

UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE
DEPARTAMENTO DE INFORMÁTICA E MATEMÁTICA APLICADA
DISCIPLINA: TESTE DE SOFTWARE II
PROFESSORA: ROBERTA COELHO

PROJETO: GERAÇÃO AUTOMÁTICA DE TESTES

OBJETIVO: Gerar Testes JUnit utilizando as ferramentas Randoop, Evosuite e ChatGPT. Analisar e comparar os testes gerados.

PRAZO DE ENTREGA: 7 DE OUTUBRO DE 2024 - SEGUNDA-FEIRA - ÀS 13HS

PASSO 1. ESCOLHA DAS CLASSES

Para a realização da atividade de geração de testes automatizados, vocês precisarão selecionar um projeto Java (ou conjunto de classes relacionadas).

Características deste conjunto:

- Conjunto de no mínimo 2 classes Java
- Classes devem possuir pelo menos 3 métodos
- Devem compilar no Java 8 (por limitação das ferramentas utilizadas)

Nesta atividade você deve enviar um zip com a lista de classes Java escolhidas.

Lembrando que o envio desta atividade é individual.

- Insira propositalmente um defeito no código de 1 das classes.

PASSO 2. CONFIGURAÇÃO DO AMBIENTE PARA JAVA 8.

Para execução da ferramenta Randoop você precisará retroceder a versão do JDK da sua máquina para Java 8. Dependendo o seu sistema operacional você encontra o passo a passo na Web.

PASSO 3. GERAÇÃO AUTOMÁTICA DE TESTES COM RANDOOP

Gere testes para classe Java na qual o defeito foi injetado utilizando as ferramentas Randoop, Evosuite e ChatGPT. A aula sobre Randoop apresenta exemplos de linha de comando que podem ser utilizados. A aula sobre Evosuite mostrou que existem outras opções além da linha de comando para execução da ferramenta, qualquer uma pode ser utilizada. Como as ferramentas Randoop e Evosuite se baseiam total ou parcialmente na geração aleatória de testes, execute a geração 3 vezes para cada

ferramenta (repita para o Chatgpt também), armazenando os testes gerados em cada execução.

PASSO 4. ANÁLISE DOS RESULTADOS

Para os teste gerados através de cada ferramenta responda as seguintes perguntas:

- a) Quantas classes de testes foram geradas para cada classe em cada execução?
- b) Quantos métodos de testes foram gerados em cada execução?
- c) Quantas linhas de código em classes de testes foram gerados em cada execução?
- d) O bug injetado pôde ser encontrado?
- e) Qual característica dos testes gerados lhe chamou mais atenção?