

## دستور کار کارگاه برنامه‌نویسی پیشرفته

نیم‌سال دوم ۹۷-۹۸

جلسه دهم

### آشنایی با گرافیک و GUI در جاوا

#### مقدمه

در این جلسه قصد داریم تا با استفاده از کتابخانه گرافیکی java swing برنامه‌ای بنویسیم که کاربران از طریق واسط گرافیکی با برنامه در ارتباط باشند. واسط کاربری در اصطلاح به هر وسیله‌ای گفته می‌شود که کاربران بوسیله‌ی آنها می‌توانند با برنامه در ارتباط باشند. در جلسات قبلی ارتباط کاربران با سامانه شما از طریق ترمینال بود. از این رو واژه الی تنها مختص واسط گرافیکی نیست گرچه ممکن است بصورت اشتباه از این واژه بجای GUI استفاده شود. GUI نوعی واسط کاربری است که در آن کاربر بوسیله‌ی اجزای گرافیکی با برنامه در ارتباط است. از این رو، این نکته بسیار اهمیت دارد که به عنوان یک برنامه نویس و توسعه گر، شما تشخیص دهید الی یک واژه عام است و GUI یکی از گونه‌های آن است.

برنامه‌ای که در این جلسه و جلسه‌ی بعدی قصد پیاده‌سازی آنرا داریم، یک برنامه چت گروهی است که کاربران این برنامه می‌توانند به یکدیگر پیام دهند. در این جلسه قرار است قسمت گرافیکی پروژه پیاده‌سازی شود و در جلسه بعدی قسمت منطق و شبکه‌ی آن پیاده‌سازی خواهد شد.

#### مدل توسعه MVC

یکی از مدل‌های رایج توسعه‌ی نرم‌افزار مدل MVC یا Model, View, Controller است. در این مدل هنگامی که می‌خواهیم یک برنامه را بسازیم، ابتدا آنرا به سه قسمت جداگانه تقسیم می‌کنیم که این سه قسمت عبارتند از:

- بخش رابطه کاربری برنامه که به اصطلاح View گفته می‌شود.
- بخش منطق برنامه که به اصطلاح Model گفته می‌شود.
- رابط بین این دو بخش که به آن Controller گفته می‌شود.

مدل MVC تنها روش توسعه برنامه‌ها نیست و روش‌های دیگری نیز هستند ولی حجم زیادی از برنامه‌ها از این چارچوب برای پیاده‌سازی برنامه‌های خود استفاده می‌کنند. مدل MVC این قابلیت را به توسعه دهندگان می‌دهد تا بتوانند بصورت مستقل قسمت Model و View خود را طراحی کنند و

آنها را بوسیله‌ی کنترلر به یکدیگر متصل کنند. این ویژگی قابلیت اعمال تغییرات در نرم‌افزار را بسیار افزایش می‌دهد که ویژگی مهمی در توسعه محصول بشمار می‌آید. برای پروژه درس نیز پیشنهاد می‌شود از این مدل برای پیاده‌سازی استفاده کنید و بهتر است خارج از این کلاس نیز درباره این چارچوب توسعه نرم‌افزار مطالعه داشته باشید.

## View

همانطور که گفته شد در این جلسه قصد داریم تا بخش View پروژه را پیاده‌سازی کنیم. قسمت View در پروژه نرم‌افزاری وظیفه‌ی ارتباط با کاربر را دارد و نباید منطق برنامه در آن پیاده‌سازی شود. کلاس‌هایی که در این قسمت طراحی می‌شوند وظیفه‌ی نمایش خروجی قسمت Controller را برعهده دارند.

## تعریف پروژه

در این پروژه قصد داریم تا یک برنامه چت گروهی بنویسیم که در آن افراد می‌توانند به یک اتاق گپ اصلی وصل شوند، برای خود یک نام استعاری انتخاب کنند و لیست افراد<sup>۱</sup> آنلاین را مشاهده کنند. سپس هر عضو گروه توانایی ارسال پیام دارد که این پیام به تمامی اعضای آن گپ ارسال می‌شود. هنگامی که پیام به دست عضوی از گروه رسید، پیام‌ها باید در قالب یک لیست به ترتیب نشان داده شود.

## پیاده‌سازی قسمت GUI

مراحل انجام کار:

ایجاد صفحه‌ی اصلی: برای ایجاد یک پنجره که در آن بتوان سایر اجزای GUI را در آن نمایش داد از کلاس JFrame استفاده می‌کنیم. برای آنکه طراحی زیباتری داشته باشیم که در آن بخواهیم جزئیات ایجاد یک پنجره را از کاربر مخفی کنیم بهتر است یک کلاس به نام ChatRoomGUI ساخته که از کلاس JFrame ارث‌بری می‌کند و در متد سازنده آن کدهای مربوط به ساختن پنجره و نمایش داده آنرا بنویسیم.

---

<sup>۱</sup>Chat room

<sup>۲</sup>Nick name

کد زیر کلاس ChatRoomGUI را پیاده‌سازی می‌کند.

```
import javax.swing.*;

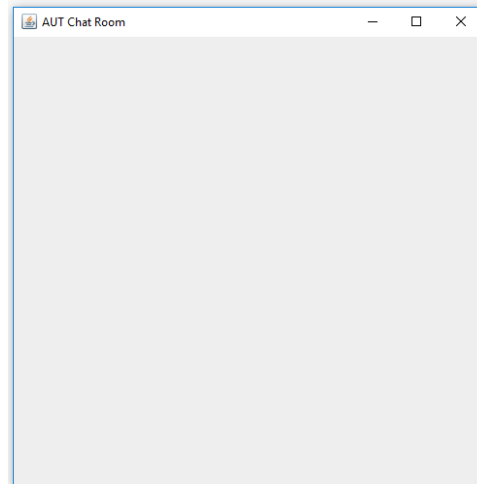
public class ChatRoomGUI extends JFrame{
    private final String WINDOWS_TITLE = "AUT Chat Room";
    private final int WIDTH = 500, HEIGHT = 500;
    private final int X = 100, Y = 100;

    public ChatRoomGUI() {
        super();
        this.setTitle(WINDOWS_TITLE);
        this.setLayout(null);
        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        this.setSize(WIDTH, HEIGHT);
        this.setLocation(X, Y);
        this.setVisible(true);
    }
}
```

متد setTitle, setSize و setLocation برای بیان ویژگی‌های پنجره هستند که از نام آنها می‌توان به عملکرد آنها پی برد. متد setLayout را در قسمت بعدی توضیح خواهیم داد. متد setDefaultCloseOperation برای آن است که بیان کند در صورتی که پنجره frame بسته شد برنامه چه کار کند. در حالت پیش‌فرض با بسته شدن پنجره برنامه خاتمه نمی‌یابد. در صورتی که ورودی تابع setDefaultCloseOperation را JFrame.EXIT\_ON\_CLOSE بگذاریم که یک Constant است، در صورت بسته شدن پنجره برنامه خاتمه می‌یابد. متد setVisible نیز برای نشان دادن صفحه بکار می‌رود.

## انجام دهید

یک پروژه جدید ساخته و کلاس ChatRoomGUI را به پروژه اضافه کنید. سپس یک کلاس Main ساخته که در آن یک شی از کلاس ChatRoomGUI را ایجاد کند. خروجی کد باید پنجره شکل ۱ را ایجاد کند.



شکل ۱ ایجاد پنجره با ابزار Java Swing

افزودن Layout: Layout ها نحوه‌ی چینش اجزای گرافیکی را در یک Frame مشخص می‌کنند. انواع مختلفی از Layout ها در swing تعریف شده است که هر کدام چینش خاصی به اجزای صفحه می‌دهند. Layout ها قابلیت responsive به پنجره می‌دهند بدین صورت که با تغییر اندازه پنجره، اجزا نیز با جای خود را تغییر می‌دهند. در صورتی که برنامه نویس نخواهد از Layout استفاده کند می‌تواند بطور دستی با توابع setPosition و setSize مکان اجزا را تعیین کند. برای این کار باید ورودی تابع setLayout را برابر null قرار داد تا swing اجازه‌ی مکان‌دهی به اجزا را بدهد. پیشنهاد می‌شود تا جای ممکن از Layout های موجود استفاده کنید. برای این تمرین از BorderLayout استفاده می‌کنیم. مستندات مربوط به BorderLayout را در لینک زیر بخوانید:

<https://docs.oracle.com/javase/tutorial/uiswing/layout/border.html>

بخش بندی مولفه ها: یکی از تمرینهای خوب برنامه نویسی در تولید رابطهای کاربری گرافیکی، بخش بندی مولفه های گرافیکی است. این بخش بندی معمولاً بر اساس کاربردها انجام می‌شود تا بتوان آنها را بهتر کنترل کرد. برای این کار از کلاس [JPanel](#) استفاده می‌کنیم. هر JPanel می‌تواند Layout مخصوص به خود را داشته باشد تا بتواند اجزای فرزند خود را در آن جا دهد.

از آنجایی که در BorderLayout برای هر قسمت آن (CENTER, PAGE\_TOP, PAGE\_END) تنها یک فرزند میتوان اضافه کرد از JPanel ها استفاده می‌کنیم تا بتوان محدودیت اضافه کردن فرزندان تو در تو را رفع کرد.

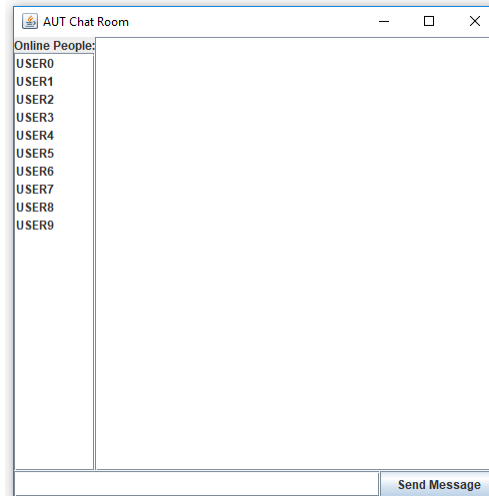
حال می‌خواهیم ابتدا یک پنجره درست کنیم که از کاربر یک نام کاربری برای ورود به گروه گپ بخواهد. برای این کار یک کلاس UsernameFrame درست کنید که از JFrame ارث بری می‌کند. کد زیر را در UsernameFrame وارد کنید:

```
import javax.swing.*;
import java.awt.*;

public class NameFrame extends JFrame {
    private static final String BTN_TXT = " Start Chatting ...";
    private static final String LABEL_TXT = " Choose Your UserName ";
    private static final int WIDTH = 300, HEIGHT = 100;
    JTextField textField;
    JButton btn;
    public NameFrame() throws HeadlessException {
        super();
        this.setLayout(new BorderLayout());
        JLabel label = new JLabel("Choose Your UserName");
        add(label, BorderLayout.PAGE_START);
        textField = new JTextField();
        add(textField, BorderLayout.CENTER);
        btn = new JButton(LABEL_TXT);
        add(btn, BorderLayout.PAGE_END);
        setSize(WIDTH, HEIGHT);
        setVisible(true);
    }
}
```

متد add در اجزای Swing برای آن است که بتوان فرزندی را به یک container دیگر اضافه کرد. در اینجا جز پدر پنجره است و جز فرزند اشیا ساخته شده از JLabel, JButton, و JTextField است.

حالا به سراغ قسمت اصلی که محیط اتاق گپ است می‌رویم. شکل زیر نمایی از صفحه اصلی است. البته پروژه شما نیازی ندارد که عینا شبیه این شکل باشد. این شکل صرفا برای نشان دادن اجزای اصلی این پنجره و وظیفه‌ی هر کدام است.



شکل ۲ پنجره اتاق گپ

اجزای این محیط عبارتند از:

- قسمتی که پیام‌های ارسالی و دریافتی را نشان می‌دهد که در شکل قسمت بزرگ پنجره است.
- قسمتی که برای نوشتن و ارسال پیام جدید استفاده می‌شود که در شکل نوار پایینی است. شامل یک TextBox و یک Button است.
- قسمتی که عضوهای آنلاین را نشان می‌دهد.

برای هر کدام از اجزای بالا کلاس مربوط به خودشان را به ترتیب با نام‌های زیر درست کنید

- ChatArea برای مدیریت قسمت اول
- MessageArea برای مدیریت قسمت دوم
- ParticipantsArea برای مدیریت قسمت سوم

ابتدا قسمت اول را پیاده‌سازی میکنیم. برای این کار باید از یک JTextArea استفاده کنیم که بتوان به آن متن اضافه کرد ولی قابلیت ادیت آن وجود نداشته باشد. قطعه کد زیر قسمتی از پیاده‌سازی این کلاس است.

```
public class ChatArea extends JTextArea {
    private static final int ROWS = 10, COLUMNS = 30;
    public ChatArea() {

        super(ROWS, COLUMNS);
        this.setEditable(false);
    }
}
```

```
this.setLineWrap(true);
}
}
```

## انجام دهید:

برای کلاس ChatArea متد addMessage را بنویسید که دو ورودی دارد ورودی اول نام کاربری است که پیام فرستاده و ورودی دوم آن پیام کاربر است. سپس این متد این دو ورودی را گرفته و نام کاربر و متن او را در TextBox نمایش می‌دهد. نحوه‌ی نمایش متن و نام کاربر به سلیقه‌ی خودتان است.

راهنمایی: از لینک زیر کمک بگیرید:

<https://www.javatpoint.com/java-jtextarea>

در گام بعدی باید قسمت ChatArea را به ChatRoomGUI اضافه کنید. برای این کار قطعه کد زیر را در ادامه کدهای ChatRoomGUI اضافه کنید. از آنجایی که در اکثر برنامه‌های چت گروهی قسمت مهم صفحه چت مشاهده پیام‌های گروه است، در این برنامه نیز باید قسمت مشاهده پیام را در قسمت مرکز قرار دهیم تا برای آن اهمیت بیشتری در نظر بگیریم. از این رو باید آنرا به قسمت CENTER اضافه کنیم.

```
ChatArea chatBox = new ChatArea();
this.add(new JScrollPane(chatBox), BorderLayout.CENTER);
```

## انجام دهید:

بر اساس نحوه چینش Component ها در BorderLayout، محل قرار گرفتن دو Panel دیگر را تعیین کنید.

مستندات مربوط به [JList](#), [JTextField](#), [JButton](#) را خوانده و با کمک آنها دو کلاس دیگر را پیاده سازی کنید و آنها را به ChatRoomGUI اضافه کنید. بخاطر اینکه MessageArea و ParticipantsArea هرکدام خود از اجزای دیگری تشکیل شدند، بهتر است از JPanel برای پیاده‌سازی این کلاس‌ها استفاده کنید و برای هر JPanel یک Layout تعریف کنید و اجزای فرزند آنرا به آن اضافه کنید.

متدی برای کلاس ParticipantsArea بنویسید که ورودی آن یک نام کاربری باشد و وظیفه‌ی آن افزودن آن نام کاربری به لیست افراد آنلاین باشد. میتوانید از لینک زیر کمک بگیرید:

[http://www.java2s.com/Tutorial/Java/0240\\_Swing/AddingandRemovinganIteminaJListComponent.htm](http://www.java2s.com/Tutorial/Java/0240_Swing/AddingandRemovinganIteminaJListComponent.htm)

سپس از این متدها استفاده کرده و برای کلاس ChatRoomGUI متدهای زیر را بنویسید:

۱. متد addNewMessage با دو ورودی نام کاربری و پیام.
  ۲. متد addNewParticipant که به عنوان ورودی یک نام کاربری دریافت میکند و آنرا به لیست کاربران آنلاین سیستم اضافه می‌کند.
  ۳. متد removeParticipant که به عنوان ورودی یک نام کاربری می‌بیند و کاربر مربوطه را از لیست کاربران آنلاین حذف می‌کند.
- در جلسه بعدی که به پیاده‌سازی منطق این پروژه می‌پردازیم، از این متدها استفاده کرده و بین منطق کد و رابط گرافیکی رابطه برقرار می‌کنیم.