آزمون نرم‌افزار

پروژه نهایی

حانیه ناصری (۸۱۰۱۰۱۲۸۶)

سید محمد امین اطیابی (۸۱۰۱۹۸۵۵۹)

معرفی سیستم تحت آزمون

برای انجام این تمرین یکی از پروژه‌های درسی انتخاب شد. پروژه انتخاب ما، پروژه سوم درس برنامه سازی پیشرفته ارائه شده در نیم سال دوم سال تحصیلی ۹۹-۹۸ است. این پروژه برنامه‌ای جهت پیشنهاد انتخاب واحد است که با ورودی گرفتن دروس ارائه شده و نمرات دانشجو، به او برنامه‌ای جهت انتخاب واحد ترم پیش رو می‌دهد. پیاده‌سازی این تمرین کامپیوتری به زبان C++ بوده است که برای انجام این پروژه پیاده سازی جدیدی به زبان Java انجام دادیم.

این نرم‌افزار با بررسی دروس گذرانده شده کاربر، معدل وی، رعایت پیشنیازی و تداخلات دروس ارائه شده سعی می‌کند بهترین برنامه ممکن را به کاربر نمایش دهد.

روش و ابزارهای استفاده شده

برای تولید خودکار آزمایه[۱] در این پروژه از روش تولید آزمایه مبتنی بر جست و جو استفاده می‌کنیم. یکی از ابزارهای خوب برای تولید آزمایه به این روش، ابزار Evosuite است. این ابزار در کنار تولید آزمایه دارای قابلیت‌های گزارش‌گیری موارد مختلفی اعم از میزان پوشش و امتیاز موتاسیون است. این ابزار قابلیت استفاده به صورت مستقل و همچنین یکپارچه شدن با سیستم بیلد Maven را دارد.

در استفاده از این ابزار، criterion پوشش بر روی branch تنظیم شد تا پوشش شاخه‌های برنامه به ما گزارش داده شود. سپس در ادامه برخی گزارشات مورد نیاز برای نمایش تعریف شد تا در فایل گزارش خروجی، موارد مورد نیاز ما ثبت گردد. به دلیل امکان ارائه گزارشات پوشش و امتیاز موتاسیون، نیاز به استفاده از ابزار ثالث برای تولید این گزارشات نداشتیم.

با توجه به اینکه هدف از انجام این پروژه خودکار سازی فرایند تولید آزمایه بوده است، یکی از راه‌های بهبود این فرایند اضافه کردن این ابزار بر روی پروسه CI/CD بود تا در حین این فرایند آزمایه‌های جدید تولید شود و گزارشات پوشش و امتیاز موتاسین کد جدید هرچه سریعتر به توسعه دهنده داده شود. کد توسعه داده شده بر روی مخزنی در گیت‌هاب[۲] قرار گرفت. به کمک قابلیت GitHub Actions که توسط این پلتفرم ارائه می‌شود تولید و اجرا آزمایه‌های خودکار را به ازای هر push توسعه دهنده بر روی مخزن راه‌اندازی کردیم. سپس برای اینکه گزارشات در دسترس بوده و به سادگی قابل نمایش برای همگی باشند، اسکریپتی به زبان پایتون نوشته شد تا گزارشات تولید شده توسط Evosuite را خوانده و در قالب فایل Readme بر روی پروژه قرار دهد تا به سادگی و طور واضح، وضعیت کدها برای همه قابل مشاهده باشد.

---

[۱] Test Case
[۲] GitHub

گزارش پوشش[۳] و امتیاز تحول[۴]

جدول زیر نتیجه انجام فرایند تولید خودکار آزمایه و اجرا آن‌ها بر روی کد تحت آزمون است.

TARGET_CLASS	Criterion	Size	Length	Branch Coverage	Mutation Score
com.example.demo.util.Constants	BRANCH	1	1	1.0	1.0
com.example.demo.model.Course	BRANCH	17	46	1.0	0.6086956521739131
com.example.demo.model.Grade	BRANCH	7	17	1.0	0.7222222222222222
com.example.demo.model.CourseSchedule	BRANCH	14	50	1.0	0.1875
com.example.demo.SemesterCoursesSuggestion	BRANCH	9	60	1.0	0.2491638795986622

تحلیل نتایج و نتیجه‌گیری

مطابق نتایج نمایش داده شده در بخشی قبلی می‌بینیم که امتیاز موتاسیون برخی کلاس‌ها پایین است. در صورت تمرین ذکر شده بود که فرض شده ورودی‌ها بدون ایراد خواهند بود در نتیجه کد پیاده‌سازی شده فاقد مکانیزم‌هایی جهت صحت سنجی ورودی و جلوگیری از خطاست. در نتیجه با وارد کردن هر گونه داده‌ای که کوچکترین سو گیری از حالت صحیح را داشته باشد در ادامه با خطا مواجه می‌شویم. با اضافه کردن برخی شرط‌ها بر رو داده‌های ورودی که باعث سحت سنجی آن‌ها می‌شود، با طور قابل ملاحظه‌ای امتیاز موتاسیون کد افزایش یافت.

---

[۳] Coverage
[۴] Mutation