به نام خدا

افزایش تست‌پذیری برنامه با اعمال ریفکتورینگ خودکار روی کد پروژه

فهرست مطالب:

# مقدمه

# شرح تئوری کار

# شرح کد پروژه

## تولید کلاس دیگرام

## محاسبه پیچیدگی و تست‌پذیری کد

## بازسازی الگو کارخانه(factory pattern)

## بازسازی الگو تزریق(injection pattern)

# شرح آزمایش‌ها

# نتیجه گیری

# کارهای آینده

# مقدمه:

هدف از این پروژه ارائه روشی برای تایین میزان آزمون پذیری برنامه ها می باشد. برای این منظور می باید بتوان با تحلیل خود برنامه میزان وابستگی بین کلاس ها را مشخص کرد. مسلما با افزایش میزان وابستگی آزمون واحد (unit test) کلاس ها پیچیده تر می گردد. در واقع این وابستگی موجب می شود که ردیابی خطا در داخل برنامه پیچیده شود چرا که با مشاهده خطا در یک متد در داخل یک کلاس به سادگی مشخص نمی شود که ایا علت خطا در آن متد است یا در متدهایی که به آن ها وابستگی وجود دارد. می توان با استفاده از الگوهایی مانند الگوی تزریق(injection pattern) و الگو کارخانه (factory pattern) و به کار گیری راهکار های بازسازی(refactoring code) این گونه اشکالات یا در اصطلاح بوی بد کد(code smell) را از میان برداشت. در این پروژه با استفاده از یک مولد کامپایلر و برای برنامه های جاوا کد پایتون برای مراحل زیر ایجاد می گردد :

1. ایجاد ابزاری برای استخراج مدل ارتباطی کلاس ها از متن پروژه های جاوا
2. ایجاد ابزاری برای تحلیل مدل وابستگی کلاس ها و محاسبه میزان آزمون پذیری کد بر اساس میزان وابستگی کلاس ها به یک دیگر
3. ایجاد ابزاری برای اعمال الگوی تزریق بر کلاس های جاوا و در واقع تولید کلاس رابط(interface) برای برقراری ارتباط با هر کلاس
4. پیاده سازی الگوی کارخانه با بررسی چگونگی ارتباط بین کلاس ها

با استفاده از ابزارهای فوق سعی می کنیم که فرایند TDD را در قالب فرایند جدید پیشنهادی خود به نام TsDD (testability driven development ) بهبود بخشید.

# شرح تئوری کار:

شرح کد پروژه:

برای خودکار انجام دادن این بازنمایی از 3 بار پیمایش DFS درخت تجزیه استفاده شده است. برای بدست آوردن درخت تجزیه و پیمایش آن به صورت DFS از کتابخانه ANTLR استفاده شده است.

کلاس FixCreatorListener:

در این کلاس Listener مربوط به بازسازی کلاس creator نوشته شده است که در زیر به موارد مهم آن می‌پردازیم: