



## تمرین کامپیوتری شماره ۱ آشنایی با چارچوب Junit



آزمون نرم افزار - پاییز 1401

مهلت تحویل:

دانشکده مهندسی برق و کامپیوتر

طراح تمرین : **امیرحسین علیزاد**،  
**محمد مهدی صالحی**

پایان روز 25 آبان

استاد: دکتر احسان خامس پناه

هدف از این پروژه کسب آشنایی اولیه با چارچوب JUnit می باشد. در ابتدا لازم است متذکر باشید که در پروژه های این درس شما از پروژه Enrollment-Assist به عنوان نمونه ای از یک برنامه ی وب قابل آزمون استفاده می شود. برای دسترسی به این پروژه به این [لینک](#) مراجعه کنید. یادتان باشد که برای هر پروژه حتما بررسی کنید که آخرین نسخه مخزن مورد نظر را دریافت کرده باشید.

**بخش اول:** آزمون های مربوط به متد hasConflict از کلاس PresentationSchedule را پیاده سازی نمایید. اطمینان حاصل کنید که کارکردهای مختلف این متد به درستی و حداقل یک بار آزموده شده باشند. حتما در نظر داشته باشید که تا جای ممکن از نوشتن آزمون های تکراری خودداری کنید.

**بخش دوم:** با یکی (یا هر دو) از روش های زیر برای متد های isPassed از کلاس StudyRecord و checkEnrollmentRules از کلاس EnrollmentList آزمون پیاده سازی کنید.

- روش Theories
- روش Parametrized<sup>1</sup>

<sup>1</sup> نیازی به نشان دادن کد آزمون های پیاده سازی شده برای بخش های اول و دوم در گزارش نیست!

بخش سوم (گزارش کار): گزارش کار خلاصه و کوتاهی برای موارد زیر تدوین نمایید:

- معیارهای که بر اساس آنان آزمون‌های خود را توسعه داده‌اید را به طور خلاصه بیان کنید.
- در مورد کاربرد متد assertTrue تحقیق کنید و نتیجه را به طور مختصر بیان کنید.
- به نظر شما چرا برای نام بردن از JUnit از لفظ چارچوب به جای کتابخانه استفاده می‌شود؟
- مشکلات هر یک از آزمون‌های زیر را در صورت وجود به طور خلاصه بیان کنید.

```
public List<String> f(list<String> inputList) {  
    return  
    inputList.stream().map(String::toUpperCase).sorted()  
    .collect(Collectors.toList());  
}  
@Test  
public void test1() {  
    List<String> result = f(Arrays.asList("Cb", "ab", "bA",  
    "aa", "Ca"));  
    List<String> expected= Arrays.asList("AA", "AB", "BA",  
    "CA", "CB");  
    Assert.assertEquals(expected, result);  
}
```

```
@Test(expected = ConverterException.class)  
public void checkConverter() {  
    Converter fixture = new Converter();  
    Assert.assertEquals(10, Converter.convert("ten"));  
    Assert.assertEquals(0, Converter.convert("Zero"));  
    Assert.assertEquals(-1, Converter.convert("Minus One"));  
}
```

```
public String format(Date date) { return "[" + date + "]; }  
@Test  
public void testFormat() {  
    String result = format(new Date());  
    Assert.assertEquals("[ " + new Date() + "]", result);  
}
```

```

public class TestCalculator() {
    Calculator fixture = Calculator.getInstance();

    @Test
    public void testAccumulate() {
        fixture.setInitialValue(0);
        int result = fixture.accumulate(50);
        Assert.assertEquals(50, result);
    }

    @Test
    public void testSubsequentAccumulate() {
        int result = fixture.accumulate(100);
        Assert.assertEquals(150, result);
    }
}

```

## 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.
- 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 بارگزاری نماید.
- کاربر UTTest-Fall01 را به مخزن خود اضافه کنید.
- برای ارتباط با دستیاران آموزشی می‌توانید از ایمیل‌های [aalizad79@gmail.com](mailto:aalizad79@gmail.com) و [msalehi20@gmail.com](mailto:msalehi20@gmail.com) استفاده کنید.