

# JABUTI e MUJAVA

---

HELSON LUIZ JAKUBOVSKI FILHO  
SILVIA REGINA VERGILIO

# JABUTI

---

- JABUTI (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 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/HelsonLJF/JaBUTi4Run.git>
- Para rodar o JABUTI executar: ***sh 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)
  - Clique em ***Ok***. Será aberto o gerenciador de projetos

# Aula prática - JABUTI

---

- Selecione em **User Package** a classe que será testada (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 de 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 ***Test Case > Executing Junit Test Set***
  - Informe em ***Path to Junit test suite source code*** o caminho para o teste (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*** o nome do arquivo de teste (com o nome do pacote e sem extensão do arquivo – “*pacote.arquivo*”)
  - Informe em ***JavBUTI'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 da JaBUTI
- Feche 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 realizar o 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 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

---

- Gera automaticamente os mutantes, executa-os junto a um conjunto de teste, posteriormente apresenta a pontuação das mutações em relação ao conjunto de teste.
- As principais funções dessa ferramenta são:
  - Geração de mutantes;
  - Análise de mutantes;
  - 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/HelsonLJF/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 os experimentos, por exemplo:
  - ***Experimentos***
    - ***classes***
    - ***result***
    - ***src***
    - ***testset***

# Aula prática - MUJAVA

---

- Na pasta ***src*** colocar o arquivo a ser mutado
- Na pasta ***classes*** colocar o arquivo ***.class*** do arquivo presente na pasta ***src***
- Na pasta ***testset*** colocar o arquivo ***.class*** do arquivo de teste
- 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 — ***pacote/arquivo***)

# Aula prática - MUJAVA

---

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

# Obrigado pela atenção!

---

Contato: [helson.Jakubovski@gmail.com](mailto:helson.Jakubovski@gmail.com)

# Referências

---

- Apresentação adaptada de: <https://jacksonpradolima.github.io/teaching/JaBUTi&MuJava.pdf>.