



## تمرین کامپیوتری شماره ۲ آشنایی با بدیل‌های آزمون (Test Doubles)



آزمون نرم‌افزار - پاییز 1401

مهلت تحویل:

دانشکده مهندسی برق و کامپیوتر

طراح تمرین: **سارینا همدانی**،  
**نازنین یوسفیان**

پایان روز 5 آذر

استاد: دکتر احسان خامس‌پناه

در این پروژه می‌خواهیم با استفاده از کتابخانه‌ی Mockito با انواع مختلف بدیل‌های آزمون و رویکردهای مختلف صحت‌سنجی (Verification) آشنا شویم.

می‌دانیم الگوهای مختلفی برای صحت‌سنجی وجود دارد. از جمله: State Verification و Behavior Verification. برای مطالعه بیشتر می‌توانید به فصل ۱۱ در [این کتاب](#) مراجعه نمایید.

از طرف دیگر دو رویکرد برای طراحی و پیاده‌سازی آزمون وجود دارد: Mockist و Classical. برای مطالعه بیشتر می‌توانید به [مقاله فاولر](#) در این زمینه مراجعه نمایید.

**بخش اول (گزارش کار):** به تست‌های زیر که برای کلاس StudyRecord نوشته شده‌اند، دقت کنید. برای هر کدام با ذکر دلیل، نوع همه Double‌های مورد استفاده و همچنین رویکرد پیاده‌سازی را مشخص کنید.

```
public class StudyRecordTest {  
    private StudyRecord studyRecord;  
  
    @Mock  
    private Course course;  
  
    private String term = "00001";  
    private double grade = 16.5;
```

```
@BeforeEach
void setup() {
    MockitoAnnotations.initMocks(this);
    studyRecord = new StudyRecord(term, course, grade);
}

@AfterEach
void teardown() {
    studyRecord = null;
}

@Test
public void testWeightedScore() {
    when(course.getCredits()).thenReturn(3);
    assertEquals(grade * 3, studyRecord.weightedScore());
}

@Test
public void testIsPassedCallsGraduateMinValidGrade() {
    GraduateLevel graduateLevel = mock(GraduateLevel.class);
    studyRecord.isPassed(graduateLevel);
    verify(graduateLevel).getMinValidGrade();
}

@Test
public void testIsPassed() {
    GraduateLevel graduateLevel = mock(GraduateLevel.class);
    when(graduateLevel.getMinValidGrade()).thenReturn(10);
    when(course.getGraduateLevel()).thenReturn(graduateLevel);
    assertTrue(studyRecord.isPassed(graduateLevel));
}
}
```

**بخش دوم:** کلاس StudentController را تست نمایید. در پیاده‌سازی تست‌های این قسمت، موارد زیر را در کدتان کامنت کنید:

- با توجه به تعاریف ارائه شده در درس، هر کدام از Double های مورد استفاده از چه نوع هستند؟

Dummy Object, Stub, Spy, Mock, Fake Object

- هر آزمون چه نوع صحت‌سنجی انجام می‌دهد؟
- هر آزمون با چه رویکردی پیاده‌سازی شده‌است؟

**بخش سوم (گزارش کار):** تابع checkLoop در کلاس AddCourseService را در نظر بگیرید. به نظر شما کدام یک از دو الگوی State Verification یا Behavior Verification برای آزمون آن مناسب‌تر است؟ پس از انتخاب الگوی موردنظر تست متناظر آن را پیاده‌سازی کنید.

### نکات تکمیلی

- پروژه در قالب گروه‌های دو نفره انجام شود.
- برای انجام پیاده‌سازی‌های خود **مخزن** را fork کرده و تغییرات را در مخزن شخصی خود انجام دهید.
- برای این پروژه و پروژه‌های بعدی یک شاخه (branch) جدید با نام ca#projectNumber ایجاد کرده و تغییرات خود را روی این شاخه اعمال کنید.
- گزارش کار در قالب یک فایل pdf با نامی شامل شماره دانشجویی اعضای گروه باشد.
- برای تحویل کفایت یکی از اعضای گروه گزارش پروژه که شامل آدرس مخزن شخصی و شناسه آخرین کامیت می‌باشد را در صفحه درس بارگذاری نماید.
- کاربر UTTest-Fall01 را به مخزن خود اضافه کنید.
- برای ارتباط با دستیاران آموزشی می‌توانید از ایمیل های [sarinahamedani3@gmail.com](mailto:sarinahamedani3@gmail.com) و [nazaninyousefian79@gmail.com](mailto:nazaninyousefian79@gmail.com) استفاده کنید.