Code Critiques in the Pharo Debugger

Seminar Software Composition

Kevin Schibli

Supervisors:
Claudio Corrodi
Yuriy Tymchuk

Professor:
Oscar Nierstrasz

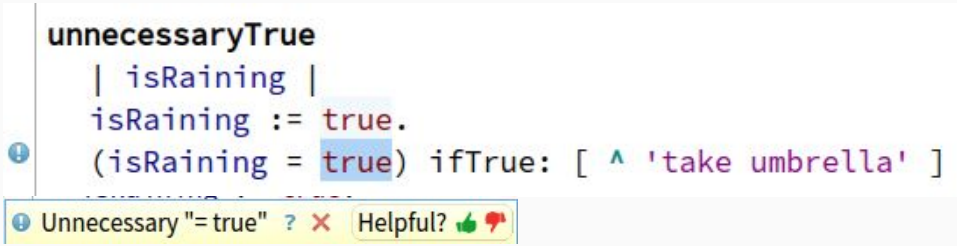
What are critiques?

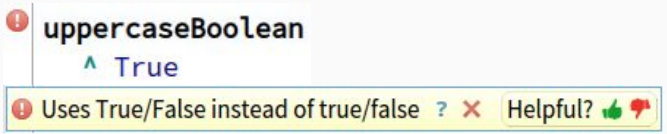
Created using static analysis

Present in most modern IDEs

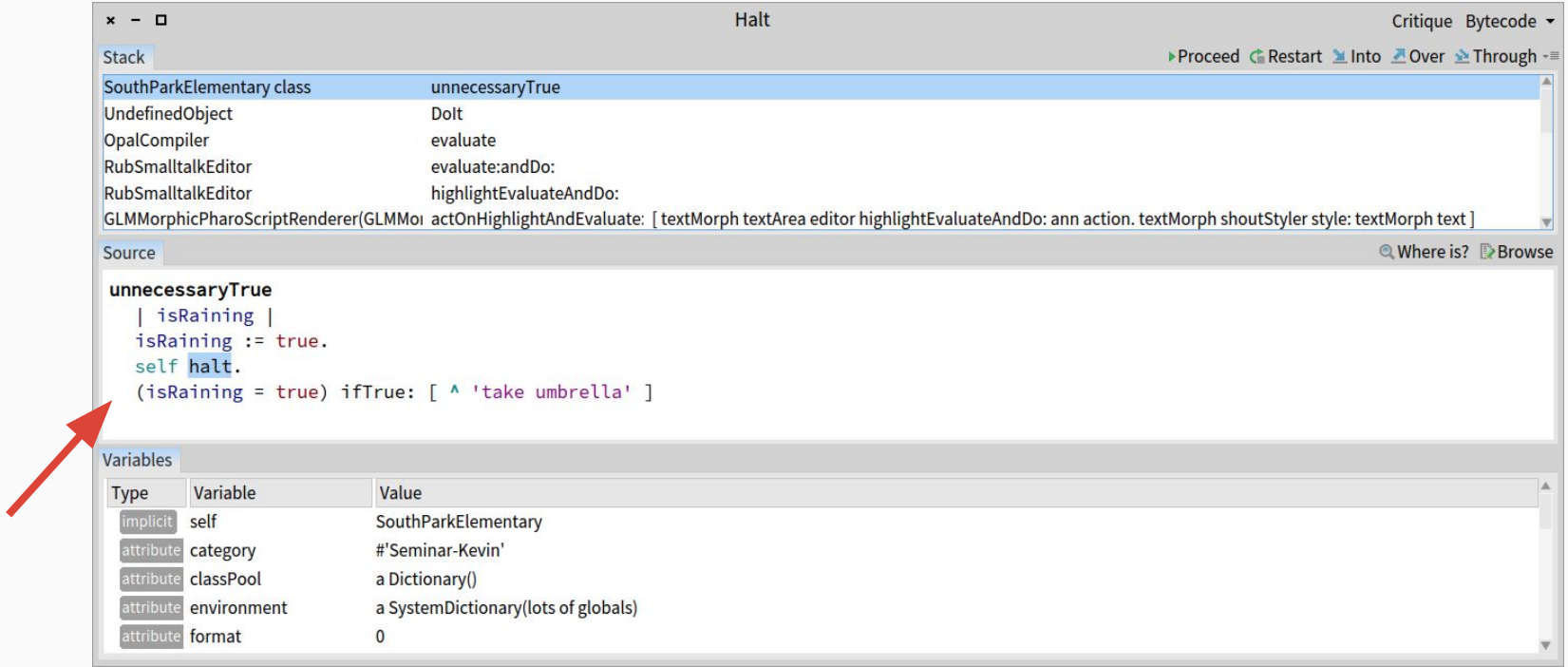
Give hints about:

- Missing documentation
- Bad practices
- Code smells
- Errors (mostly for static typed languages)

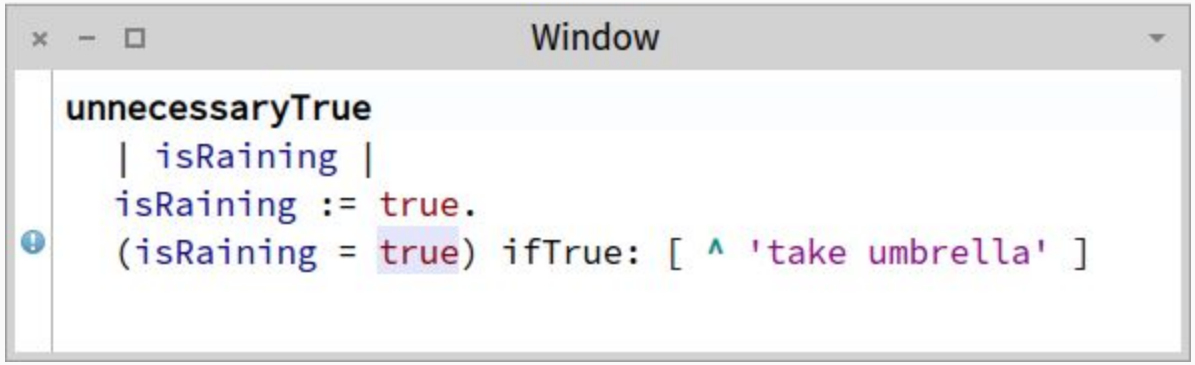




In Pharo people code in the debugger



Standalone window with RubPluggableTextMorph

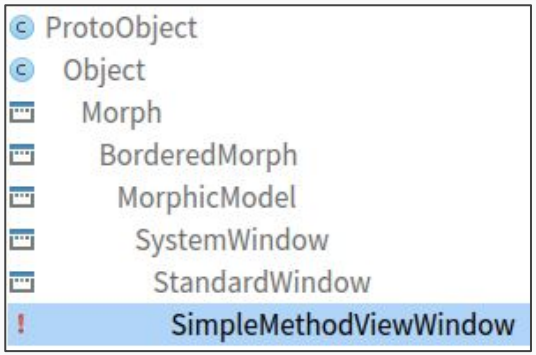


Standalone window with RubPluggableTextMorph

```
initialize
  super initialize.
  self initializeNewEditor.
  self addMorph: self editor fullFrame: LayoutFrame identity.
  ^ self
```

```
initializeNewEditor
  editor := RubPluggableTextMorph new.
  editor
    model: self model;
    beForSmalltalkCode;
    editingMode: self editingMode.
  ^ editor
```

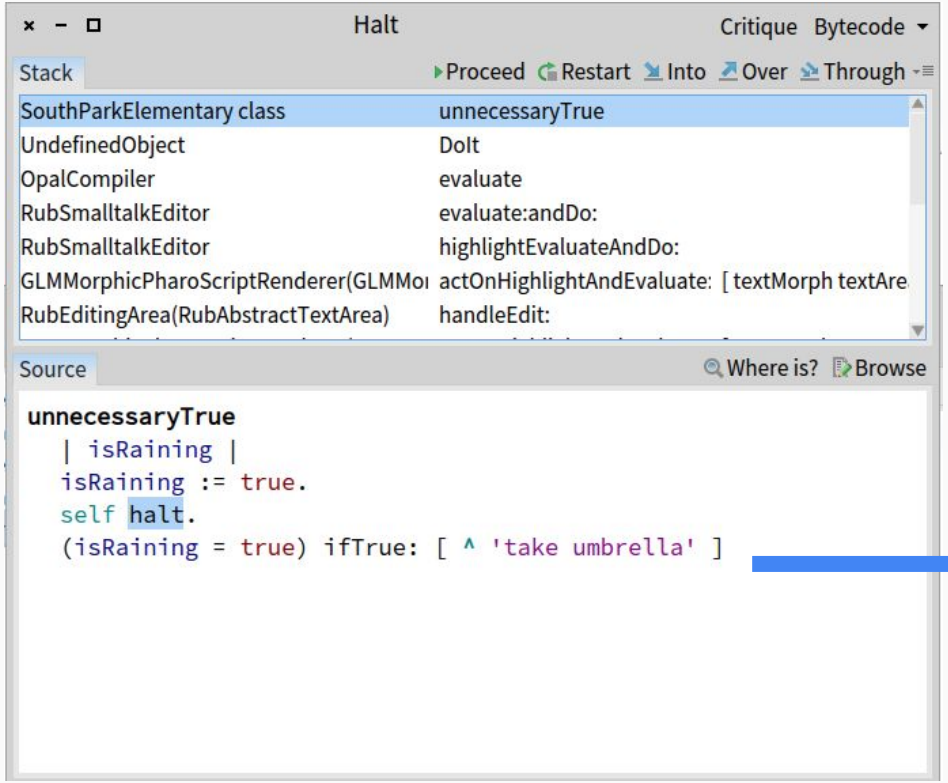
```
show: aMethod
  (aMethod isKindOf: CompiledMethod) ifTrue: [
    self setMethod: aMethod
  ] ifFalse: [
    self model setText: aMethod printString.
    self setTextEditingMode
  ].
  self isInWorld ifFalse: [ self openInWorld ]
```



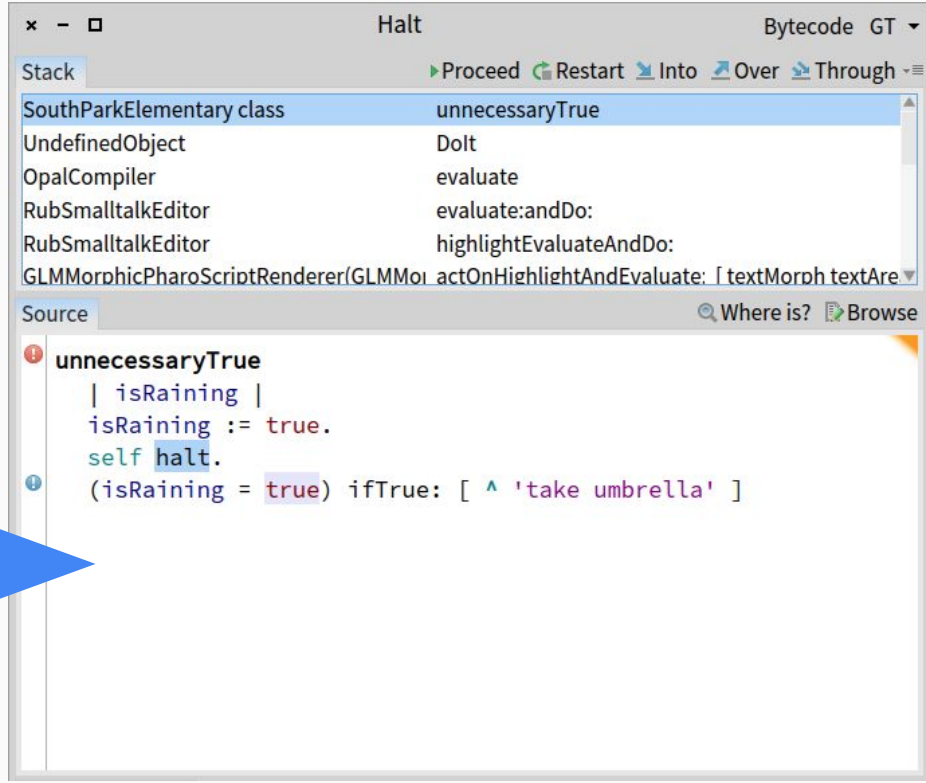

```
setMethod: aMethod
  self setMethodEditingModeFor: aMethod.
  self setText: aMethod sourceCode.
  aMethod annotateRubricText: self model
```

Change the editor of the debugger

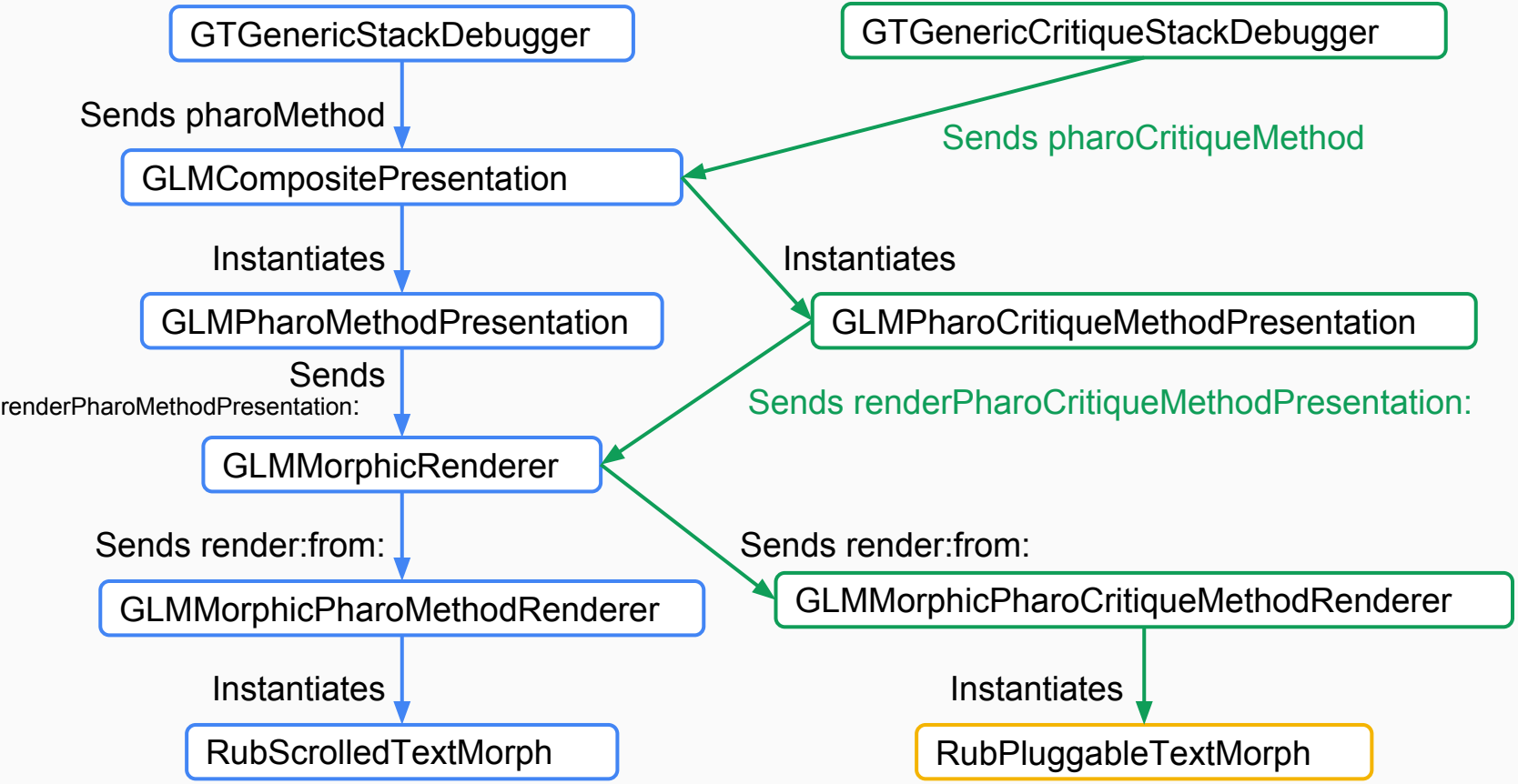
`GTGenericStackDebugger`



`GTGenericCritiqueStackDebugger`



Change the editor of the debugger



Change the editor of the debugger

GLMMorphicPharoMethodRenderer

```
morph
  |morph|
   morph := RubScrolledTextMorph new
     beForSmalltalkCode;
     getSelectionSelector: #primarySelectionInterval;
     model: textModel;
     color: Smalltalk ui theme backgroundColor;
     textFont: StandardFonts codeFont;
     yourself.
  ^ morph
```

GLMMorphicPharoCritiqueMethodRenderer

```
morph
  |morph|
   morph := RubPluggableTextMorph new
     beForSmalltalkCode;
     getSelectionSelector: #primarySelectionInterval;
     model: textModel;
     color: Smalltalk ui theme backgroundColor;
     textFont: StandardFonts codeFont;
     yourself.
  ^ morph
```

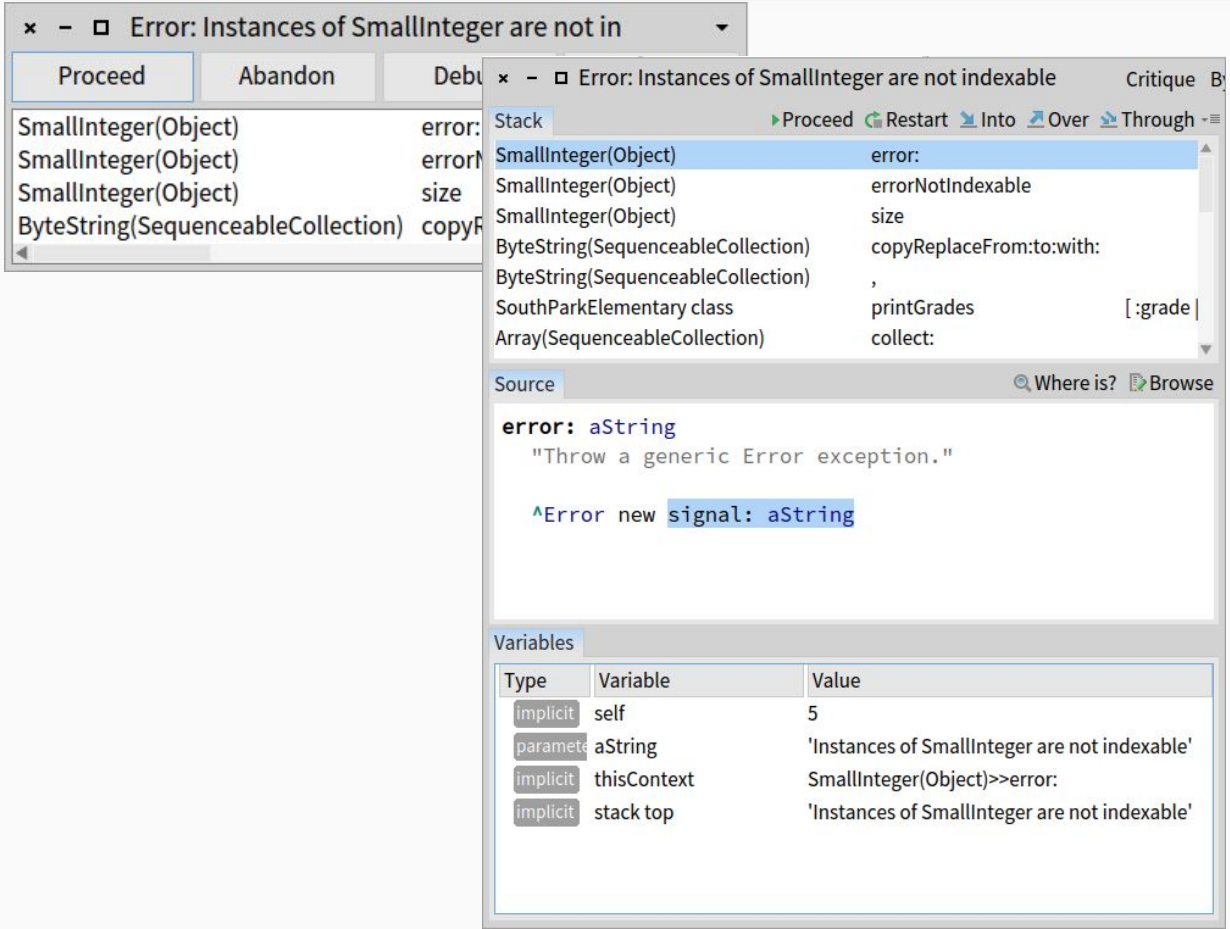
Change the editor of the debugger

GTGenericCritiqueStackDebugger

```
methodCodeWidgetIn: composite forContext: aContext
  ^ composite pharoCritiqueMethod
    title: 'Source';
    format: [ aContext sourceCode ];
    smalltalkClass: [  aContext methodClass ];
    doItReceiver: [ aContext receiver ];
    doItContext: [  aContext ];
+   editingMode: [ RubSmalltalkCodeMode new classOrMetaClass: aContext methodClass ];
+   textModel: [ :aTextModel |
+     aContext method annotateRubricText: aTextModel.
+     aContext annotateRubricText: aTextModel.
+   ]
```

What else can we do now?

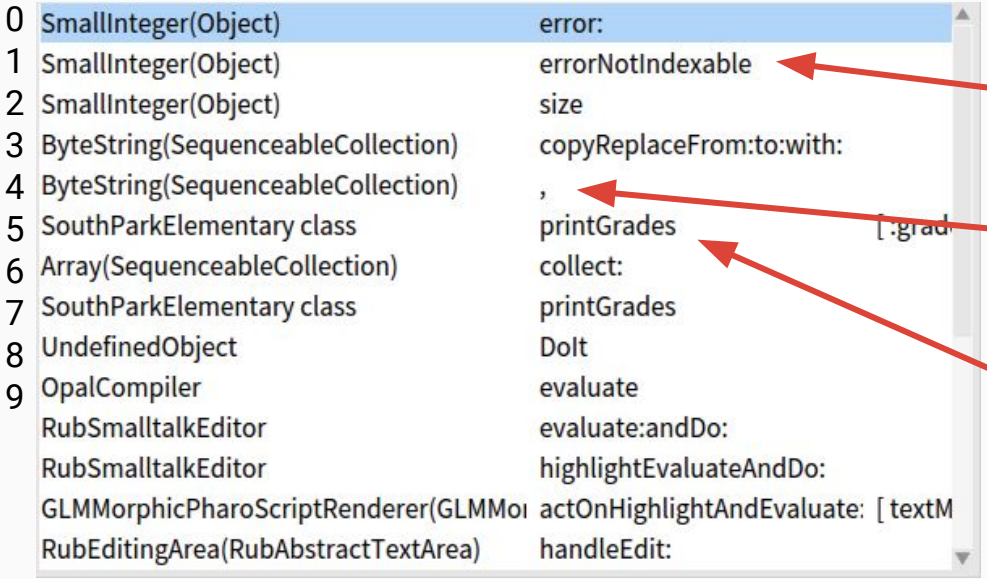
Dynamic critiques



`student,grade`

String Integer

Implement dynamic critiques



Something is not indexable...

Sequenceable Collection>>#, was called just before...

The error is here

Implement dynamic critiques

RuntimeChecker

```
isConcatError: aContext
  | notIndexableContext concatenationContext errorContext|
  notIndexableContext := self context: aContext descendBy: 1.
  concatenationContext := self context: aContext descendBy: 4.
  (notIndexableContext method = (Object >> #errorNotIndexable)
    and: [ concatenationContext method = (SequenceableCollection >> #,) ])
    ifTrue: [
      errorContext := self context: aContext descendBy: 5.
      ^ ReConcatenationError new
        title: '''' , concatenationContext receiver , ''' cannot be concatenated with ' , aContext receiver asString;
        errorContext: errorContext;
        initializeSourceAnchor: (ReSourceAnchor new initializeEnitity: errorContext method) yourself ]
    ifFalse: [ ^ nil ]
```

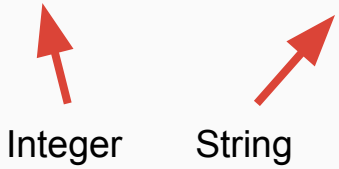
Implement dynamic critiques

isConcatenationError

```
student,grade
```

isAdditionError

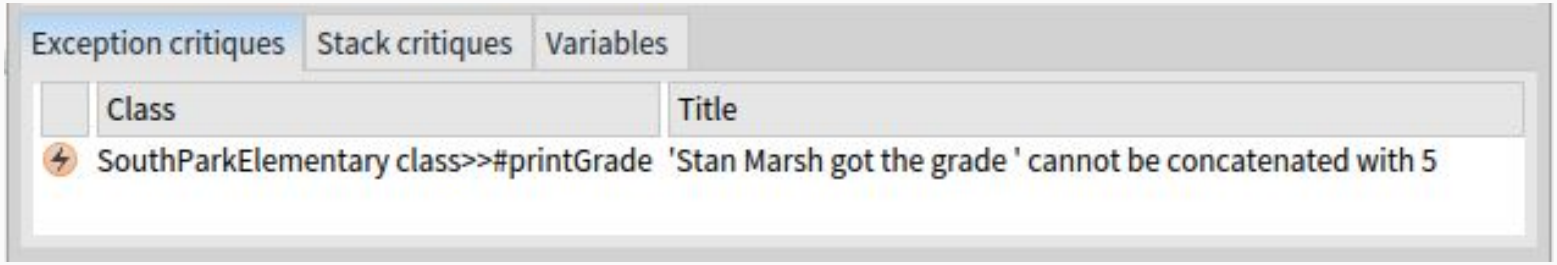
```
grade + bonus
```

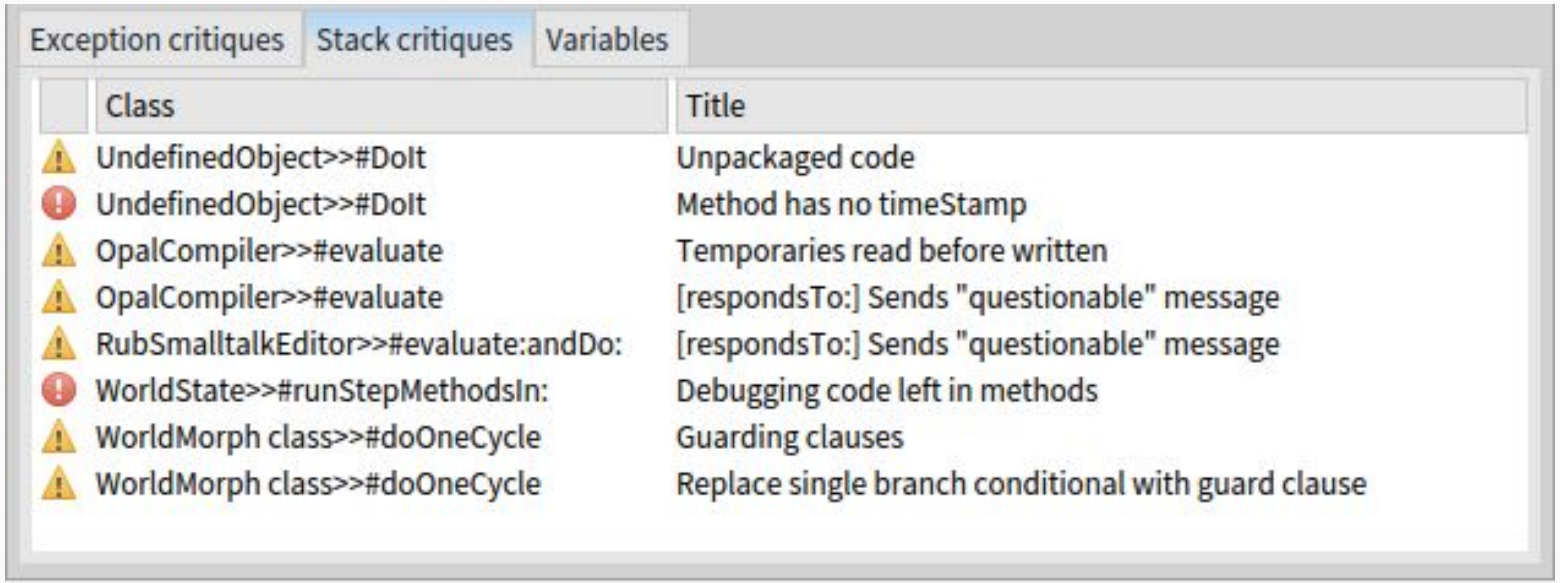


isNilError

```
grade rounded
```

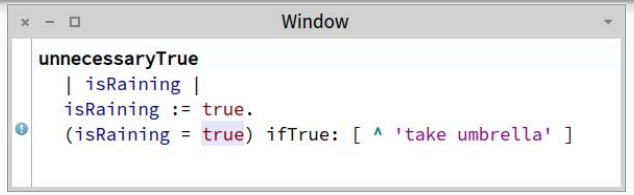
Display issue list in the debugger

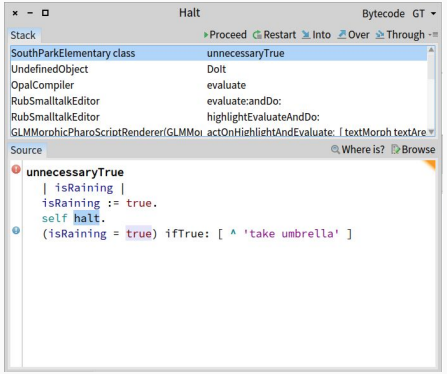


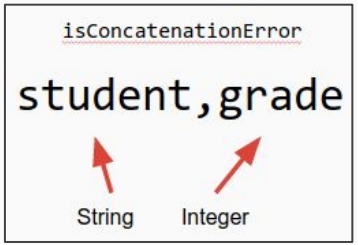


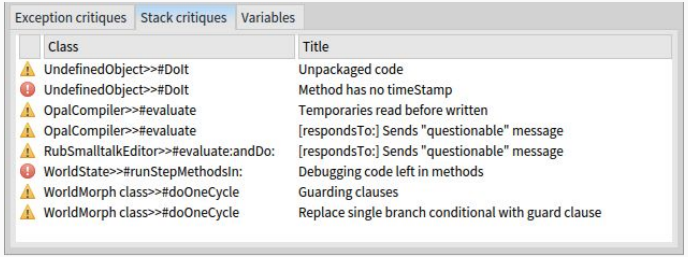
Conclusion

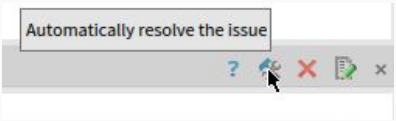
- Standalone window with critiques
- Change the editor of the debugger
- Implement dynamic critiques
- Display issue list in the debugger
- Autofix for dynamic critiques











Demo