# تمرین سوم آزمون نرم‌افزار[1]
# پوشش گراف

**مقدمه**

گراف‌های جهت دار پایه و اساس بسیاری از معیارهای پوشش هستند. با توجه به برنامه‌ای که تحت آزمایش است، ایده‌ی مورد نظر بدست آوردن یک مدل سطح بالا بر پایه‌ی گراف از برنامه موردنظر است. به عنوان مثال، متداول ترین مدل سطح بالای برنامه تبدیل سورس کد آن به گراف CFG می باشد. درک این نکته مهم است که گراف در حقیقت همان برنامه نیست و اگرچه مدل های سطح بالای برنامه برای ما مفید هستند اما با این وجود کاملاً با برنامه واقعی تفاوت دارند. بر این اساس، یک معیار پوششی مبتنی بر گراف، مجموعه تست‌های یک برنامه را از نظر چگونگی «پوشاندن» مدل انتزاعی سطح بالای برنامه را ارزیابی می کند.

1. برای متد visitOwnerPets در فایل ClinicServiceImpl تست های مورد نیاز برای Edge ، Node coverage Coverage و prime path coverage را مطابق با CFG بنویسید.
2. برای متد updateVisit در فایل VisitRestController تست های مورد نیاز برای All DU path را مطابق با CFG متد بنویسید.
3. برای متد updateOwner در فایل OwnerRestController ابتدا مسیرهای مورد نیاز برای Node Coverage و Edge Coverage را بنویسید سپس مسیرهایی که در Node coverage لحاظ می شوند اما در Edge Coverage در نظر گرفته نمی شوند را ذکر کنید.
4. برای متد saveUser در فایل UserServiceImpl ابتدا مسیرهای مورد نیاز برای Node Coverage و Edge Coverage مطابق با CFG بنویسید سپس مسیر هایی که در Node Coverage لحاظ می‌شوند اما در Edge Coverage در نظر گرفته نمی شوند و همچنین مسیر هایی که در edge coverage لحاظ می شوند اما در Prime path coverage در نظر گرفته نمی شوند را بنویسید.
5. درباره افزونه‌های تولید گزارش پوشش و امکانات آن‌ها خصوصا jacoco مطالعه نمایید (نیاز نیست چیزی بنویسید).
6. برای چهار متد فوق به تعداد دلخواه test case بنویسید تا به 100% Branch coverage دست پیدا کنید. در نظر داشته باشید که نیاز نیست سرویس‌ها را در کانتکس وب (با ریکوئست و ریسپانس) تست کنید و می‌توانید آن‌ها را متدهای ساده در نظر بگیرید. با دستوری مشابه **mvnw jacoco:report** می‌توانید گزارش پوشش را تولید کنید. طبعا متدهایی که پوشش کاملی دارند نیاز به افزودن تست ندارند.

---

1 طراحان: محمدعلی عادلی‌منش و کیارش آذرنیا

**نکات**

- تعداد دفعات تکرار حلقه را برای تمام توابع یک بار در نظر بگیرد.
- برای تمام متد ها اگر درون آن تابعی صدا زده می شود (مانند get و set) آن را مانند یک خط کد معمولی در نظر بگیرید و وارد جزئیات آن نشوید.
- گزارش تمرین یک فایل PDF شامل CFG، لیست تمام مسیرهای خواسته شده و تصاویر مربوط به Branch Coverage خواهد بود و در موارد کتبی می‌تواند تایپی یا تصویر نوشتار باشد.
- برای بدست آوردن Branch Coverage از ابزار Jacoco استفاده کنید.
- حتما پروژه را از ریپازیتوری آن دریافت نمایید و قبل از انجام از بروز بودن آن نسبت به شاخه‌ی مربوط به تمرین در مخزن اطمینان حاصل کنید.
- هش آخرین کامیت ریپازیتوری و فایل گزارش خود (در صورت نیاز) را در محل مربوطه ثبت نمایید. کاربر uttest را به مخزن پرایوت خود اضافه نمایید. برای این کار پیشنهاد ما این است که ابتدا مخزن تمرین را Fork کنید و سپس تنظیمات لازم را در Settings ریپازیتوری خود انجام دهید.
- در این تمرین، هم‌افزایی (اشتراک ایده، لینک‌های مفید، مشارکت در بحث‌های مربوطه، کمک به حل مشکلات حاشیه‌ای مثلا کانفیگ لازم برای IDE) در گروه کلاس توصیه می‌شود اما همکاری (اشتراک کد و پاسخ سوالات) تنها در گروه دو نفره‌ی تعریف‌شده قابل انجام است.