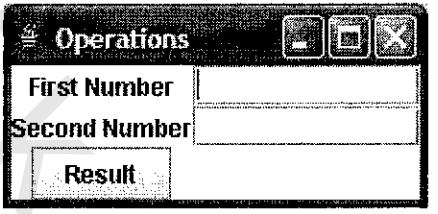
```
137:                }
138:
139:                else if ( selectedOperation.equals("*") )
140:                {
141:                    switchOperation = 3;
142:                }
143:
144:                else if ( selectedOperation.equals("/") )
145:                {
146:                    switchOperation = 4;
147:                }
148:
149:            if ( obj == result )
150:            {
151:                double result = 0;
152:                String message ="";
153:
154:                switch (switchOperation)
155:                {
156:                    case 1:
157:                            result = num1 + num2;
158:                            message = "Sum of " +
    num1 + " and " + num2 + " = " + result;
159:                            break;
160:
161:                    case 2:
162:                            result = num1 - num2;
163:                            message = "Subraction of "
    + num1 + " and " + num2 + " = " + result;
164:                            break;
165:
166:                    case 3:
167:                            result = num1 * num2;
168:                            message = "Product of " +
    num1 + " and " + num2 + " = " + result;
```

```
169:                                    break;
170:
171:                            case 4:
172:                                    result = num1 / num2;
173:                                    message = "Division of " +
    num1 + " and " + num2 + " = " + result;
174:                                    break;
175:                            default:
176:                                    message = "Unsupported
    Operation";
177:                                    break;
178:                    }
179:
180:
     JOptionPane.showMessageDialog(MyOperations.this,
    message);
181:                }
182:            }
183:    }
184:
185:    class MyKeyListener extends KeyAdapter
186:    {
187:        public void keyReleased(KeyEvent e)
188:        {
189:            char c = e.getKeyChar();
190:
191:            if ( c == '\n' )
192:            {
193:                result.doClick();
194:            }
195:        }
196:    }
197:    }
```

مثال (6): حساب الرقم الأكبر والأصغر:

أولاً: هدف المثال:

اكتب برنامجاً يقوم بإدخال رقمين في أداة إطار JFrame كما هو واضح في الشكل التالي ثم يقوم بطبع الرقم الأكبر والأصغر ، كما يجب أن يتعامل البرنامج مع احتمال أن يقوم المستخدم بإدخال حروف بدلاً من أرقام. إذا ضغط المستخدم علي زر الإدخال Enter فيجب أن يقوم البرنامج بطباعة الناتج كأن المستخدم قام بالضغط علي الزر Result بالضبط.



الشكل (18-6) تنفيذ البرنامج

ثانياً: كود البرمجة:

```
1:  import javax.swing.*;
2:  import java.awt.*;
3:  import java.awt.event.*;
4:
5:  public class MyOperations extends JFrame
6:  {
7:    JLabel labelNum1,labelNum2;
8:    JTextField textNum1,textNum2;
9:    JButton result;
10:  GridBagLayout grid;
11:  GridBagConstraints gbc;
12:
13:  MyOperations()
14:  {
15:       addWindowListener(new WindowAdapter()
16:            {
```

```
17:                 public void windowClosing(WindowEvent
    we)
18:                 {
19:                     dispose();
20:                     System.exit(0);
21:                 }
22:             }
23:         );
24:
25:         Container c = getContentPane();
26:
27:         grid = new GridBagLayout();
28:         gbc = new GridBagConstraints();
29:         c.setLayout(grid);
30:
31:         labelNum1 = new JLabel("First Number");
32:         labelNum2 = new JLabel("Second Number");
33:
34:         textNum1 = new JTextField(10);
35:         textNum2 = new JTextField(10);
36:
37:         MyKeyListener k = new MyKeyListener();
38:         textNum1.addKeyListener(k);
39:         textNum2.addKeyListener(k);
40:
41:         ButtonListener b = new ButtonListener();
42:
43:         result = new JButton("Result");
44:         result.addActionListener(b);
45:
46:         gbc.gridx = 1;
47:         gbc.gridy = 1;
48:         grid.setConstraints(labelNum1,gbc);
49:         c.add(labelNum1);
```

```
50:
51:         gbc.gridx = 2;
52:         gbc.gridy = 1;
53:         grid.setConstraints(textNum1,gbc);
54:         c.add(textNum1);
55:
56:         gbc.gridx = 1;
57:         gbc.gridy = 2;
58:         grid.setConstraints(labelNum2,gbc);
59:         c.add(labelNum2);
60:
61:         gbc.gridx = 2;
62:         gbc.gridy = 2;
63:         grid.setConstraints(textNum2,gbc);
64:         c.add(textNum2);
65:
66:         gbc.gridx = 1;
67:         gbc.gridy = 3;
68:         grid.setConstraints(result,gbc);
69:         c.add(result);
70:
71:         setTitle("Operations");
72:         pack();
73:         setVisible(true);
74:   }
75:
76:  public   double   getMinimum(double   num1,double
     num2)
77:  {
78:         if ( num1 <= num2 )
79:         {
80:             return num1;
81:         }
82:
83:         else
```

```
84:       {
85:           return num2;
86:       }
87:  }
88:
89:  public double getMaximum(double num1,double
     num2)
90:  {
91:       if ( num1 <= num2 )
92:       {
93:           return num2;
94:       }
95:
96:       else
97:       {
98:           return num1;
99:       }
100:      }
101:
102:      public static void main(String args[])
103:      {
104:          MyOperations myOperations = new
     MyOperations();
105:      }
106:
107:      class ButtonListener implements ActionListener
108:      {
109:          public void actionPerformed(ActionEvent e)
110:          {
111:              Object obj = e.getSource();
112:
113:              double num1 = 0;
114:
115:                  try
116:                  {
```

```
117:                    num1                    =
   Double.parseDouble(textNum1.getText());
118:                }
119:
120:                catch(Exception ex)
121:                {
122:
    JOptionPane.showMessageDialog(MyOperations.this,
   "Please Enter a Number");
123:                    textNum1.selectAll();
124:                    textNum1.requestFocus();
125:                    return;
126:                }
127:
128:                double num2 = 0;
129:
130:                try
131:                {
132:                    num2                    =
   Double.parseDouble(textNum2.getText());
133:                }
134:
135:                catch(Exception ex)
136:                {
137:
    JOptionPane.showMessageDialog(MyOperations.this,
   "Please Enter a Number");
138:                    textNum2.selectAll();
139:                    textNum2.requestFocus();
140:                    return;
141:                }
142:
143:            if ( obj == result )
144:            {
145:                double          max          =
```

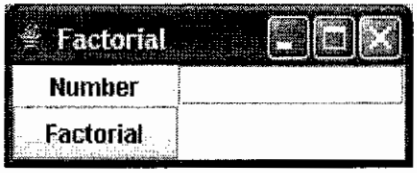
```
    getMaximum(num1,num2);
146:                double min =
    getMinimum(num1,num2);
147:
148:                String message = "Minimum of " +
    num1 + " and " + num2 + " = " + min + "\n";
149:                message += "Maximum of " + num1 +
    " and " + num2 + " = " + max + "\n";
150:
     JOptionPane.showMessageDialog(MyOperations.this,
    message);
151:            }
152:        }
153:    }
154:
155:    class MyKeyListener extends KeyAdapter
156:    {
157:        public void keyReleased(KeyEvent e)
158:        {
159:            char c = e.getKeyChar();
160:
161:            if ( c == '\n' )
162:            {
163:                result.doClick();
164:            }
165:        }
166:    }
167:    }
```

مثال (7): حساب مضروب Factorial رقم:

أولاً: هدف المثال:

اكتب برنامجاً يقوم بإدخال رقم في أداة إطار JFrame كما هو واضح في الشكل التالي ثم يقوم بطباعة مضروب Factorial الرقم. يجب أن يقوم البرنامج بإظهار رسالة خطأ في حالة

إدخال رقم أصغر من الصفر ، كما يجب أن يتعامل البرنامج مع احتمال أن يقـوم المسـتخدم بإدخال حروف بدلاً من أرقام. إذا ضغط المستخدم علي زر الإدخال Enter فيجب أن يقوم البرنامج بطباعة الناتج كأن المستخدم قام بالضغط علي الزر Factorial بالضبط.



الشكل (18-7) تنفيذ البرنامج

ثانياً: كود البرمجة:

```
1:  import javax.swing.*;
2:  import java.awt.*;
3:  import java.awt.event.*;
4:
5:  public class MyFactorial extends JFrame
6:  {
7:    JLabel labelNum1;
8:    JTextField textNum1;
9:    JButton factorial;
10:  GridBagLayout grid;
11:  GridBagConstraints gbc;
12:
13:  MyFactorial()
14:  {
15:       addWindowListener(new WindowAdapter()
16:            {
17:                 public void windowClosing(WindowEvent
     we)
18:                 {
19:                      dispose();
20:                      System.exit(0);
21:                 }
22:            }
```

```
23:     );
24:
25:     Container c = getContentPane();
26:
27:     grid = new GridBagLayout();
28:     gbc = new GridBagConstraints();
29:     c.setLayout(grid);
30:
31:     labelNum1 = new JLabel("Number");
32:
33:     textNum1 = new JTextField(10);
34:
35:     textNum1.addKeyListener(new KeyAdapter()
36:         {
37:             public void keyReleased(KeyEvent e)
38:             {
39:                 char c = e.getKeyChar();
40:
41:                 if ( c == '\n' )
42:                 {
43:                     factorial.doClick();
44:                 }
45:             }
46:         }
47:     );
48:
49:     ButtonListener b = new ButtonListener();
50:
51:     factorial = new JButton("Factorial");
52:     factorial.addActionListener(b);
53:
54:     gbc.gridx = 1;
55:     gbc.gridy = 1;
56:     grid.setConstraints(labelNum1,gbc);
57:     c.add(labelNum1);
```

openkan.com

```
58:
59:         gbc.gridx = 2;
60:         gbc.gridy = 1;
61:         grid.setConstraints(textNum1,gbc);
62:         c.add(textNum1);
63:
64:         gbc.gridx = 1;
65:         gbc.gridy = 2;
66:         grid.setConstraints(factorial,gbc);
67:         c.add(factorial);
68:
69:         setTitle("Factorial");
70:         pack();
71:         setVisible(true);
72:   }
73:
74:   public static void main(String args[])
75:   {
76:         MyFactorial myFactorial = new MyFactorial();
77:   }
78:
79:   class ButtonListener implements ActionListener
80:   {
81:         public void actionPerformed(ActionEvent e)
82:         {
83:               Object obj = e.getSource();
84:
85:               int num1 = 0;
86:
87:                     try
88:                     {
89:                           num1                                    =
      Integer.parseInt(textNum1.getText());
90:                     }
91:
```

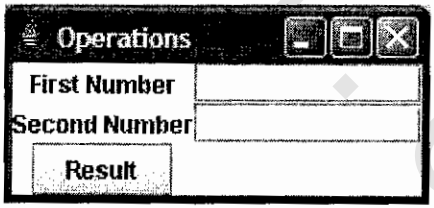
```
92:                 catch(Exception ex)
93:                 {
94:
      JOptionPane.showMessageDialog(MyFactorial.this,"P
    lease Enter a Number");
95:                     textNum1.selectAll();
96:                     textNum1.requestFocus();
97:                     return;
98:                 }
99:
100:                if ( obj == factorial )
101:                {
102:                    if ( num1 < 0 )
103:                    {
104:
      JOptionPane.showMessageDialog(MyFactorial.this,"P
    lease enter a positive number");
105:                        return;
106:                    }
107:
108:                    long result = 1;
109:
110:                    if ( num1 == 0 )
111:                    {
112:                        result = 1;
113:                    }
114:
115:                    else
116:                    {
117:                        for ( int i = 1 ; i <= num1 ; i++ )
118:                        {
119:                            result *= i;
120:                        }
121:                    }
122:
```

```
123:                    String message = "Factorial = " +
    result;
124:
        JOptionPane.showMessageDialog(MyFactorial.this,m
    essage);
125:                }
126:            }
127:            }
128:        }
```

مثال (8): مجموع متتابعة هندسية Geometric Sequence:

أولاً: هدف المثال:

اكتب برنامجاً يقوم بإدخال رقمين x و n في أداة إطار JFrame كما هو واضح في الشكل التالي ثم يقوم بطبع قيمة $1 + x + x^2 + x^3 + \ldots + x^n$ ، كما يجب أن يتعامل البرنامج مع احتمال أن يقوم المستخدم بإدخال حروف بدلاً من أرقام. إذا ضغط المستخدم علي زر الإدخال Enter فيجب أن يقوم البرنامج بطباعة الناتج كأن المستخدم قام بالضغط علي الزر Result بالضبط.



الشكل (18-8) تنفيذ البرنامج

ثانياً: كود البرمجة:

```
1:  import javax.swing.*;
2:  import java.awt.*;
3:  import java.awt.event.*;
4:
5:  public class MyOperations extends JFrame
6:  {
```

```
7:  JLabel labelNum1,labelNum2;
8:  JTextField textNum1,textNum2;
9:  JButton result;
10: GridBagLayout grid;
11: GridBagConstraints gbc;
12:
13: MyOperations()
14: {
15:     addWindowListener(new WindowAdapter()
16:     {
17:         public void windowClosing(WindowEvent
    we)
18:         {
19:             dispose();
20:             System.exit(0);
21:         }
22:     }
23:     );
24:
25:     Container c = getContentPane();
26:
27:     grid = new GridBagLayout();
28:     gbc = new GridBagConstraints();
29:     c.setLayout(grid);
30:
31:     labelNum1 = new JLabel("First Number");
32:     labelNum2 = new JLabel("Second Number");
33:
34:     textNum1 = new JTextField(10);
35:     textNum2 = new JTextField(10);
36:
37:     MyKeyListener k = new MyKeyListener();
38:     textNum1.addKeyListener(k);
39:     textNum2.addKeyListener(k);
40:
```

obeikan.com

```
41:         ButtonListener b = new ButtonListener();
42:
43:         result = new JButton("Result");
44:         result.addActionListener(b);
45:
46:         gbc.gridx = 1;
47:         gbc.gridy = 1;
48:         grid.setConstraints(labelNum1,gbc);
49:         c.add(labelNum1);
50:
51:         gbc.gridx = 2;
52:         gbc.gridy = 1;
53:         grid.setConstraints(textNum1,gbc);
54:         c.add(textNum1);
55:
56:         gbc.gridx = 1;
57:         gbc.gridy = 2;
58:         grid.setConstraints(labelNum2,gbc);
59:         c.add(labelNum2);
60:
61:         gbc.gridx = 2;
62:         gbc.gridy = 2;
63:         grid.setConstraints(textNum2,gbc);
64:         c.add(textNum2);
65:
66:         gbc.gridx = 1;
67:         gbc.gridy = 3;
68:         grid.setConstraints(result,gbc);
69:         c.add(result);
70:
71:         setTitle("Operations");
72:         pack();
73:         setVisible(true);
74:     }
75:
```

obeikan.com

```
76:  public static void main(String args[])
77:  {
78:      MyOperations myOperations = new
     MyOperations();
79:  }
80:
81:  class ButtonListener implements ActionListener
82:  {
83:      public void actionPerformed(ActionEvent e)
84:      {
85:          Object obj = e.getSource();
86:
87:          double num1 = 0;
88:
89:              try
90:              {
91:                  num1 =
     Double.parseDouble(textNum1.getText());
92:              }
93:
94:              catch(Exception ex)
95:              {
96:
     JOptionPane.showMessageDialog(MyOperations.this,"
     Please Enter a Number");
97:                  textNum1.selectAll();
98:                  textNum1.requestFocus();
99:                  return;
100:                 }
101:
102:                 double num2 = 0;
103:
104:                 try
105:                 {
106:                     num2 =
```

```
    Double.parseDouble(textNum2.getText());
107:                }
108:
109:                catch(Exception ex)
110:                {
111:
    JOptionPane.showMessageDialog(MyOperations.this,"
    Please Enter a Number");
112:                    textNum2.selectAll();
113:                    textNum2.requestFocus();
114:                    return;
115:                }
116:
117:            if ( obj == result )
118:            {
119:                double result = 1;
120:                int i;
121:                double y = num1;
122:
123:                for ( i = 1 ; i <= num2 ; i++ )
124:                {
125:                    result += num1;
126:                    num1 *= y;
127:                }
128:
129:                String message = "Sum = " + result;
130:
     JOptionPane.showMessageDialog(MyOperations.this,
    message);
131:            }
132:        }
133:    }
134:
135:    class MyKeyListener extends KeyAdapter
136:    {
```

```
137:        public void keyReleased(KeyEvent e)
138:        {
139:            char c = e.getKeyChar();
140:
141:            if ( c == '\n' )
142:            {
143:                result.doClick();
144:            }
145:        }
146:    }
147:    }
```

ملخص الفصل:

- تعلمنا في هذا الفصل العديد من تطبيقات مبادئ اللغة.
- في الفصل القادم نتعلم - بإذن الله - العديد من التطبيقات الأخري ، فتابع معنا الفصل القادم.

openkan.com