



## مقدمه

هدف از این پروژه کسب آشنایی اولیه با آزمون واحد<sup>1</sup> و چارچوب JUnit می باشد. فعالیت ها مرتبط با آزمون واحد کوچکترین بخش های قابل آزمایش یک برنامه را به طور جداگانه، برای عملکرد صحیح مورد بررسی قرار می دهد. چارچوب JUnit ابزاری برای آزمون نرم افزار در زبان جاوا می باشد که به برنامه نویسان این امکان را می دهد تا با یک روش استاندارد آزمون های خود را بنویسند و آن ها را اجرا کنند.

## تکلیف اول

(۱) ابتدا پروژه را از لینک [<https://github.com/Piralilu/CA1>] دریافت کرده (Clone کنید) و سپس یک repository در صفحه شخصی خود ایجاد کرده و تغییرات لازم را روی آن اعمال کنید.

الف- برای هر سه کلاس موجود در دایرکتوری Model آزمون هایی که صحت عملکرد درست کلاس ها را از جهات مختلف تایید می کنند بنویسید و همچنین از نوشتن مورد آزمون های تکراری و معادل خودداری کنید.

ب- متد Print در کلاس FactorPrinter را ابتدا بطور کامل برای خودتان تحلیل کنید و سپس آزمون های برای پوشش حداکثر حالات ممکن برای این متد پیاده سازی کنید.

---

<sup>1</sup> Unit Testing

۲) برای پاسخ به پرسش‌های بعدی، این مسئله را در نظر بگیرید: یک سامانه انتخاب واحد دانشگاهی رعایت پیش‌نیازهای یک درس را برای اخذ آن چک می‌کند. تابع زیر با دریافت لیستی از سطرهای کارنامه دانشجو و یک درس، این قضیه را چک می‌کند.

```
// اطلاعات درس
class Course {
    int id;
    List<Integer> pre;    // شناسه درس‌های پیش‌نیاز
}

// یک سطر کارنامه دانشجو
class Record {
    int termId;
    int courseId;
    double grade;
    boolean isMehman;
}

public static boolean hasPassedPre(List<Record> rec, Course course) {
    for (int i = 0; i < course.pre.size(); i++) {
        boolean prePassed = false;
        for (int j = 0; j < rec.size(); j++) {
            if (rec.get(j).courseId != course.pre.get(i))
                continue;
            if (rec.get(j).grade >= 10 && (!rec.get(j).isMehman ||
                rec.get(j).grade >= 12)) {
                prePassed = true;
                break;
            }
        }
        if (!prePassed)
            return false;
    }
    return true;
}
```

شرط دوم تابع بیان‌کننده این است که برای گذراندن یک درس باید دست کم نمره ۱۰ کسب شود و اگر درس به صورت مهمان اخذ شده دست کم نمره ۱۲ باید کسب شده باشد.

برای بررسی درستی این تابع، آزمون‌هایی را پیاده‌سازی کنید.

۳) مشکلات هر یک از آزمون های زیر را در صورت وجود بیان کنید و راه حل های (احتمالی) برای تصحیح هر کدام را ارائه دهید.

```
@Test
public void testA() {
    Integer result = new SomeClass().aMethod();
    System.out.println("Expected result is 10. Actual result is " + result);
}
```

```
@Test
public void testC() expects Exception {
    int badInput = 0;
    new AnotherClass().process(badInput);
}
```

```
@Test
public void testInitialization() {
    // Initialize the configuration and resources
    Configuration.initialize();
    ResourceManager.initialize();
    // Perform assertions to validate the initialization
    // ...
}

@Test
public void testResourceAvailability() {
    // Check if a specific resource is available
    boolean isResourceAvailable = ResourceManager.isResourceAvailable("exampleResource");
    assertTrue(isResourceAvailable);
}
```

4) توضیح دهید آیا می توان با استفاده از آزمون واحد از درستی یک برنامه چند ریسمانی اطمینان حاصل کرد؟

## نکات پایانی

- پاسخ نهایی به صورت یک فایل zip شامل گزارش کلی و فایل‌های مربوط به پروژه، در سامانه بارگذاری شود.

- فایل نهایی با فرمت زیر و به صورت لاتین بارگذاری شود:

STST-HW1-Last name-first name

- در صورت وجود هرگونه ابهام یا مشکل می‌توانید از طریق رایانامه‌های زیر با دستیاران آموزشی در تماس باشید:

- [shendabadi@ut.ac.ir](mailto:shendabadi@ut.ac.ir)
- [shahriar.piralilu@ut.ac.ir](mailto:shahriar.piralilu@ut.ac.ir)
- [peyman.shabani@ut.ac.ir](mailto:peyman.shabani@ut.ac.ir)