

به نام خدا

پروژه درس برنامه‌نویسی پیشرفته

پاییز ۱۴۰۰

۱. تمامی فایل‌های کد را به همراه فایل‌های تنظیمات به صورت یک فایل آرشیو (zip != rar) که به قالب زیر نام‌گذاری شده است، بارگذاری نمایید.

AP-MTP-FirstName_LastName-StudentNumber.zip

مثال : AP-MTP-Parham_Ahmadi -9831071.zip

۲. در صورت مشاهده هرگونه تقلبی، طبق موارد گفته شده در قوانین درس برخورد خواهد شد.

۳. در صورت وجود هرگونه ابهام می‌توانید از طریق گروه تلگرامی با تدریس‌یاران در ارتباط باشید.

مهلت تحویل: تا ۶ بهمن ۱۴۰۰ ساعت ۲۳:۵۵ شب

در صورتی که زمان نهایی شدن نمرات قبل از مهلت ارسال باشد مجبور به تحویل زودهنگام خواهیم بود

پیش‌گفتار	2
تعریف پروژه	5
شبکه اجتماعی توییتر (Twitter)	6
توضیحات اولیه	6
فاز چهارم: پیاده‌سازی رابط کاربری گرافیکی	7
صفحه ورود و ثبت‌نام	7
صفحه Home یا Timeline	8
صفحه ایجاد توییت	9
صفحه پروفایل کاربر	10
نوار بالایی برنامه	11
نوار کناری صفحه	11
کلیدهای میان‌بر	12
نکات پیاده‌سازی	13
ساختار فولدرهای پروژه	14

پیش‌گفتار

استفاده مناسب از مفاهیم تدریس شده ضروری است. طراحی خوب و منطقی کلاس‌ها و اینترفیس‌ها باید متناسب با اصول برنامه‌نویسی شی‌گرا باشد. رعایت اصول پنهان‌سازی اطلاعات (information hiding)، سلسله مراتب ارث‌بری جهت استفاده مجدد از کدها (code reusability)، استفاده از چندریختی و سایر نکات تدریس شده الزامی است. سعی کنید پیش از شروع پیاده‌سازی، تعداد و نام کلاس‌ها، فیلدها و متدهای مورد نظر برنامه را تحلیل کرده و مطابق با تحلیل و طراحی انجام‌شده، برنامه را پیاده‌سازی کنید.

مستندسازی به کمک Javadoc، کامنت‌گذاری و رعایت اصول کد نویسی خوانا برای همه کلاس‌های پیاده‌سازی شده الزامی است.

برنامه را پیش از بارگذاری به خوبی تست و اشکال‌زدایی کنید! همه کلاس‌ها و متدهای موجود در برنامه‌ها کاملاً مورد بررسی و آزمون قرار بگیرند تا از درستی عملکرد برنامه اطمینان حاصل کنید. حالت‌های مختلف ورودی توسط کاربر باید بررسی شود و در صورت لزوم، پیغام خطای مناسب نمایش داده شود.

در قسمت‌های مختلف پروژه باید خطاهای مختلف بررسی شوند و در قبال آن رفتار مناسبی از برنامه دریافت شود. پس شما باید برای تمامی قسمت‌ها عملیات Exception Handling را برای استثنای Checked و همچنین در مواقع لزوم برای Unchecked پیاده‌سازی کنید.

در این پروژه شما می‌بایست کد خود را در گیت نیز قرار دهید. توجه شود صرف قرارگیری و آپلود پروژه در گیت کفایت نمی‌کند و شما باید کامیت‌های مناسب و مستمر داشته باشید.

به عنوان مثال اگر ویژگی جدیدی به کد قبل اضافه کرده‌اید می‌توانید مانند زیر پیام کامیت را مشخص کنید:

[feature] users can follow eachother

یا اگر ایرادی را برطرف کرده‌اید:

[bug] client can't connect to server

یا اگر اصلاحات یا اضافاتی بر ویژگی‌های قبلی داشته‌اید:

[enhancement] speed of tweet posting optimized

روش‌هایی که برای کامیت ذکر شده‌اند کاملاً قراردادی بوده و شما از هر روش خوانای دیگری می‌توانید استفاده کنید.

لطفاً از ساخت مخزن (**Repository**) به صورت عمومی (**Public**) قبل از تمام شدن مهلت ارسال خودداری کنید

پروژه دارای تحویل آنلاین (از طریق اسکایپ) است و تسلط بر آن تأثیر زیادی در نمره شما دارد.

تعریف پروژه

پروژه پیش‌رو شامل 4 بخش اساسی است که به ترتیب مباحث تدریس شده، تعریف شده است. طبیعتاً تمام مباحث این پروژه تا این لحظه به شما تدریس نشده است، اما برنامه‌ریزی شده با توجه به برنامه درس، ددلاین‌های فازهای پروژه عملیاتی شوند.

توجه داشته باشید که این پروژه برای تمام ترم است و پروژه پایان‌ترم مجزا تعریف نخواهد شد

در تعریف این پروژه هر رنگ در متن (و نه در عکس‌ها) معنی خاص خود را دارند.

رنگ **نارنجی تیره** به معنی پیاده‌سازی اجباری است.

رنگ **سبز** به معنی پیاده‌سازی امتیازی است.

رنگ **آبی کم‌رنگ** به معنی توضیحات بیشتر است.

این پروژه 3 فاز دارد که در ادامه توضیحات هر فاز آمده است.

در صورت وجود هرگونه ابهام در تعریف پروژه می‌توانید با تدریس‌یارها در ارتباط باشید.

شبکه اجتماعی توییتر (Twitter)



در این پروژه ما سعی در پیاده سازی شبکه اجتماعی توییتر با رابط کاربری گرافیکی را داریم.

توضیحات اولیه

توییت (به انگلیسی: Twitter) یک شبکه اجتماعی و سرویس ارائه دهنده میکرو بلاگ است که به کاربران اجازه می‌دهد تا ۲۵۶ حرف، پیام متنی را که **توییت** (به انگلیسی: Tweet) نامیده می‌شود، ارسال کنند. کاربرانی که در توییتر ثبت نام کرده‌اند می‌توانند توییت ارسال کنند، به توییت‌ها جواب بدهند، یک توییت را دوباره بازنشر کنند (به انگلیسی: Retweet) و همچنین آنها را لایک کنند. به علاوه موارد ذکر شده کاربرانی که ثبت نام شده‌اند می‌توانند کاربران دیگر را دنبال کنند و توییت‌های آنان را نیز مشاهده کنند، به یکدیگر پیام متنی ارسال کرده و با هم صحبت کنند.

فاز چهارم: پیاده‌سازی رابط کاربری گرافیکی

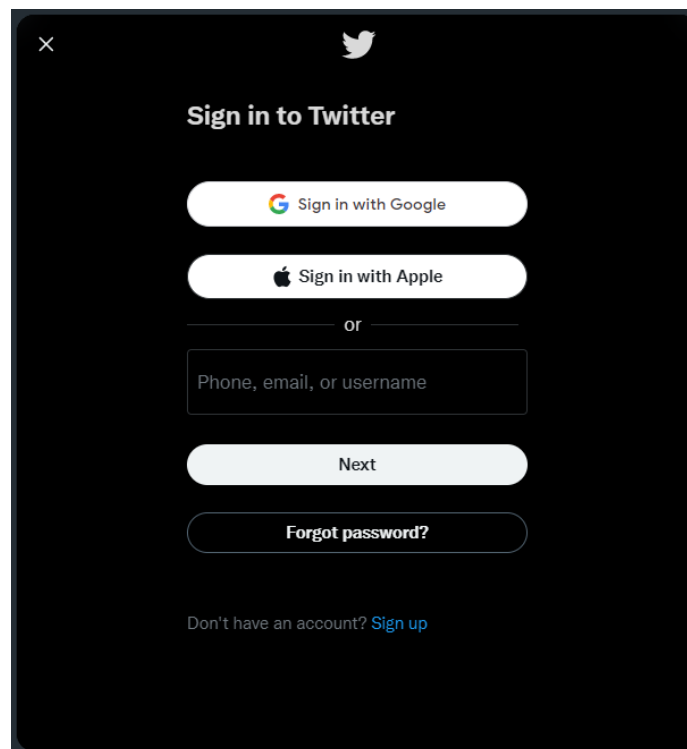
در این فاز قصد داریم تا برای برنامه‌ای که در سه فاز گذشته پیاده‌سازی کرده‌ایم، با استفاده از ابزار Java-Fx یک رابط کاربری گرافیکی طراحی و پیاده‌سازی کنیم.

توجه داشته باشید که رابط کاربری گرافیکی مورد نظر صرفاً برای برنامه کاربر (Client) خواهد بود.

صفحه ورود و ثبت‌نام

در ابتدای اجرای برنامه باید صفحه‌ای نمایش داده شود که در آن کاربر با وارد کردن نام کاربری و رمز عبور بتواند به صفحه Timeline خود وارد شود. همچنین در این صفحه باید گزینه‌ای وجود داشته باشد که صفحه ثبت‌نام را نمایش دهد و کاربرانی که قصد ثبت‌نام در برنامه را دارند باید مشخصات خود را در این صفحه وارد و درخواست ثبت‌نام بدهند، در صورت موفقیت‌آمیز بودن ثبت‌نام، صفحه ورود دوباره نمایش داده شود. همچنین در صفحه ثبت‌نام باید گزینه‌ای به نام انصراف وجود داشته باشد که با استفاده از دوباره به صفحه ورود برگردانده شود.

صفحه ورود باید شامل گزینه Remember me باشد. در صورتی که این گزینه فعال باشد، در نوبت بعدی اجرای برنامه صفحه ورود نمایش داده نمی‌شود و مستقیماً به صفحه Timeline وارد می‌شود.



صفحه Home یا TimeLine

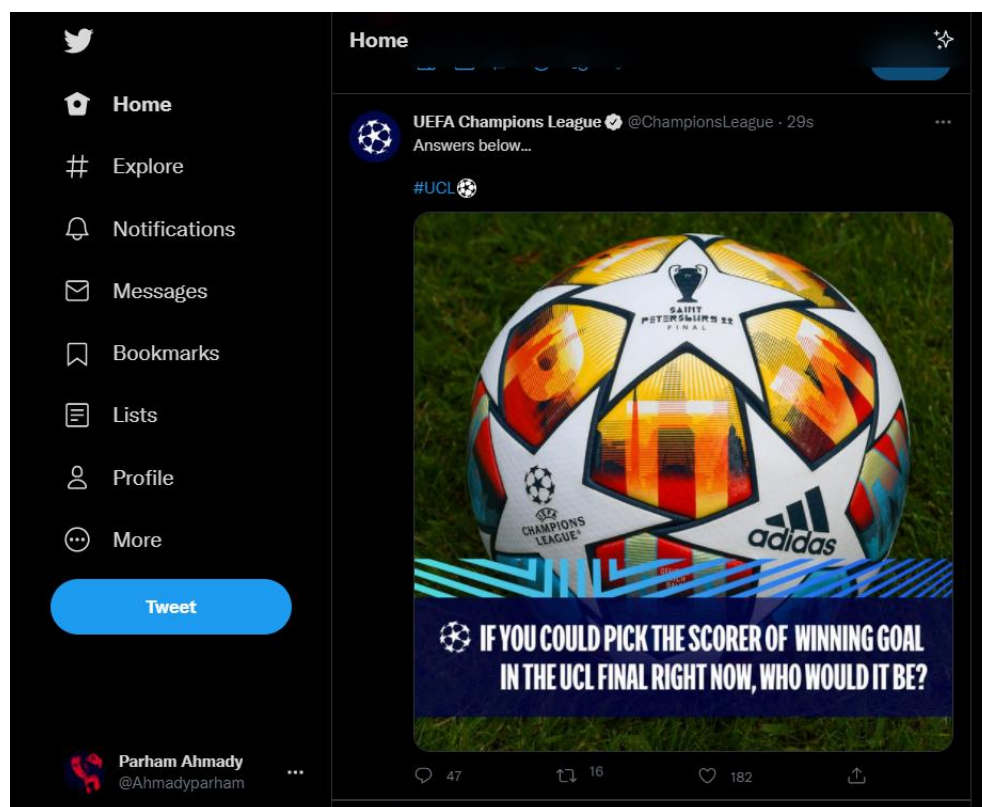
این صفحه بعد از ورود کاربر نمایش داده می‌شود. در این صفحه کاربر باید بتواند توییت‌های کاربرانی را که دنبال کرده‌است و همچنین توییت‌های خود را به ترتیب زمان انتشار مشاهده کند. مشخصات توییت‌های نمایش داده شده باید به گونه‌ای باشد که تاریخ ارسال، نام کاربری فرستنده، محتوا، تعداد لایک‌ها در آن دیده شود. همچنین نام کاربری باید به گونه‌ای باشد که با کلیک بر روی آن صفحه پروفایل کاربری آن کاربر را نمایش دهد. برای توییت‌هایی که توسط خود کاربر فرستاده شده نیز باید گزینه حذف وجود داشته باشد. و در آخر نیز باید دکمه‌ای برای بازنشر توییت وجود داشته باشد (این گزینه برای توییت‌های سایر کاربران است).

باید دکمه‌ای وجود داشته باشد که به کمک آن بتوان TimeLine را بروز کرد (Refresh).

توجه! پس از بروز کردن TimeLine، برنامه نباید به هیچ عنوان به حالت Freeze برود.

قابلیت پاسخ دادن به یک توییت و نمایش پاسخ‌های یک توییت.

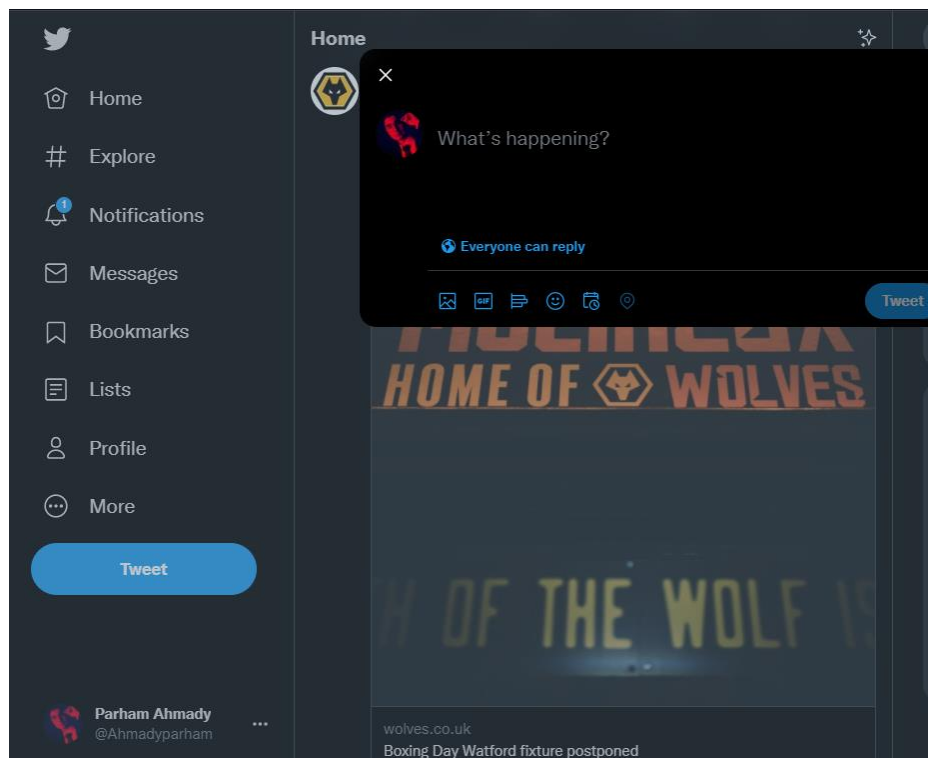
برای دریافت نمره امتیازی ویژگی‌های فوق باید از سرویس‌هایی که در فازهای قبل پیاده‌سازی کرده‌اید استفاده کنید و در صورت تعریف جدید نصف نمره را دریافت خواهید کرد.



صفحه ایجاد توییت

در این صفحه کاربر باید بتواند یک توییت ایجاد کند، در این صفحه باید یک دکمه ارسال قرار داده شود که با استفاده از آن توییت ارسال شده و برنامه به صفحه **Timeline** برمی‌گردد و یک دکمه نیز برای انصراف از ارسال توییت وجود داشته باشد.

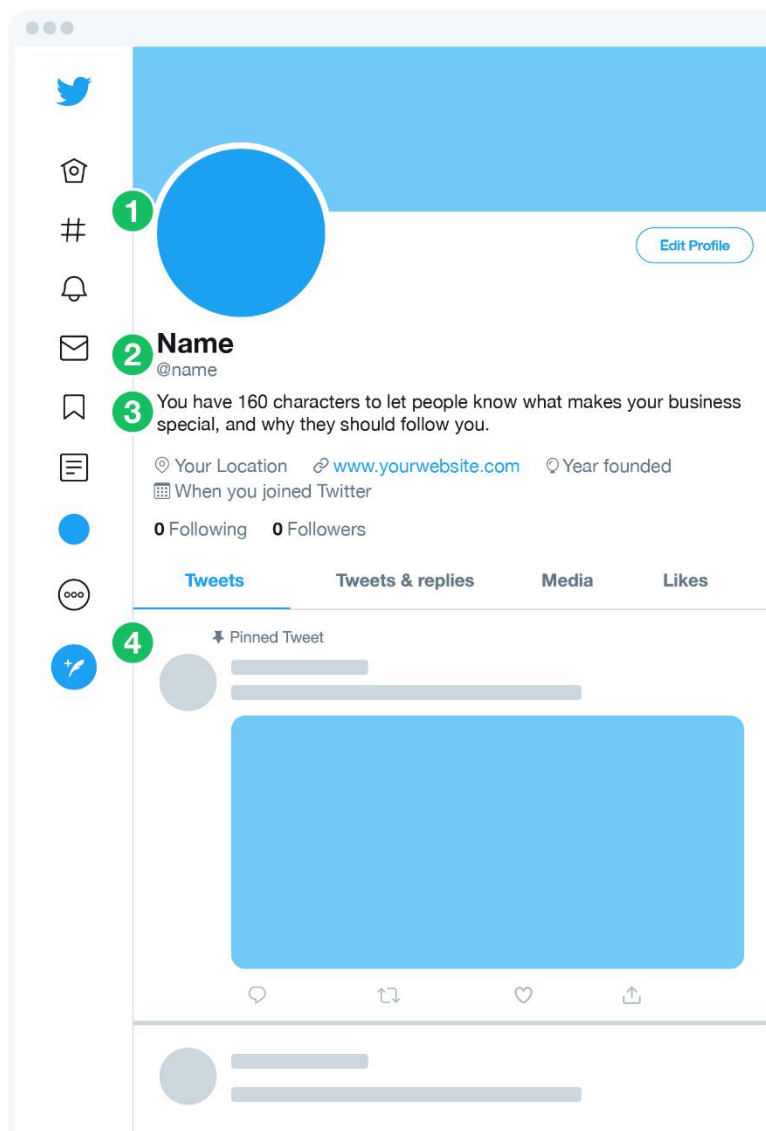
در صورت بروز خطا در هریک از صفحات مشخص شده باید پیغام مناسب به کاربر داده شود.



صفحه پروفایل کاربر

در این صفحه باید اطلاعات کاربر مانند نام کاربری، عکس پروفایل، بیوگرافی، توییت‌ها و تاریخ عضویت نمایش داده شود. همچنین باید گزینه‌ای وجود داشته باشد که به کمک آن بشود آن کاربر را **Follow** یا **Unfollow** کرد. همچنین اگر صفحه نمایش داده شده مربوط به خود کاربری باشد که در برنامه وارد شده است، کاربر با کلیک بر روی عکس پروفایل باید بتواند آن را عوض کند.

عکس پروفایل در فازهای قبلی پیاده‌سازی نشده است و شما باید این ویژگی را به موجودیت‌ها و سرویس‌هایی که تر فازهای پیشین پیاده‌سازی کرده‌اید، بیافزایید.



1- عکس پروفایل

2- نام کاربری

3- بیوگرافی

4- توییت‌های کاربر

نوار بالایی برنامه

این نوار دارای 3 منوی Help , View و Application و Options می‌باشد.

منوی Application دارای دو گزینه Exit و Log Out می‌باشد. با کلیک بر روی گزینه Exit برنامه عملیات خروجی که برایش تعریف شده (Exit / Hide on system tray) را اجرا می‌کند.

منو Help شامل 2 گزینه About و Help است. با کلیک بر روی گزینه About ، اطلاعات شما به عنوان توسعه‌دهنده مانند ایمیل و شماره دانشجویی در یک پنجره نمایش داده می‌شود. با کلیک بر روی گزینه Help پنجره‌ای باز می‌شود که در آن هر یک از دکمه‌های موجود در برنامه را توضیح داده شده است.

منو View شامل گزینه Toggle Full Screen است. با کلیک بر روی گزینه Toggle Full Screen ، برنامه به صورت full screen در می‌آید و با کلیک دوباره، برنامه به حالت قبلی برمی‌گردد.

در همین منو قسمتی وجود دارد که شما با انتخاب از یک لیست پیش فرض می‌توانید رنگ‌بندی رابط گرافیکی را از بین light theme و dark theme انتخاب کنید. رنگ‌بندی شامل رنگ پس‌زمینه و دکمه‌ها می‌باشد.

با کلیک روی گزینه Options یک پنجره باز می‌شود که با استفاده از آن می‌توانید مشخص کنید برنامه هنگام خروج، به طور کلی بسته شود یا در System Tray مخفی شود.

توجه کنید تمام این تنظیمات باید در یک فایل ذخیره شود و بعد از اجرای نرم افزار، آخرین تنظیمات ذخیره شده اعمال شود.

برای تمام منوها و اجزای آن‌ها، باید پیاده‌سازی Accelerator و Mnemonic انجام شود. برای آشنایی می‌توانید به این [لینک](#) مراجعه کنید. به عنوان نمونه با استفاده از کلیدهای Ctrl + E برنامه بسته شود و یا با استفاده از کلیدهای Alt + V منوی View به نمایش درآید.

نوار کناری صفحه

در سمت چپ صفحه باید نوازی وجود داشته باشد که شامل گزینه‌هایی برای نمایش دادن صفحه ایجاد توییت، پروفایل کاربر وارد شده، Chat Room و صفحه TimeLine می‌باشد.

کلیدهای میان‌بر

باید کلیدهای میان‌بری طراحی کنید که با استفاده از آن:

- صفحه ایجاد توییت نمایش داده شود.
- صفحه **TimeLine** بروزرسانی شود. به عنوان مثال با فشردن کلید **F5** صفحه **TimeLine** بروز شود.
- صفحه پروفایل کاربر نمایش داده شود.
- بتوان در صفحه **TimeLine** به سمت بالا و پایین حرکت کرد (**Scrolling**). به عنوان مثال با استفاده از کلیدهای جهت‌دار (**Arrow Keys**) در صفحه **TimeLine** به بالا یا پایین حرکت کرد.
- **Toggle Side Bar**. به عنوان مثال اگر کاربر از **Toggle Side Bar** استفاده کند، نوار سمت چپ تصویر بین حالت‌های **Minimize** و **Maximize** تغییر حالت می‌دهد.

تصاویری که در این پروژه مورد استفاده قرار گرفته‌اند، صرفاً برای آشنایی شما با برنامه است، و پیاده‌سازی جزئیات درج شده در هر تصویر مورد نیاز نیست، و فقط مواردی که در متن ذکر شده‌اند باید طراحی و پیاده‌سازی شوند.

نکات پیاده‌سازی

برای پیاده‌سازی این سامانه ملزم به استفاده از سرویس‌هایی هستید که در 3 فاز گذشته پیاده‌سازی کرده‌اید. اما می‌توانید در آن‌ها تغییراتی ایجاد کنید. به عنوان مثال برای ارسال توییت باید از همان تابعی در سمت سرور استفاده کنید که در فاز 3 استفاده می‌کردید. اما برای تغییر عکس پروفایل کاربر می‌توانید متد جدید یا سرویس جدیدی پیاده‌سازی بکنید.

در فاز 3، سرویسی به نام **Command Parser** پیاده‌سازی شده است که در این فاز نیازی به استفاده از آن نیست و می‌توانید برای مرتب‌سازی پروژه خود سرویس جدیدی تعریف کنید.

ساختار فولدرهای پروژه

- هر کلاسی که نقش سرویس را ایفا می‌کند باید ابتدا متدهای آن به صورت **abstract** و در یک **interface** تعریف شود و سپس پیاده سازی آن در پکیجی به نام **impl** قرار بگیرد.
 - این کار به این منظور است که وظایف یک سرویس کاملاً مشخص است ولی پیاده سازی در ورژن‌های مختلفی از برنامه می‌تواند تغییر کند.
- نحوه اسم‌گذاری: اگر **interface** داشته باشیم به اسم **TestController**، کلاسی که آن را پیاده‌سازی می‌کند باید نام کلاس **TestControllerImpl** باشد.
- اگر در قسمت‌های دیگر پروژه از **interface** یا **abstract class** استفاده کرده‌اید باید مانند مثال‌های قبل عمل کنید. (نباید **interface** و کلاس پیاده سازی آن دقیقاً در یک **package** باشند).
- باید تمامی تنظیمات قابل تغییر مانند درگاه (Port) سرور، محل ذخیره شدن فایل‌ها و ... را چه در برنامه سرور و چه در کلاینت از فایل تنظیمات (**Properties**) بخوانید. در این خصوص ویدیوی آموزشی بارگذاری شده است.
- اگر از عکس خاصی برای زیبایی بخشی به رابط کاربری گرافیکی استفاده می‌کنید، به هیچ عنوان نباید آدرس فایل آن عکس را به صورت **Hardcode** در برنامه بنویسید، بلکه باید بتوان آدرس را در فایل تنظیمات (**Properties**) تعیین کرد. همچنین از آپلود عکس‌ها در سامانه کورسز خودداری کنید.

نکات فوق اصول اولیه طراحی با زبان جاوا هستند و رعایت آن‌ها الزامی است