

Sobre a Jaguar

A Jaguar é uma ferramenta de localização de defeitos para o Eclipse. A Jaguar apresenta uma lista com os comandos mais suspeitos de conter defeitos. Para gerar a lista, a Jaguar usa os resultados da execução dos testes JUnit.

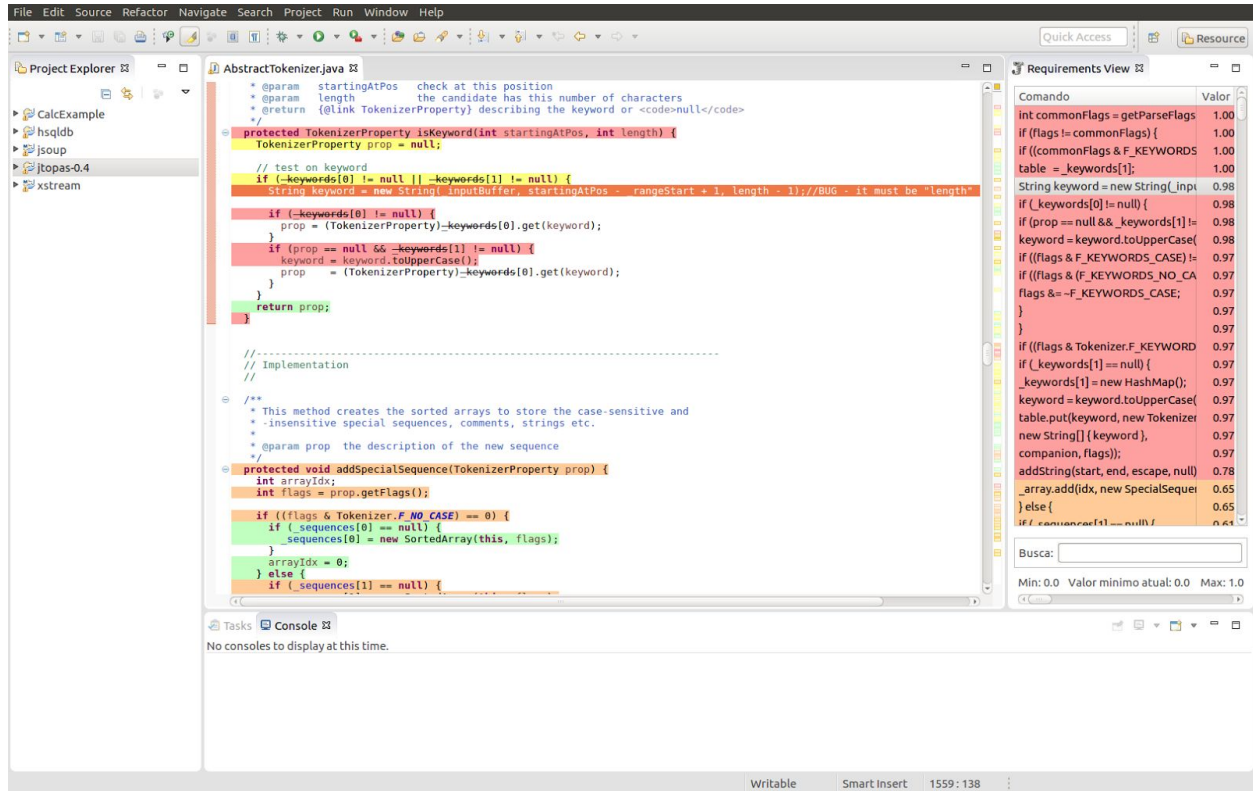


Figura 1: Plugin da Jaguar para o Eclipse.

A Figura 1 mostra a Jaguar no lado direito do Eclipse. A Figura 2 mostra detalhes da ferramenta - A área 1 contém duas colunas: *Comando* e *Valor*. A coluna *Comando* possui a lista com os comandos suspeitos de conter um defeito. A coluna *Valor* mostra o valor de suspeição de cada comando. Esse valor indica a probabilidade que cada comando tem de conter o defeito. Quanto maior o valor, mais suspeito ele é. Os comandos estão ordenados em ordem decrescente de valor.

As cores representam quatro níveis de valores: vermelho representa os comandos mais suspeitos, laranja são comandos mais suspeitos após os vermelhos, amarelos são os comandos com chance moderada de conter o defeito, e os comandos em verde são os menos suspeitos.

Navegação

Ao clicar em um comando na Jaguar, o arquivo que contém esse comando será aberto na área de edição do Eclipse (veja Figura 1). A Jaguar possui dois filtros: um controle *slider* (veja Figura

2, área 3) para filtrar os **comandos** por valor de suspeição; e uma busca textual (veja Figura 2, área 2), para selecionar **comandos** que contenham um termo de interesse.

Comando	Valor
int commonFlags = getParseFlags	1.00
if (flags != commonFlags) {	1.00
if ((commonFlags & F_KEYWORDS	1.00
table = _keywords[1];	1.00
String keyword = new String(_input	0.98
if (_keywords[0] != null) {	0.98
if (prop == null && _keywords[1] !=	0.98
keyword = keyword.toUpperCase(0.98
if ((flags & F_KEYWORDS_CASE) !=	0.97
if ((flags & (F_KEYWORDS_NO_CA	0.97
flags &= ~F_KEYWORDS_CASE;	0.97
}	0.97
}	0.97
if ((flags & Tokenizer.F_KEYWORD	0.97
if (_keywords[1] == null) {	0.97
_keywords[1] = new HashMap();	0.97
keyword = keyword.toUpperCase(0.97
table.put(keyword, new Tokenizer	0.97
new String[] { keyword },	0.97
companion, flags));	0.97
addString(start, end, escape, null)	0.78
_array.add(idx, new SpecialSeque	0.65
} else {	0.65
if (_sequences[1] == null) {	0.61

Busca:

Min: 0.0 Valor mínimo atual: 0.0 Max: 1.0

Figura 2: Detalhes da Jaguar

Vídeo

Preparamos um vídeo que mostra um exemplo da Jaguar em uso. Para vê-lo, abra o arquivo jaguar.mp4, localizado na área Desktop.

Teste a Jaguar

Na área Desktop, clique no ícone **Try Jaguar** para explorar a ferramenta antes de começar a realizar os experimentos. O programa **jtopas** foi disponibilizado para uso nesse exemplo.

1. Clique com o botão direito do mouse no programa jtopas.
2. Escolha a opção **Jaguar > Run Jaguar**.
3. O plugin será aberto no lado direito do Eclipse.
4. Clique nos comandos para ver como funciona a Jaguar.
5. O defeito está no comando da **linha 1559**:
String keyword = new String(_inputBuffer,startingAtPos-_rangeStart,length-1);
que fica no método **isKeyword(int,int)** da classe **AbstractTokenizer**. O valor atribuído à variável **keyword** deveria ser **length** ao invés de **length - 1**.
6. Feche o Eclipse e leia o arquivo **instructions.pdf** para iniciar as tarefas do experimento.