



## مهندسی نرم افزار ۱

مدرس: رامتین خسروی

بهار ۱۴۰۳

تکلیف سه

### آزمون واحد

هدف این تکلیف تمرین روی طراحی و پیاده سازی آزمون واحد است. نسخه ای از پروژه TinyME که در اختیار شما قرار گرفته، فاقد آزمون برای نیازمندی های مرتبط با اعتبار کارگزاری ها است. لازم است تعدادی آزمایش (test case) برای اطمینان از درستی این نیازمندی ها طراحی و پیاده سازی کنید.

### نوشتن کدهای آزمون

با توجه به منطقی که در [سند توصیف دامنه پروژه](#) در بخش Handling Broker's Credit ذکر شده، آزمایش هایی برای کنترل و به روز رسانی اعتبار کارگزاران در درخواست های سفارش جدید، اصلاح سفارش و حذف سفارش بنویسید.

به این منظور یک کلاس آزمون به نام `ir.ramtung.tinyme.domain.BrokerCreditTest` ایجاد کنید و آزمایش های خود را در قالب متدهای آن بنویسید. برای آشنایی با نحوه تعریف آزمایش ها، استفاده از AssertJ و Mockito می توانید کدهای آزمون موجود در پروژه را ببینید.

🤔 از آنجا که منطق کنترل و به روز رسانی اعتبار در متدهای مختلفی از کلاس های برنامه پخش شده است، از یک نگاه بهتر است آزمایش های شما در کلاس های آزمون موجود (مثل `SecurityTest` یا `MatcherTest`) قرار بگیرد. اما از طرف دیگر چون موضوع همه آزمایش ها مرتبط با اعتبار است شاید داشتن یک کلاس مجزا (مثل `BrokerCreditTest`) بهتر باشد. به نظرتان کدام راه حل بهتر است؟

مستقل از این که در این مورد به چه نتیجه ای می رسید آزمایش ها را در `BrokerCreditTest` پیاده سازی کنید تا شیوه ارزیابی پاسخ شما یکسان باشد.

## نحوه ارزیابی

برای ارزیابی پاسخ تکلیف، تغییراتی را روی کلاس‌های متن برنامه ایجاد خواهیم کرد. به این ترتیب تعدادی نسخه تغییر یافته از برنامه خواهیم داشت که هر کدام یک خطای (معقول) برنامه‌نویسی را شبیه‌سازی می‌کند. سپس آزمون‌های شما را روی هر یک از نسخه‌های تغییر یافته اجرا می‌کنیم و انتظار داریم دست کم یکی از آزمایش‌های شما شکست بخورد (fail کند)<sup>1</sup>. **دقت نمایید فرض ما بر این است که شما کلاس‌های موجود در متن برنامه را تغییر نمی‌دهید.**

علاوه بر مورد بالا که مؤثر بودن آزمایش‌های شما را می‌سنجند، تمیزی کد آزمون از عوامل ارزیابی پاسخ شماست. به این منظور به نکات زیر دقت کنید.

- هر آزمایش یک هدف داشته باشد، یعنی وجود یک خطا را بیازماید
- نام آزمایش‌ها را طوری انتخاب کنید که هدف آن را به طور واضح بیان کند
- تا حد امکان از داشتن کد تکراری پرهیز کنید.

توجه داشته باشید کد آزمون‌های موجود در پروژه از نظر تمیزی بدون نقص نیستند.

## نحوه تحویل

برای تحویل تکلیف، فقط فایل BrokerCreditTest.java را در ای‌لرن آپلود نمایید. دقت نمایید این تکلیف تحویل حضوری خواهد داشت.

⚠️ طی تحویل حضوری، علاوه بر بحث و ارائه بازخورد درباره پاسخ شما، درباره نحوه عملکرد بخش‌های مختلف کد اصلی پروژه (نه فقط پاسخ شما) از شما سؤال‌هایی می‌پرسیم که مطمئن شویم کد پروژه را به طور کامل فهمیده‌اید.

---

<sup>1</sup> این روش ارزیابی مجموعه آزمایش‌ها را [Mutation Testing](#) می‌نامند.