JABUTI & MUJAVA

JACKSON ANTONIO DO PRADO LIMA

SILVIA REGINA VERGILIO

JABUTI

- Proteum (Java Bytecode Understanding and Testing)
- Ferramenta desenvolvida no Instituto de Ciências Matemáticas e de Computação – ICMC/USP
- Apoia o teste estrutural para programas Java
- Implementa os critérios baseados em fluxo de controle e critérios baseados em fluxo de dados
- Realiza a análise sobre o bytecode Java e não sobre o programa fonte

AULA PRÁTICA (JABUTI)

- git clone https://github.com/jacksonpradolima/JaBUTi4Run.git

AULA PRÁTICA (JABUTI)

- Para rodar o JaBUTi executar ***run.sh***
- Se for a primeira vez
  - Clique em ***File*** > ***Open Class***
  - Informe a classe (binary - .class) que será testada
  - Em ***classpath*** informe o caminho para o arquivo (sem o nome do pacote).
    - Por exemplo:
      - ../src (certo)
      - ../src/paper (errado)

AULA PRÁTICA (JABUTI)

- Clique em ***OK***. Será aberto o gerenciador de projetos.
- Selecione em ***User Packager*** a classe que será (O ***classpath*** precisa estar correto para aparecer o arquivo da classe)
- Clique em >> (Segundo botão) para selecionar a classe a ser instrumentada
- Clique em Select e dê um nome ao projeto
- Clique em ***OK***
- Clique em ***File*** > ***Save Instrumented Classes*** e depois em ***Yes***, se a classe possuir um método principal, caso contrário ***No*** e depois ***OK***.

AULA PRÁTICA (JABUTI)

- Clique em ***Teste Case*** > ***Executing JUnit Test Set***
  - Informe em ***Path to JUnit test suite source code*** o caminho para o test (arquivo java sem o nome do pacote)
  - Informe em ***Path to JUnit test suite binary code*** o caminho para o teste (arquivo binário sem o nome do pacote)
  - Informe em ***Test suite full qualified name*** nome do arquivo de teste (sem extensão do arquivo e com o nome do pacote)
  - Informe em ***JaBUTi's library*** o jar do JaBUTi

AULA PRÁTICA (JABUTI)

- Verifique se o caminho para o javac está correto
- Clique em ***Compile Test Case*** e verifique se o arquivo ***.class*** foi gerado
- Clique em ***Run Normally (no trace)*** – Essa ação irá verificar e executar os casos de teste
- Clique em ***Run Collecting Trace Information*** – Essa ação irá habilitar um botão vermelho no JaBUTi e isso é para atualizarmos as informações do JaBUTi
- Fechar a janela do ***Test Case***

AULA PRÁTICA (JABUTI)

- Clique em ***Update*** > ***Update***
- Verifique a cobertura em ***Summary*** escolhendo o escopo da cobertura
- Clique em ***Reports*** > ***Custom Reports*** e dê um nome ao relatório

MUJAVA

- Ferramenta para teste de mutação em programas Java
- Provê uma grande gama de operadores de mutação para Java, tanto os operadores tradicionais de mutação (adaptados para orientação a objetos) quanto operadores no nível de classe
- Desenvolvido através da colaboração entre duas universidades:
  - Korea Advanced Institute of Science and Technology (KAIST) (South Korea)
  - George Mason University (USA)

MUJAVA (CONT.)

- Gera automaticamente os mutantes, executa-os junto a um conjunto de testes, posteriormente apresenta a pontuação das mutações em relação ao conjunto de teste.
- As principais funções dessa ferramenta são:
  1. Geração de mutantes;
  2. Análise de mutantes;
  3. Gerenciamento de casos de teste fornecidos pelo usuário.
- Essa ferramenta ainda implementa abordagens que automaticamente detectam alguns tipos de mutantes equivalentes.

AULA PRÁTICA (MUJAVA)

- git clone https://github.com/jacksonpradolima/MuJava4Run.git

AULA PRÁTICA (MUJAVA)

- Na pasta configuration determinar o caminho do projeto
- Criar uma sessão de experimentos (exemplo ***session1*** e ***session2*** da pasta ***examples***), ou através do executável ***makeStructure***
- ***makeStructure*** cria a estrutura de pastas, nesse caso coloque as pastas geradas dentro de uma pasta para experimentos, por exemplo:
  - experimentos
    - classes
    - result
    - src
    - testset

AULA PRÁTICA (MUJAVA)

- Na pasta ***src*** colocar o arquivo a ser mutado. ***(ver observação)***
- Na pasta ***classes*** colocar o arquivo .class do arquivo presente na pasta ***src.*** ***(ver observação)***
- Na pasta ***testset*** colocar o arquivo .class do arquivo de teste. ***(ver observação)***
- Na pasta ***result*** ficará os mutantes gerados
- ***Obs.: Caso possua algum nome de pacote o mesmo deve estar na estrutura de pastas (pasta dentro de pasta – paper/TriTyp).***

AULA PRÁTICA (MUJAVA)

- Para gerar os mutantes executar ***generator.sh***
- Para testar os mutantes executar ***tester.sh***

MATERIAL

- www.inf.ufpr.br/japlima