

# Messenger Application

از شما خواسته شده تا برنامه پیام رسانی طراحی کنید که کاربران با اتصال به آن، امکان تبادل پیام و ارتباط با یکدیگر را داشته باشند.

## بخش اول

**فاز اول:** در بخش اول قصد داریم یک برنامه سازی شبکه (Socket Programming) به دو کاربر آلیس و باب امکان ارسال پیام به سرور را بدهیم. به این صورت که پیام رسانی server می باشد و آلیس و باب client برای سرور هستند که با اتصال به آن امکان ارسال پیام و دریافت جواب از او دارند. برای پیاده سازی نیاز به کلاس های Server.java، Alice.java و Bob.java داریم.

ابتدا کلاس های Server.java و Alice.java را به گونه ای تعریف کنید که کلاینت (آلیس) به سرور وصل شود و پس از اتصال، سرور به آلیس پیام "Enter name:" را ارسال کند. آلیس با دریافت این پیام و چاپ آن رشته "Alice" را وارد می کند و این رشته به سرور ارسال می شود. در مرحله بعد نیز به همان شکل سرور از آلیس رمز او را می پرسد و آلیس در جواب پیامش، رمز را وارد می کند و برای سرور ارسال می کند.

سرور پس از دریافت اسم و رمز آلیس، آنها را در آرایه ای از جنس User (که صرفا دارای ویژگی های نام و رمز میباشد) ذخیره می کند. پس از آنکه اطلاعات آلیس در این آرایه ذخیره شد، سرور منتظر دریافت رشته از آلیس است و او هر پیامی که به سرور ارسال کند، سرور همان متن را به صورت اکو ارسال می کند. برای مثال اگر آلیس رشته "Hi!" را وارد کند، سرور این رشته را دریافت کرده و عبارت "Echo: Hi!" را برای آلیس ارسال می کند و آلیس آن را چاپ می کند.

**فاز دوم:** اگر آلیس با سرور در ارتباط باشد اما ناگهان باب خواستار اتصال به سرور باشد، سرور نمی تواند به صورت موازی با آلیس و باب ارتباط برقرار کند. در ادامه می خواهیم با استفاده از Multi-Thread Programming (برنامه نویسی چند نخی) امکان اتصال همزمان باب و آلیس (و حتی تعدادی بیشتر کاربر) به این پیام رسان را فراهم کنیم.

برای این کار کافیهست هر زمان کلاینتی درخواست اتصال به سرور داد، سوکتی از طرف سرور برای او اختصاص یابد که این کار با اختصاص دادن یک ترد به هر کلاینت ممکن می شود. برای مثال، آلیس به سرور درخواست اتصال می دهد و سرور تردی به وجود می آورد که با آن سوکت مخصوص آلیس را مدیریت می کند. سپس هنگامی که باب درخواست اتصال به سرور بدهد، سرور با مشاهده این درخواست، ترد دیگری به باب اختصاص می دهد تا عملیات او را انجام دهد و همزمان ارتباط خود با آلیس را حفظ کند. (راهنمایی: اضافه کردن یک کلاس ClientHandler.java که از Runnable ارث بری می کند)

## نمره اضافه:

- اعمال Synchronization برای مدیریت بخش های بحرانی (در صورت نبود بخش بحرانی آن را با یک متغیر مشترک به وجود آورید).
- داشتن گرافیک با JavaFX