

تقديم به همه هموطنان عزیزم

آموزش زبان برنامه نویسی جاوا

گرافیک در جاوا - پکیج Swing

جلسه بیست و یکم

کلاس JFileChooser

نویسنده: رحمان زارعی

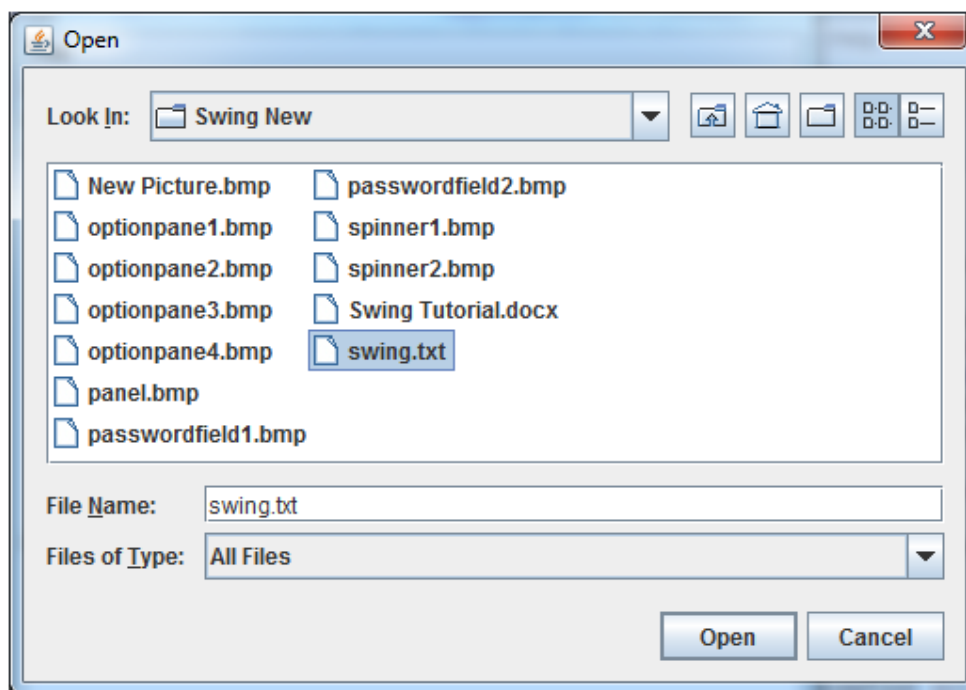
جاوا را ساده، آسان و شیرین بنوشید!!!!



این جلسه آموزشی رایگان است، فروش و ویرایش آن ممنوع و حرام می باشد. اما این کتاب را می توانید همین جور که هست در سایت و شبکه اجتماعی خود به اشتراک بگذارید.

خب به ایستگاه کلاس JFileChooser یکی دیگر از اجزای گرافیکی یا component های موجود در پکیج Swing جاوا رسیدیم. کلاس JFileChooser همان طور که از اسمش پیداست یک مکانیزم ساده است که انتخاب یک فایل در محلی از کامپیوتر توسط کاربر را فراهم می کند.

برای درک بهتر نسبت به شکل ظاهری این اجزای گرافیکی در برنامه تصویر (۱) را مشاهده کنید:



تصویر (۱)-FileChooser

- همان طور که در تصویر (۱) مشاهده می کنید، در کامپیوتر با FileChooser برای انتخاب فایل مورد نظر در محلی از کامپیوتر زیاد برخورد کرده ایم.

- برای استفاده از این کلاس در برنامه خود باید پکیج زیر را بالای کلاس برنامه خود import کنید:

```
import javax.swing.JFileChooser;
```

یا پکیج زیر رو نیز می توانید استفاده کنید:

```
import javax.swing.*;
```

- این پکیج همه کلاس های گرافیکی موجود در Swing جاوا را شامل می شود.

• سازنده های پر کاربرد کلاس JFileChooser :

سازنده	کاربرد
JFileChooser()	ایجاد یک JFileChooser که به directories پیشفرض اشاره دارد. منظور از directories (دایرکتوری) همون Folder یا پوشه موجود در کامپیوتر می باشد.
JFileChooser(File currentDirectory)	ایجاد یک JFileChooser که یک شی از نوع File که حاوی آدرس فایل مورد نظر در کامپیوتر هستش به عنوان پارامتر به آن می دهیم.
JFileChooser(String currentDirectoryPath)	ایجاد یک JFileChooser که آدرس محل مورد نظر در کامپیوتر یا همون پوشه مورد نظر را به آن می دهیم. مثلا میخوایم وقتی از FileChooser استفاده می کنیم پوشه image در درایو F را باز کند ، آدرس پوشه image در درایو F را به عنوان پارامتر به این سازنده می دهیم.

توضیحات رو ببخمال! بریم سراغ مثال که چه چیزی بهتر از مثال 😊

مثال:

```
package fileChooser_javalike;

import javax.swing.*;

import java.awt.event.*;
import java.io.*;

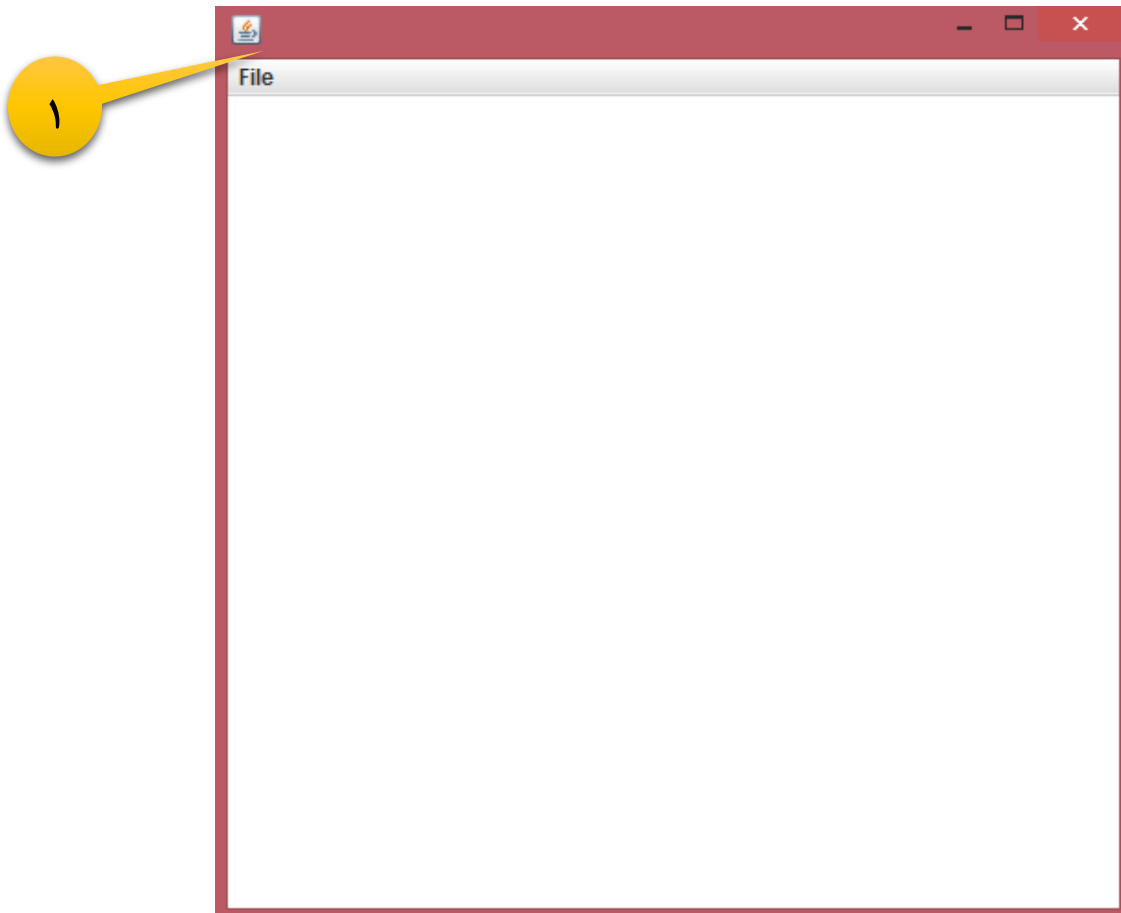
public class FileChooserExample extends JFrame implements ActionListener {
    JMenuBar mb;
    JMenu file;
    JMenuItem open;
    JTextArea ta;

    FileChooserExample() {
        open = new JMenuItem("Open File");
        open.addActionListener(this);
        file = new JMenu("File");
        file.add(open);
        mb = new JMenuBar();
        mb.setBounds(0, 0, 800, 20);
        mb.add(file);
        ta = new JTextArea(800, 800);
        ta.setBounds(0, 20, 800, 800);
        add(mb);
        add(ta);
    }

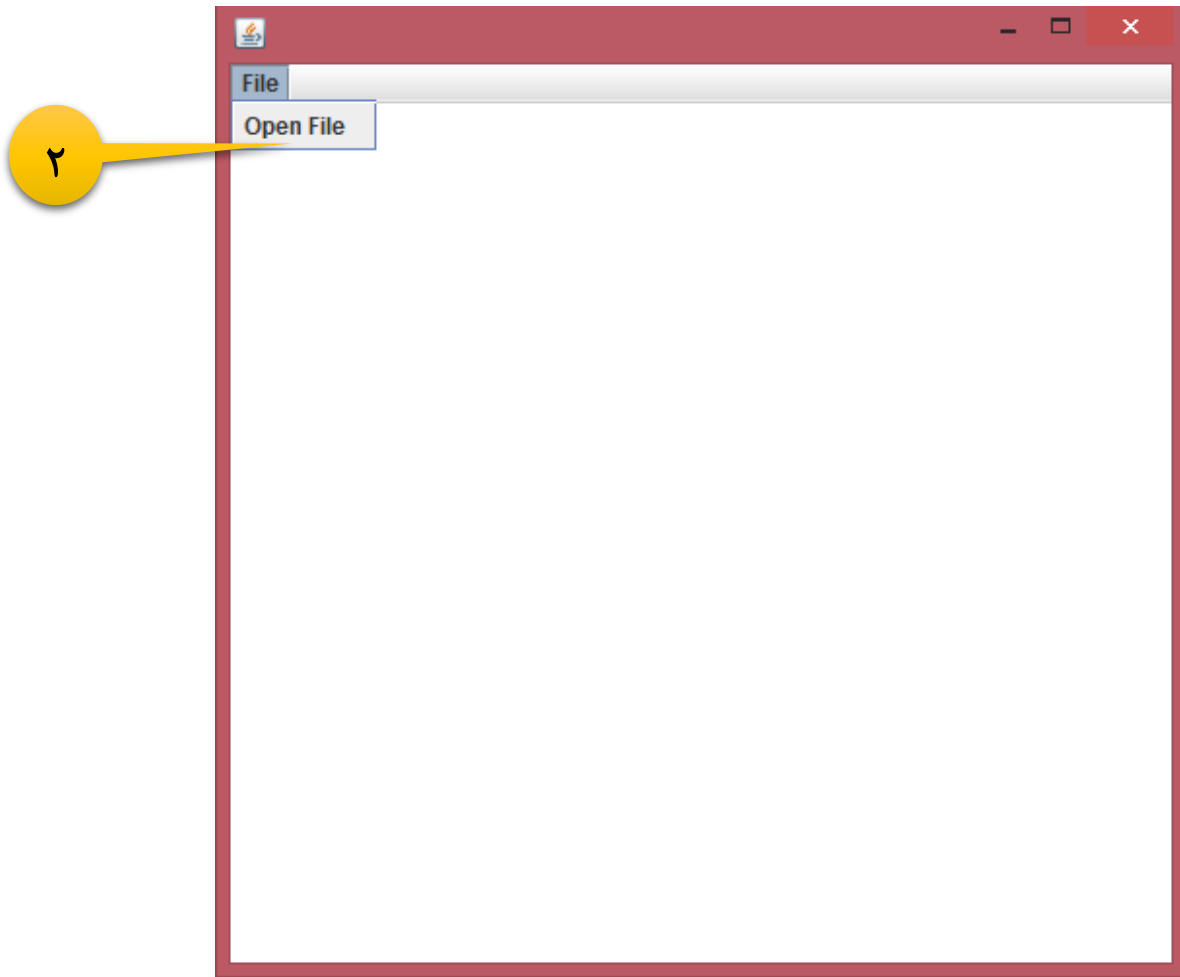
    public void actionPerformed(ActionEvent e) {
        if (e.getSource() == open) {
            JFileChooser fc = new JFileChooser();
            int i = fc.showOpenDialog(this);
            if (i == JFileChooser.APPROVE_OPTION) {
                File f = fc.getSelectedFile();
                String filepath = f.getPath();
                try {
                    BufferedReader br = new BufferedReader(new FileReader(
                        filepath));
                    String s1 = "", s2 = "";
                    while ((s1 = br.readLine()) != null) {
                        s2 += s1 + "\n";
                    }
                    ta.setText(s2);
                    br.close();
                } catch (Exception ex) {
                    ex.printStackTrace();
                }
            }
        }
    }
}
```

```
    }  
    }  
}  
  
public static void main(String[] args) {  
    FileChooserExample om = new FileChooserExample();  
    om.setSize(500, 500);  
    om.setLayout(null);  
    om.setVisible(true);  
    om.setDefaultCloseOperation(EXIT_ON_CLOSE);  
}  
}
```

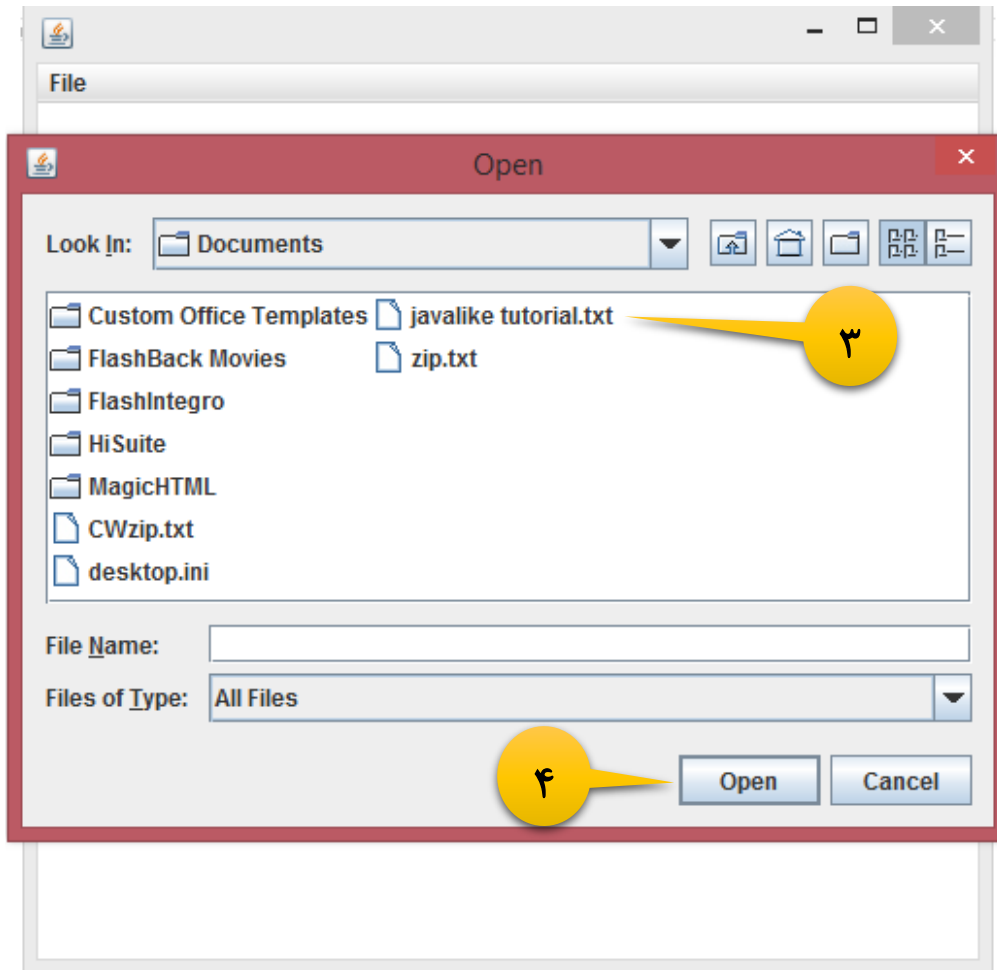
خروجی: تصاویر (۲) و (۳) و (۴)



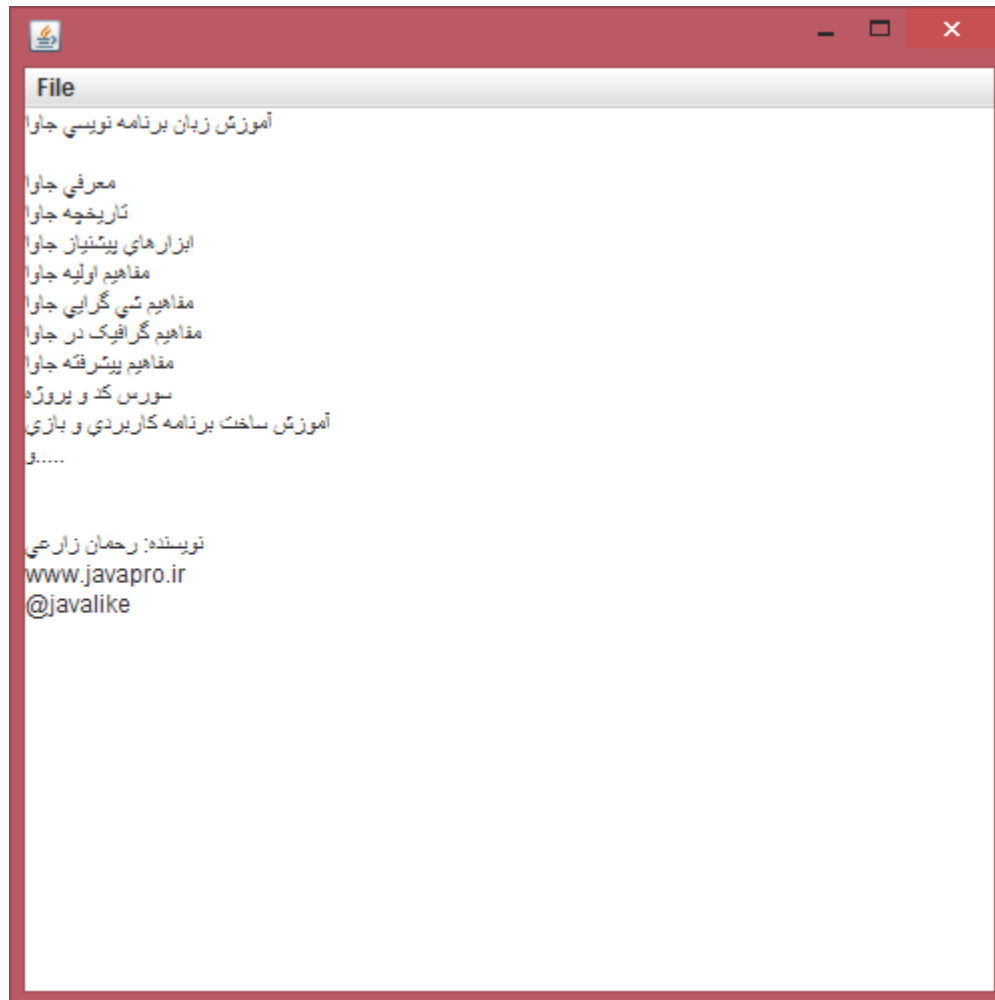
تصویر (۲)



تصویر (۳)



تصویر (۴)



تصویر (۵)

- در برنامه بالا ما یک فایل متنی با فرمت `txt` را باز کردیم و محتوای درون آن را داخل یک `TextArea` ریختیم.

توضیحات:

```
public class FileChooserExample extends JFrame implements ActionListener {
```

کلاس ما برای استفاده از متد `actionPerformed` اینترفیس `ActionListener` را `implements` کرده است.

```
1. open = new JMenuItem("Open File");
2. open.addActionListener(this);
3. file = new JMenu("File");
4. file.add(open);
5. mb = new JMenuBar();
6. mb.setBounds(0, 0, 800, 20);
```

```

7. mb.add(file);
8. ta = new JTextArea(800, 800);
9. ta.setBounds(0, 20, 800, 800);
10.    add(mb);
11.    add(ta);

```

- در سازنده کلاس خود موارد زیر را انجام داده ایم: (توضیحات اشاره به شماره هر خط در کد بالا دارد)
 ۱. ایجاد یک شی از نوع کلاس JMenuItem که رشته "Open File" را به عنوان پارامتر به سازنده آن داده ایم. شی از نوع این کلاس آیتم های منو ما را تشکیل می دهد.
 ۲. به آیتم خود یک ActionListener اضافه کرده ایم که برای دریافت رویداد های مربوط به کلیک کردن روی آیتم مورد نظر استفاده می شود.
 ۳. از کلاس JMenuItem نیز شی ساخته ایم. رشته "File" که نام منو ما را تشکیل می دهد را به عنوان پارامتر به سازنده این کلاس داده ایم. شی ایجاد شده از این کلاس نقش منو ما را بازی می کند.
 ۴. آیتم منو خود را به منو موجود در منو بار اضافه کرده ایم.
 ۵. از کلاس JMenuItem شی ایجاد کرده ایم. شی ایجاد شده از این کلاس نقش منو بار ما را بازی خواهد کرد.
 ۶. تعیین ابعاد و مختصات منو بار
 ۷. اضافه کردن منو به منو بار
 ۸. ایجاد یک
 ۹. تعیین ابعاد و مختصات JTextArea
 ۱۰. اضافه کردن منو بار به فریم
 ۱۱. اضافه کردن JTextArea به فریم

```

1. public void actionPerformed(ActionEvent e) {
2.   if (e.getSource() == open) {
3.     JFileChooser fc = new JFileChooser();
4.     int i = fc.showOpenDialog(this);
5.     if (i == JFileChooser.APPROVE_OPTION) {
6.       File f = fc.getSelectedFile();

```

۱. متد actionPerformed که وظیفه دریافت رویدادهای مربوط به اجزا یا component های گرافیکی را برعهده دارد. با هربار رویداد رخدادی خاص این متد صدا زده می شود و دستورات درون آن اجرا می شود.
۲. این دستور منبع رویداد رخ داده در برنامه را پیدا میکند. و میگوید اگر برابر شی open که از نوع کلاس JMenuItem هستش دستور مورد نظر را اجرا کند. دستور مورد نظر مربوط به شرط if

۳. از کلاس JFileChooser شی ایجاد می کنیم.

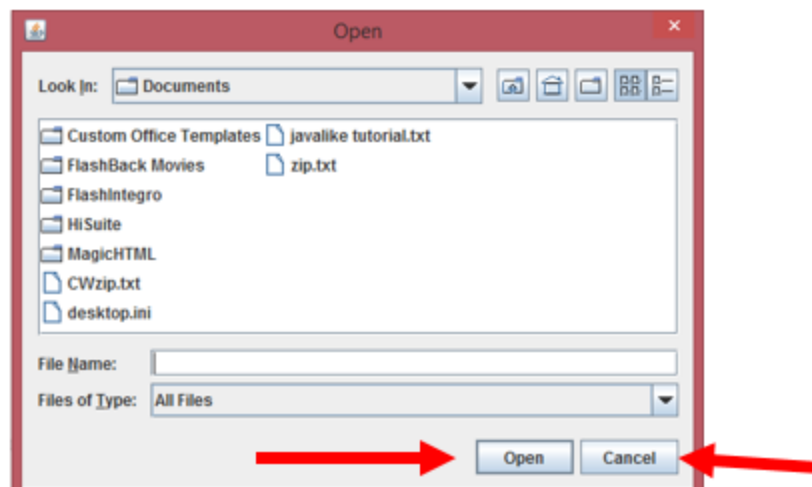
```
int i = fc.showOpenDialog(this);
```

۴. ما FileChooser برای انتخاب فایل در محلی از کامپیوتر استفاده می کنیم در اینجا تعیین کرده ایم که قصد داریم فایل را باز کنیم (بخوانیم) برای این کار از متد showOpenDialog که پارامتر آن از نوع کلاس Component هستش استفاده کرده ایم کلمه کلیدی this به شی از کلاسمون که component یا اجزای گرافیکی JFrame را به ارث برده است اشاره دارد. بعد از انتخاب فایل موجود در کامپیوتر متد showOpenDialog یک ایندکس از نوع عدد صحیح را برای ما برمیگرداند.

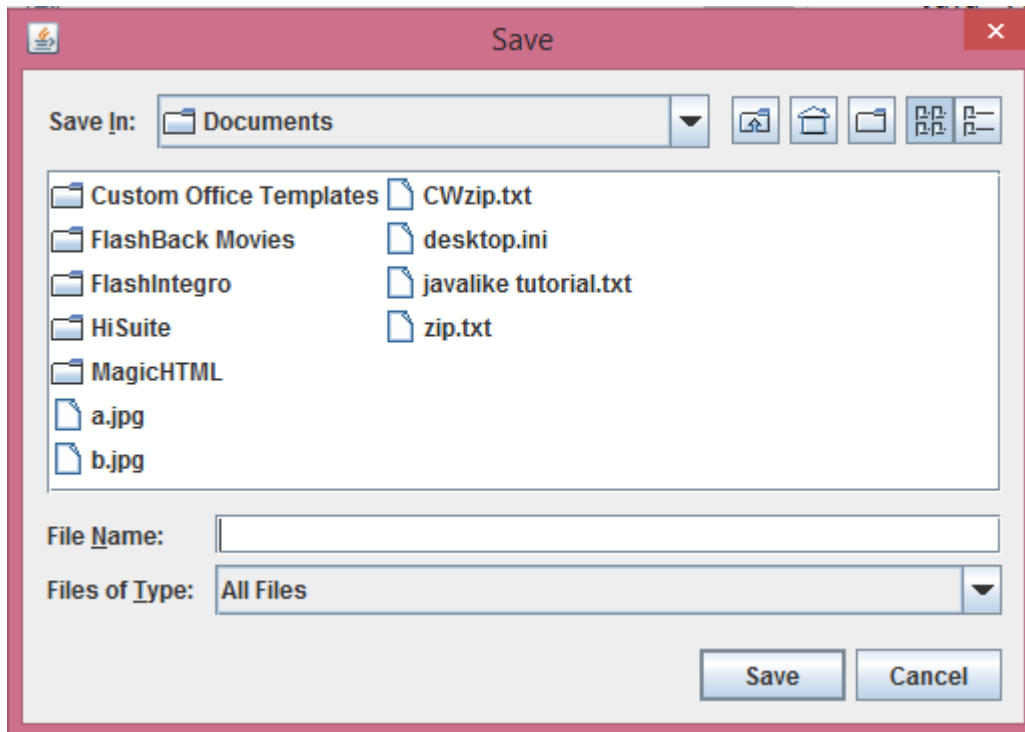
```
if (i == JFileChooser.APPROVE_OPTION) {
```

۵. اینجا گفته ایندکسی که برگردانده شده اگر برابر مقدار JFileChooser.APPROVE_OPTION هستش دستور شرط رو اجرا کن.وقتی متد showOpenDialog صدا زده میشه پیام انتخاب فایل به صورت تصویر(۶) نمایش داده می شود، با زدن دکمه open متد showOpenDialog مقدار 0 را برمی گرداند ، با زدن دکمه cancel متد showOpenDialog مقدار 1 را برمیگرداند. خب دستور ثابت JFileChooser.APPROVE_OPTION نیز حاوی مقدار 0 می باشد که بیان گر موافقت در انتخاب فایل یا همان open کردن فایل می باشد. مثلا دستور ثابت مربوط به cancel کردن برابر JFileChooser.CANCEL_OPTION می باشد وقتی دکمه cancel را انتخاب میکنیم مقدار 1 برای ما برگردانده میشه که برابر دستور JFileChooser.CANCEL_OPTION می باشد.

نکته: متد showSaveDialog که درون کلاس JFileChooser می باشد برای نمایش پیام save کردن می باشد. تصویر(۷)



تصویر(۶)

تصویر (۷) - متد `showSaveDialog`

۶. حالا اگر انتخاب ما دکمه `open` بود فایل موجود در محل کامپیوتر خوانده میشه و درون یک شی از نوع کلاس `File` ریخته می شود.

```
7. String filepath = f.getPath();
8. try {
9.     BufferedReader br = new BufferedReader(new FileReader(
10.         filepath));
11.     String s1 = "", s2 = "";
12.     while ((s1 = br.readLine()) != null) {
13.         s2 += s1 + "\n";
14.     }
15.     ta.setText(s2);
16.     br.close();
17. } catch (Exception ex) {
18.     ex.printStackTrace();
19. }
20. }
21. }
22. }
```

۷. حالا آدرس فایل درون یک متغیر از نوع رشته ریخته می شود.

۸. خط ۸ تا ۱۹ فرآیند خواندن فایل در جاوا هستش که در جلسات کار با فایل بررسی کرده ایم.
 ۹. از کلاس `BufferedReader` برای خواندن فایل در جاوا استفاده می شود.
 ۱۱. دو متغیر از نوع رشته تعریف کرده ایم.
 ۱۲. سطر به سطر فایل خوانده شده را می پیماییم و درون متغیر `s1` میریزیم
 ۱۳. حالا مقدار متن هر سطر را به متغیر `s2` اضافه می کنیم.
 ۱۵. متغیر `s2` که حاوی تمام متن سطر های فایل می باشد را درون شی از نوع `TextArea` می ریزیم.
 ۱۶. فایل را می بندیم.
- بلوک `try-catch` برای کنترل استثناهای احتمالی در مورد کار با فایل استفاده شده است.
- سایر دستورات را در جلسات گذشته بررسی کرده ایم.

مثال پیشتر.....

مثال: برنامه زیر یک فایل از محلی از کامپیوتر میخواند (`open`) و در محل مورد نظر در کامپیوتر فایل را ذخیره می کند (`save`).

```
package fileChooser_javalike;

import javax.swing.*;

import java.awt.Dimension;
import java.awt.Image;
import java.awt.event.*;
import java.io.*;

public class FileChooserExample extends JFrame implements ActionListener {
    JMenuBar mb;
    JMenu file;
    JMenuItem open;
    JMenuItem save;
    JLabel label;

    FileChooserExample() {
        open = new JMenuItem("Open File");
        save = new JMenuItem("Save File");
        open.addActionListener(this);
        save.addActionListener(this);
    }
}
```

```
file = new JMenu("File");
file.add(open);
file.add(save);
mb = new JMenuBar();
mb.setBounds(0, 0, 800, 20);
mb.add(file);
label = new JLabel();
label.setBounds(0, 20, 400, 400);
add(mb);
add(label);
setSize(400, 400);
setLayout(null);
setVisible(true);
setDefaultCloseOperation(EXIT_ON_CLOSE);
}

{

}

static void copyFileUsingStream(String source, String dest) {

try {
InputStream is = new FileInputStream(new File(source));
OutputStream os = new FileOutputStream(new File(dest));

byte[] buffer = new byte[1024];
int length;
while ((length = is.read(buffer)) > 0) {
os.write(buffer, 0, length);
}
is.close();
os.close();
} catch (IOException e) {
}

}

File openFile = null;
File saveFile = null;
boolean on;

public void actionPerformed(ActionEvent e) {

JFileChooser fc = new JFileChooser();
if (e.getSource() == open) {
```

```
int i = fc.showSaveDialog(this);

if (i == JFileChooser.APPROVE_OPTION) {

openFile = fc.getSelectedFile();
on = true;
JOptionPane.showMessageDialog(this, openFile.getName()
+ " file was opened successfully", "Open File",
JOptionPane.OK_OPTION);
}

}

if (e.getSource() == save) {

String nameFile = new String();
if (on) {
nameFile = openFile.getName();
fc.setSelectedFile(new File(nameFile));
}
int i = fc.showSaveDialog(this);

if (i == JFileChooser.APPROVE_OPTION) {
if (on) {
saveFile = fc.getSelectedFile();
;

nameFile = nameFile.replace('.', ' ');

String format[] = nameFile.split(" ");
int a = JOptionPane.showConfirmDialog(this,
"Are you sure you save the file?");
if (a == JOptionPane.YES_OPTION) {
copyFileUsingStream(openFile.getPath(),
saveFile.getPath() + "."
+ format[format.length - 1]);
JOptionPane.showMessageDialog(this, openFile.getName()
+ " Files saved to succeed", "Save File",
JOptionPane.OK_OPTION);
}

}

}

}

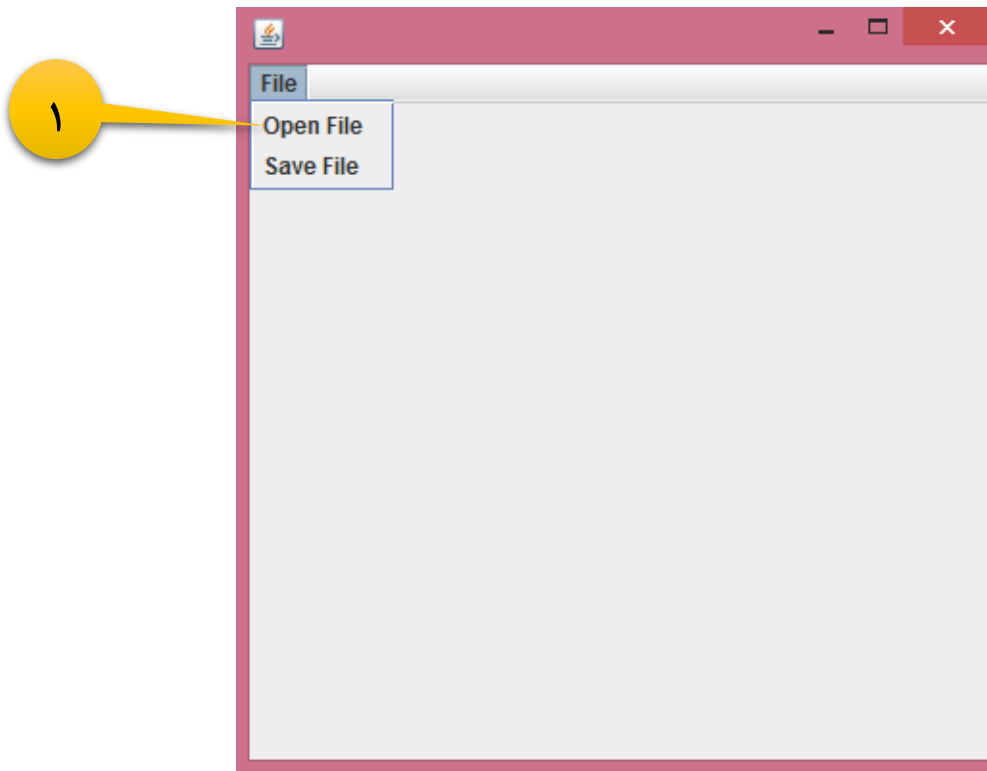
}

}
```

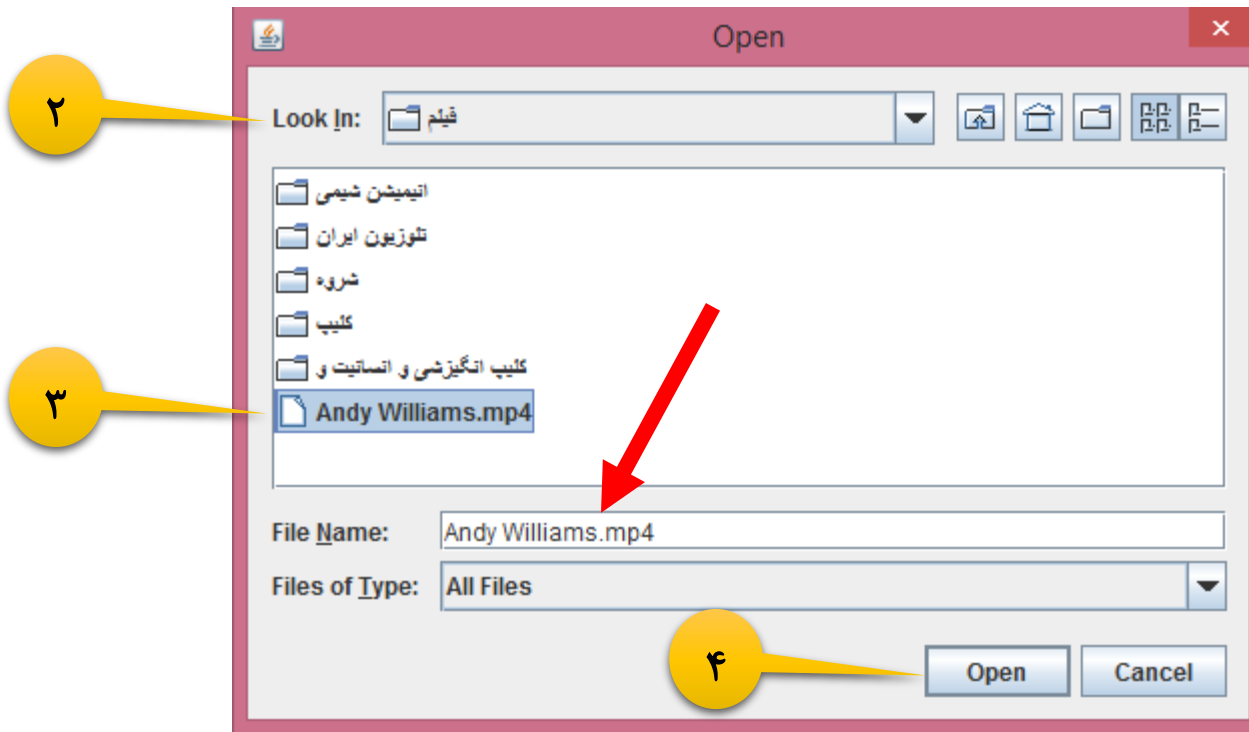
```
public static void main(String[] args) {  
FileChooserExample om = new FileChooserExample();  
}  
}
```

خروجی: برنامه دارای یک منو **file** می باشد که دارای دو آیتم **save** و **open** هستش. ما از طریق **open** فایل مورد نظر را از پوشه مورد نظر در کامپیوتر میخوانیم و از طریق آیتم **save** همان فایل را در مکان دیگر از کامپیوتر **save** یا ذخیره می کنیم. برای تست یک فایل با فرمت **mp4** را از گوشه ای کامپیوتر **open** کرده و در محل دیگری از کامپیوتر همان فایل را **save** کرده ایم. تصاویر (۸)، (۹)، (۱۰)، (۱۱)، (۱۲)، (۱۳) و (۱۴)

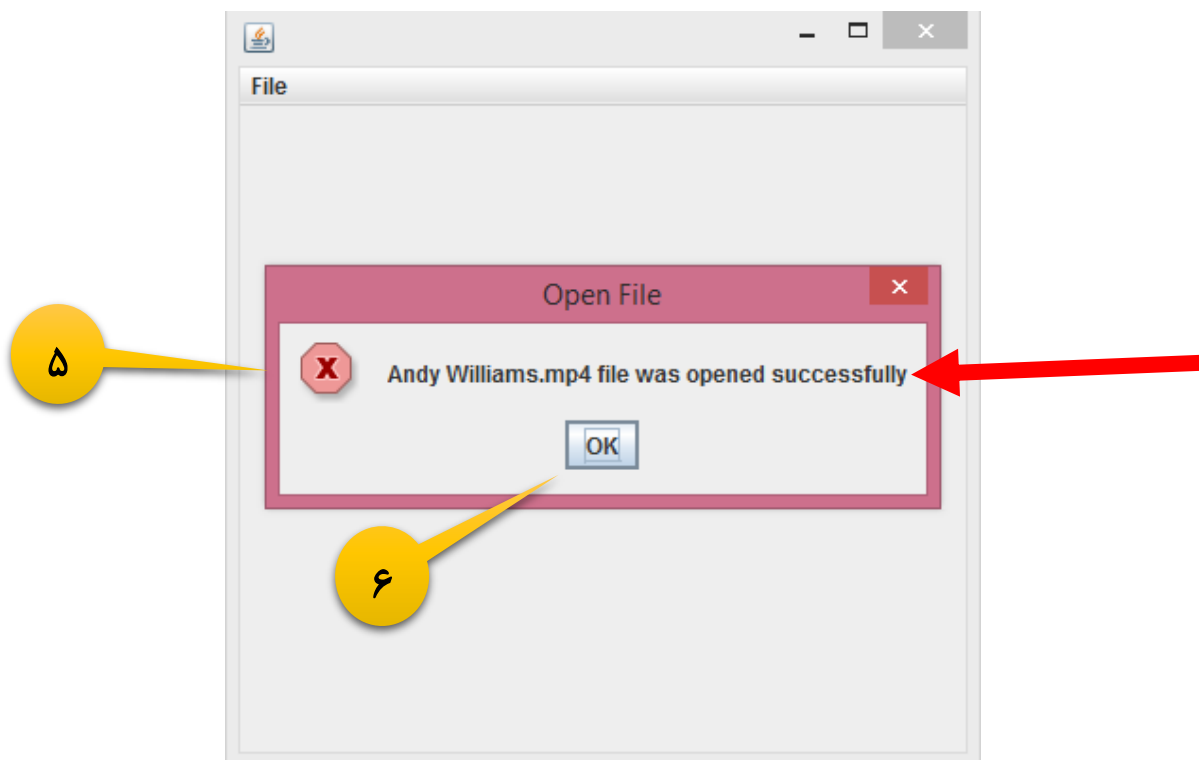
- این برنامه قادر است هر فایل با هر فرمتی را بخواند و در گوشه از کامپیوتر **save** کند. (چیزی شبیه کپی پیست کردن)



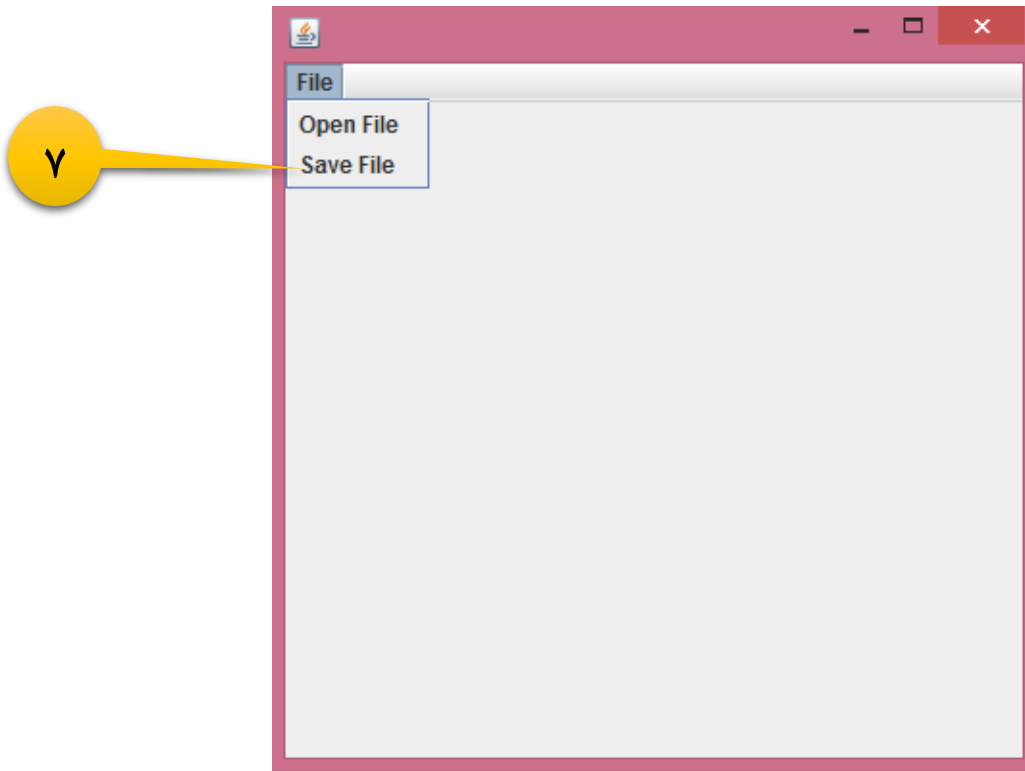
تصویر (۸)



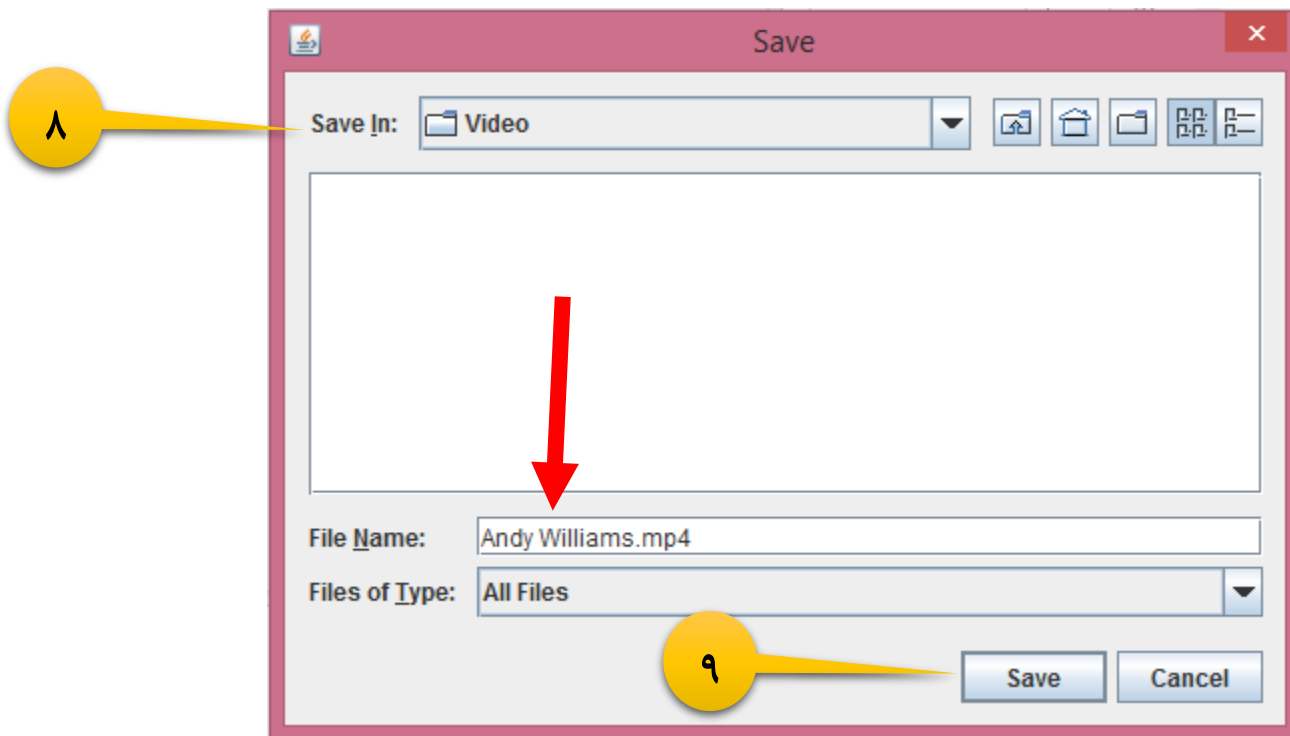
تصویر (۹)



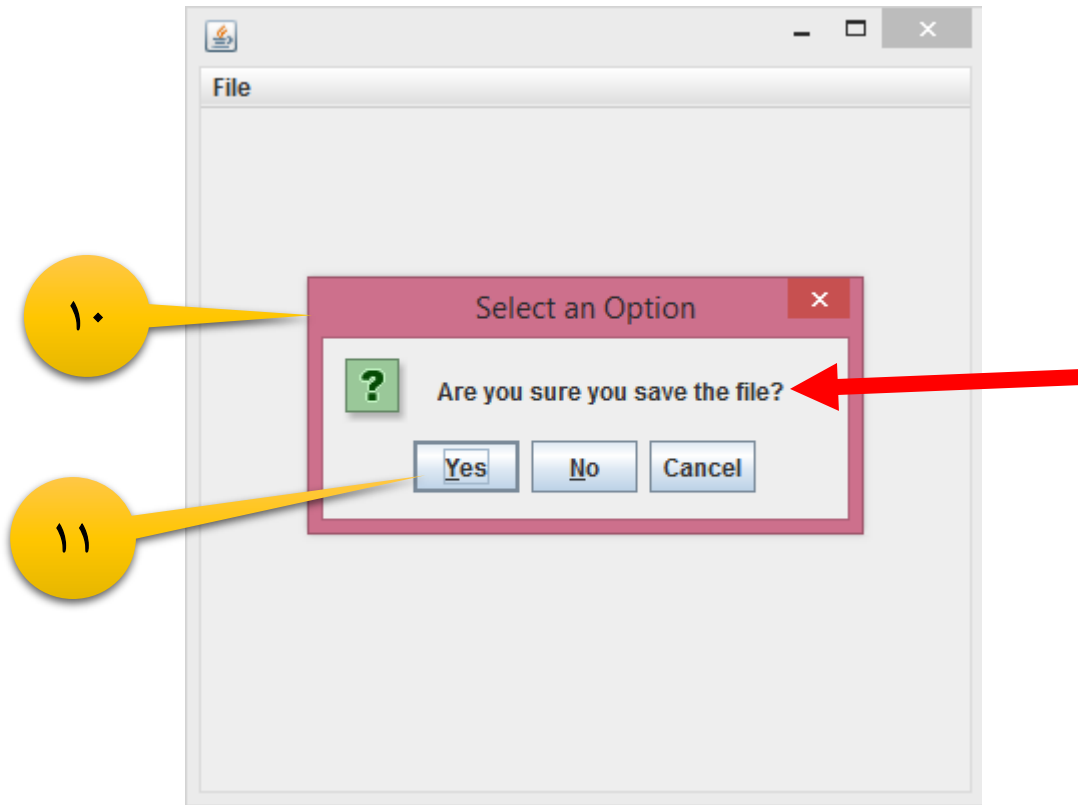
تصویر (۱۰)



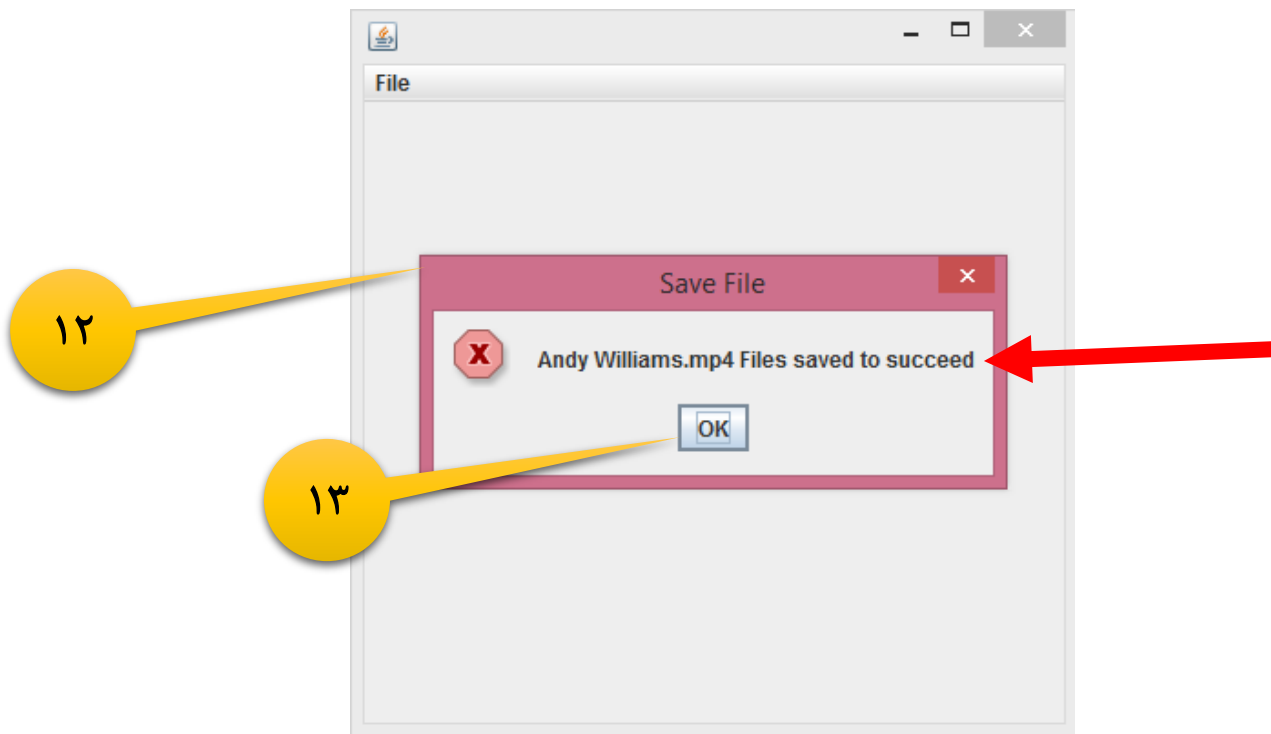
تصویر (۱۱)



تصویر (۱۲)



تصویر (۱۳)



تصویر (۱۴)

خب اینم از یک مثال کاربردی 😊 ، تمام دستورات درون کد بالا را در این جلسه و سایر جلسات بررسی کرده ایم.

پیروز و موفق باشید

سایت آموزش زبان جاوا به زبان ساده، آسان و شیرین!!!

www.JAVAPRO.ir

آموزش جاوا SE را با تجربه شخصی و به زبان خودمونی یاد بگیرید!!!!

بازدید از کانال

بازدید از سایت

هر روز مفاهیم و مثال های جدید به سایت اضافه می شود برای اطلاع از مطالب جدید روی سایت عضو کانال شوید.

دخل و تصرف ، ویرایش و کپی زدن تمامی آموزش های جاوا لایک به دور از اخلاق حرفه ای ست و حرام می باشد.