Hera: A Year in Production

Growing a BDD Framework in Pharo

Koen De Hondt

ESUG 2026 · Plovdiv, Bulgaria

all: objects all: theTime

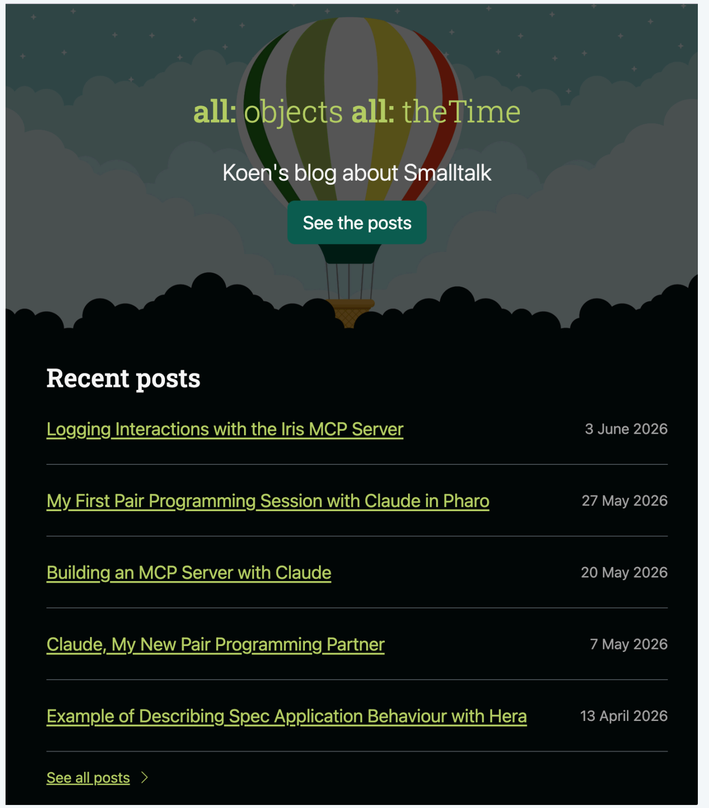
About Me

Long-time Smalltalker

Pharo contributor

Passionate tool builder

Blogger about Smalltalk



https://all-objects-all-the-time.st

Agenda

01

Hera

A BDD framework for Pharo

02

A Year of Evolution

A lot has happened since ESUG 2025 — functional overview

03

Hera in Industry

Automating regression test cases at Mediagenix

04

Experiments with AI-assisted Test Automation

AI writes Gherkin scenarios with Hera

05

What’s Next

Future plans

Part 01

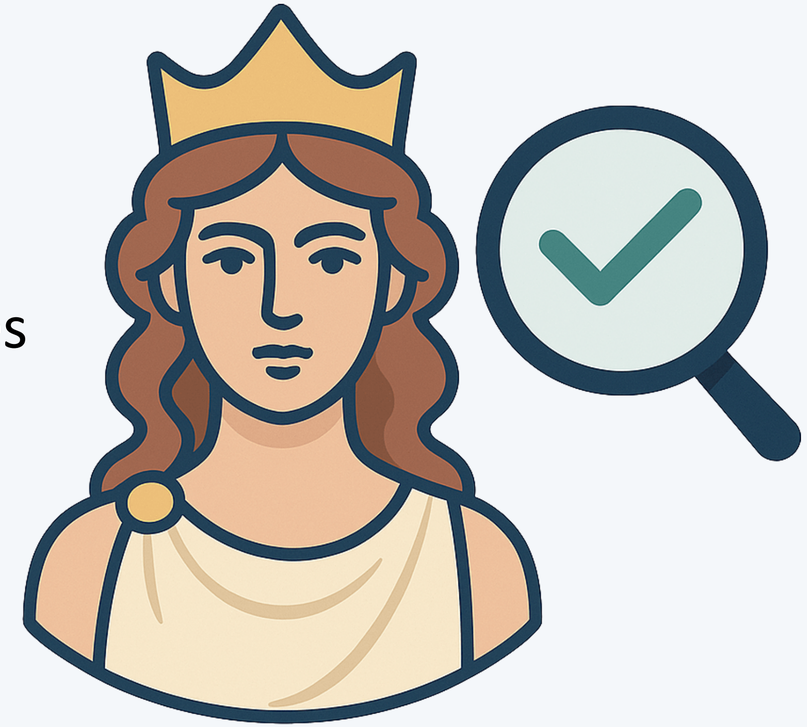
Hera

A BDD framework for Pharo

Hera

“Cucumber for Smalltalk”

- Tools for developing and running Gherkin features
- Gherkin features are organized in acceptance test classes
- Gherkin features and steps are methods



Open source

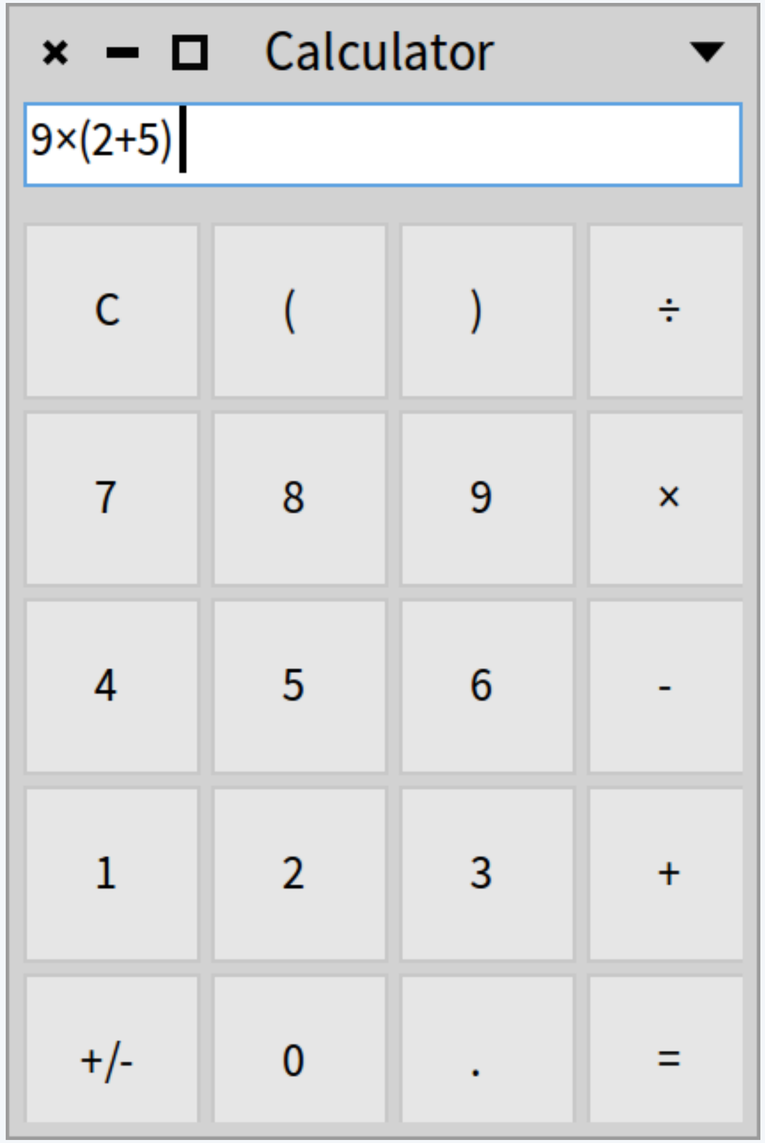
- Developed in Pharo
- https://all-objects-all-the-time.st/#/projects/hera
- https://github.com/koendehondt/hera-for-pharo

See https://cucumber.io and https://cucumber.io/docs/gherkin

Example Application

Simple calculator

Pharo Spec presenter



Gherkin

See https://cucumber.io/docs/gherkin

Script

```
Feature: Calculator

    Background:
        Given the calculator is open

    Scenario: Adding two numbers
        When I press the "1" button
        And I press the "+" button
        And I press the "2" button
        And I press the "=" button
        Then the calculator displays "3"

    Scenario: Multiplying two numbers
        When I press the "4" button
        And I press the "x" button
        And I press the "3" button
        And I press the "=" button
        Then the calculator displays "12"
```

Gherkin keywords

Feature:	High-level description of a feature
Background:	Shared steps for all scenarios
Rule:	One business rule (groups scenarios)
Scenario:	A concrete example / test case
Given / When / Then	Step
And / But	Continuation of previous step
Scenario Outline:	Parameterised scenario template
Examples:	Data table for Scenario Outline

Given · When · Then

Given

Initial context

Sets the system in a known state before the action. Describes preconditions.

Think: what must be true before this scenario can run?

```
Given the calculator is open
```

When

Action / Event

Describes an event triggered by the user or system. The core action of the scenario.

Think: what happens?

```
When I press the "1" button
And I press the "+" button
And I press the "2" button
And I press the "=" button
```

Then

Expected outcome

Describes the observable result. Should reference the UI or output only — not internal state.

Think: what should I see?

```
Then the calculator displays "3"
```

And / But — continuation of the previous step (reads more naturally than repeating Given/When/Then)

Step Format

Type	*Description with or without arguments*
Optional structured argument	

Given	the calculator is open

When	I press the "+" button

Then	the calculator displays "3"

Then	the "Results" list shows:

Name	Age
John	62
Jane	29
Bill	35
Megan	5

Step Arguments are Cucumber Expressions

Expression	Matches	Step example	Captured value
{string}	Quoted string	I search for {string}	"Pharo" 'BDD'
{word}	One word (no spaces)	the status is {word}	open closed
{}	Anything (anonymous)	I press the {} button	Save & Close Cancel
{int}	Integer	I have {int} apples	42 -1 0
{float}	Float / decimal	The price is {float}	3.14 -0.5
(text)	Optional text	I have {int} item(s)	item items
a/b	Alternative text	I start/restart the service	start restart

See https://github.com/cucumber/cucumber-expressions

Structured Step Arguments

Doc string

For multi-line text

Start and end with """
Each line aligned with left of """

```
Then I receive this receipt:
    """
    Apples                  3,34 €
    Green beans             4,67 €
    Cola                    8.40 €
    ————————————————————
    Total                  16.41 €
    """
```

Data table

For tabular data

Cells separated by |
Similar to markdown tables

```
Then the "Results" list shows:
    | Name   | Age |
    | John   | 62  |
    | Jane   | 29  |
    | Bill   | 35  |
    | Megan  | 5   |
```

Acceptance Test Class

Script

```
Feature: Calculator

    Background:
        Given the calculator is open

    Scenario: Adding two numbers
        When I press the "1" button
        And I press the "+" button
        And I press the "2" button
        And I press the "=" button
        Then the calculator displays "3"

    Scenario: Multiplying two numbers
        When I press the "4" button
        And I press the "×" button
        And I press the "3" button
        And I press the "=" button
        Then the calculator displays "12"
```

Class

```
HeraSpecAcceptanceTest << #CalculatorAcceptanceTest
        slots: {};
        package: 'Hera-Demo'
```

Feature Method in Acceptance Test Class

Script

```
Feature: Calculator

    Background:
        Given the calculator is open

    Scenario: Adding two numbers
        When I press the "1" button
        And I press the "+" button
        And I press the "2" button
        And I press the "=" button
        Then the calculator displays "3"

    Scenario: Multiplying two numbers
        When I press the "4" button
        And I press the "×" button
        And I press the "3" button
        And I press the "=" button
        Then the calculator displays "12"
```

Feature method

```
featureCalculator

    <heraFeature: 'Calculator'>

    ^ (self feature: 'Calculator')
        background: (self background
            given: 'the calculator is open');
        scenarios: {
            (self scenario: 'Adding two numbers')
                when: 'I press the "1" button';
                and: 'I press the "+" button';
                and: 'I press the "2" button';
                and: 'I press the "=" button';
                then: 'the calculator displays "3"'.
            (self scenario: 'Multiplying two numbers')
                when: 'I press the "4" button';
                and: 'I press the "×" button';
                and: 'I press the "3" button';
```

And more

Step Methods in Acceptance Test Class

Script

```
Feature: Calculator

    Background:
        Given the calculator is open

    Scenario: Adding two numbers
        When I press the "1" button
        And I press the "+" button
        And I press the "2" button
        And I press the "=" button
        Then the calculator displays "3"

    Scenario: Multiplying two numbers
        When I press the "4" button
        And I press the "×" button
        And I press the "3" button
        And I press the "=" button
        Then the calculator displays "12"
```

Step methods

```
stepTheCalculatorIsOpen

    <heraStepDefinition: #(Given 'the calculator is open')>

    self openPresenter: CalculatorPresenter new


stepTheCalculatorDisplays: expectedString

    <heraStepDefinition: #(Then match 'the calculator displays {string}')>

    | actualString |
    actualString := (self presenterAt: 'display') text asString.
    self
        assert: actualString = expectedString
        description: [ 'The calculator displays "{1}"' format: { actualString } ]
```

Inherited Step Methods

Script

```
Feature: Calculator

    Background:
        Given the calculator is open

    Scenario: Adding two numbers
        When I press the "1" button
        And I press the "+" button
        And I press the "2" button
        And I press the "=" button
        Then the calculator displays "3"

    Scenario: Multiplying two numbers
        When I press the "4" button
        And I press the "×" button
        And I press the "3" button
        And I press the "=" button
        Then the calculator displays "12"
```

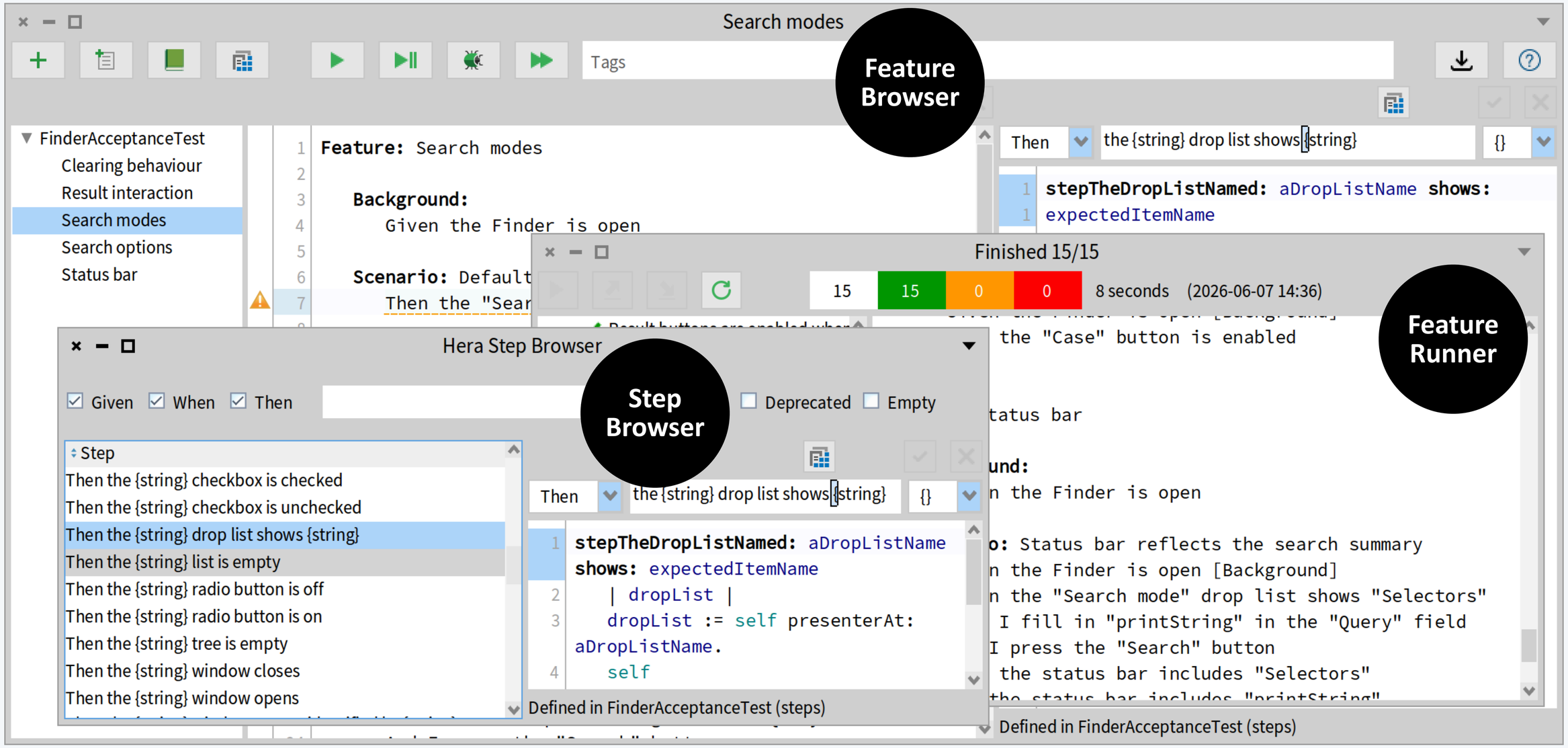
Step method inherited from HeraSpecAcceptanceTest

```
stepIPressTheButtonNamed: aButtonName

    <heraStepDefinition: #(When match 'I press the {string} button')>

    self click: aButtonName
```

Tools


Search modes
Tags
Feature Browser
FinderAcceptanceTest
Clearing behaviour
Result interaction
Search modes
Search options
Status bar
Feature: Search modes
Background:
Given the Finder is open
Scenario: Default
Then the "Sear
Then
the {string} drop list shows {string}
{}
stepTheDropListNamed: aDropListName shows:
expectedItemName
Finished 15/15
15
15
0
0
8 seconds
(2026-06-07 14:36)
Feature Runner
the "Case" button is enabled
tatus bar
und:
n the Finder is open
o: Status bar reflects the search summary
n the Finder is open [Background]
n the "Search mode" drop list shows "Selectors"
I fill in "printString" in the "Query" field
I press the "Search" button
the status bar includes "Selectors"
Defined in FinderAcceptanceTest (steps)
Hera Step Browser
Step Browser
Given
When
Then
Deprecated
Empty
Step
Then the {string} checkbox is checked
Then the {string} checkbox is unchecked
Then the {string} drop list shows {string}
Then the {string} list is empty
Then the {string} radio button is off
Then the {string} radio button is on
Then the {string} tree is empty
Then the {string} window closes
Then the {string} window opens
Then
the {string} drop list shows {string}
{}
stepTheDropListNamed: aDropListName
shows: expectedItemName
| dropList |
dropList := self presenterAt:
aDropListName.
self
Defined in FinderAcceptanceTest (steps)

Part 02

A Year of Evolution

A lot has happened since ESUG 2025 — functional overview

Gherkin Compatibility

Supported

Feature, Rule, Background, Scenario, Scenario Outline, Given, When, Then, And, But, *

Data table, doc string

Tag

Free-form text descriptions in Feature, Rule, Background, Scenario, Scenario Outline

{string}, {word}, {int}, {float}, {}

Hera Extensions

Data tree

{date}, {time}, {timestamp}

Not supported

Comments with #

Optional text with (…)
Alternative text with …/…

Scenario Outline

Scenario Outline

Parameterised scenarios

Runs one scenario per row in the Examples table.

```
Scenario Outline: Operations on two operands

    When I press the "<operand 1>" button
    And I press the "<operation>" button
    And I press the "<operand 2>" button
    And I press the "=" button
    Then the calculator displays "<result>"
    Examples:
        | operand 1  | operation  | operand 2 | result  |
        | 1          | +          | 2         | 3       |
        | 2          | ×          | 3         | 6       |
```

Data Tree as Structured Argument

Doc string

For multi-line text

Start and end with """
Each line aligned with left of """

```
Then I receive this receipt:
    """
    Apples                  3,34 €
    Green beans             4,67 €
    Cola                    8.40 €
    ————————————————————
    Total                  16.41 €
    """
```

Data table

For tabular data

Cells separated by |
Similar to markdown tables

```
Then the "Results" list shows:
    | Name   | Age |
    | John   | 62  |
    | Jane   | 29  |
    | Bill   | 35  |
    | Megan  | 5   |
```

Data tree

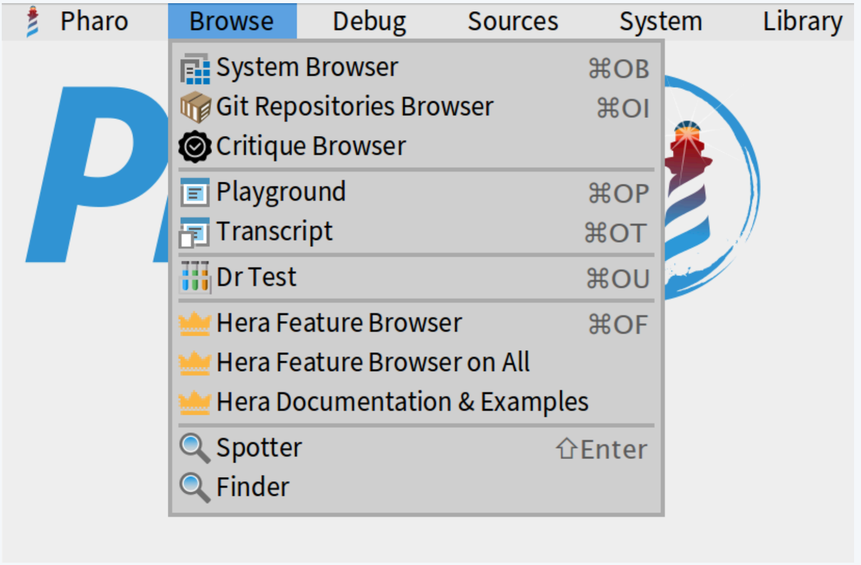
For tree structures

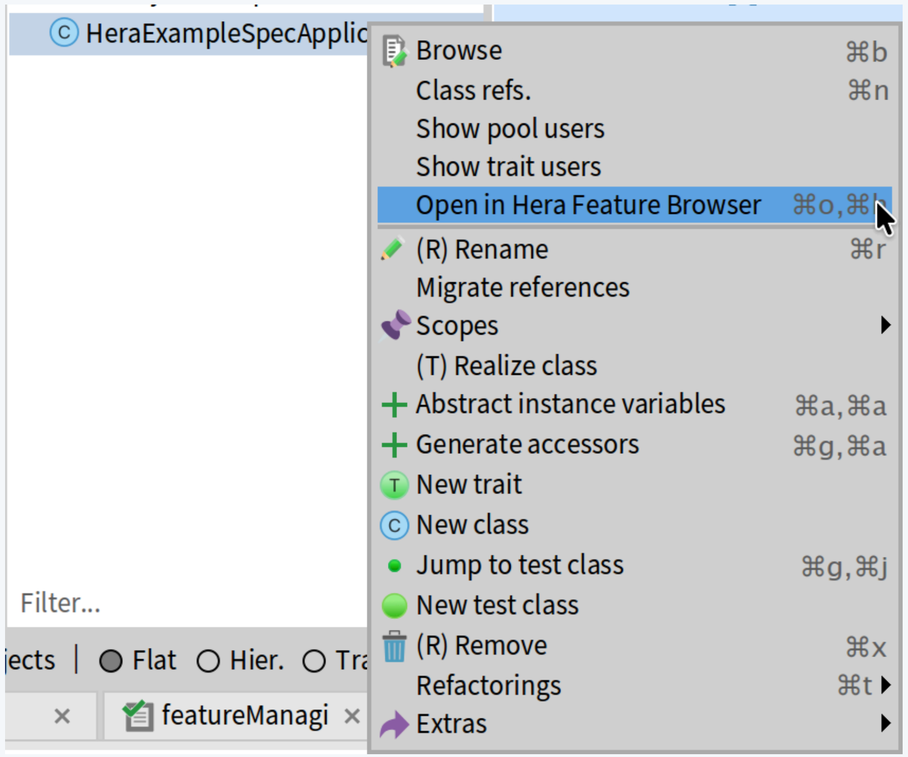
Each tree item starts with >
Multiple >>> indicate depth

```
Given a family tree:
    >      John
    >>       Jane
    >>>        Kevin
    >>       Megan
```

Integration in Pharo

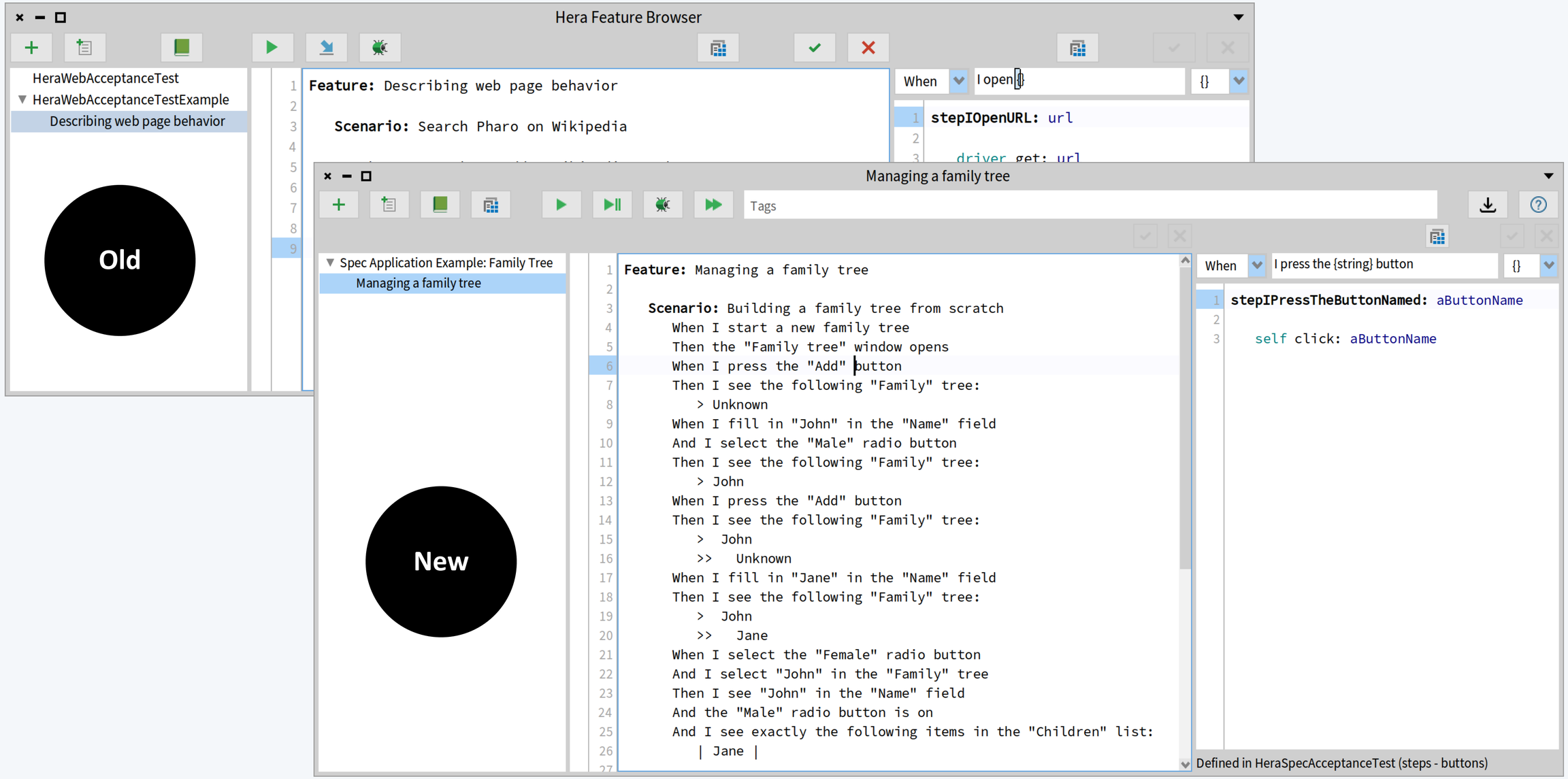
Feature Browser included in the Pharo menu bar



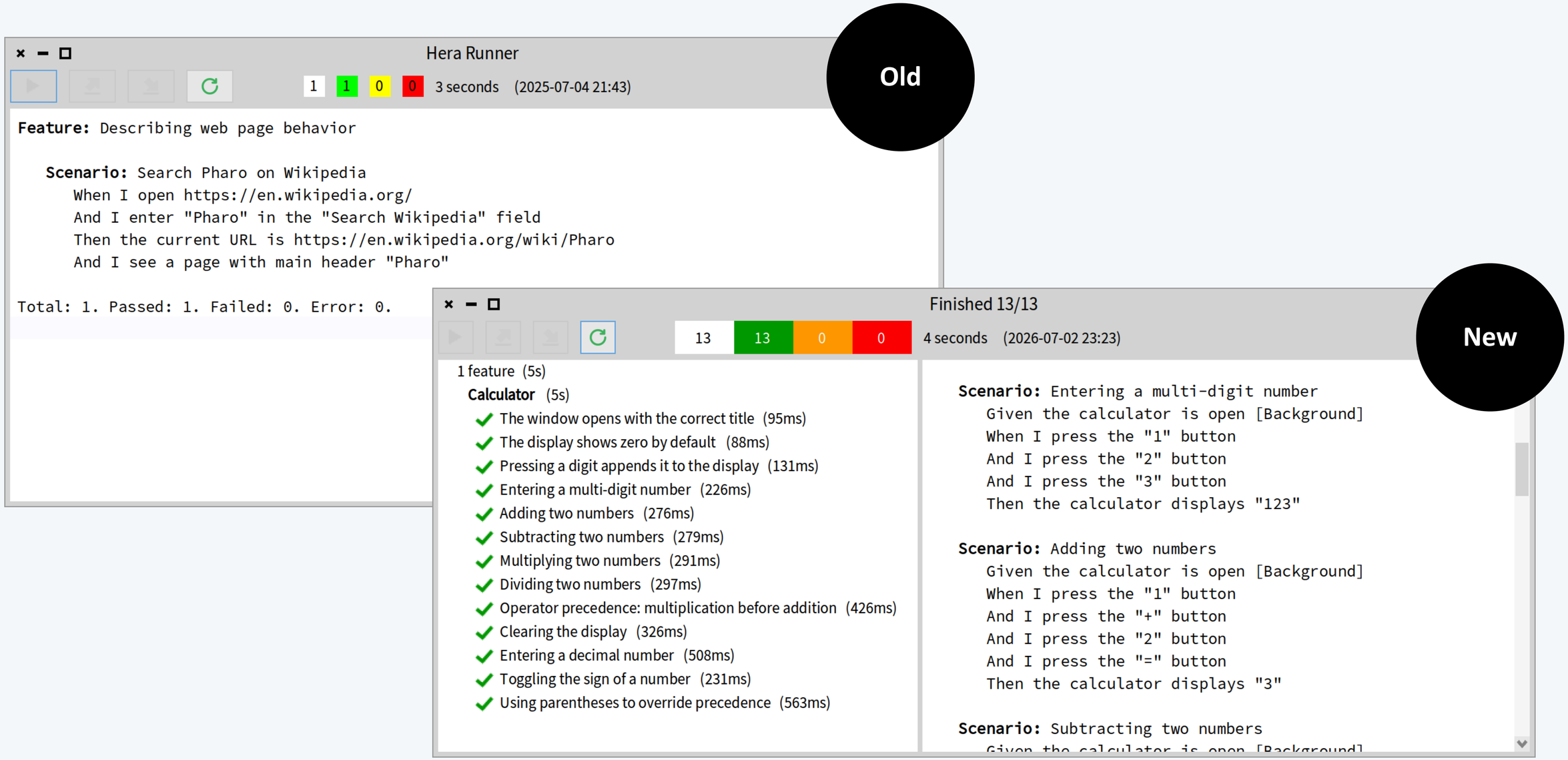


Open the Feature Browser on acceptance test classes and feature methods in the System Browser

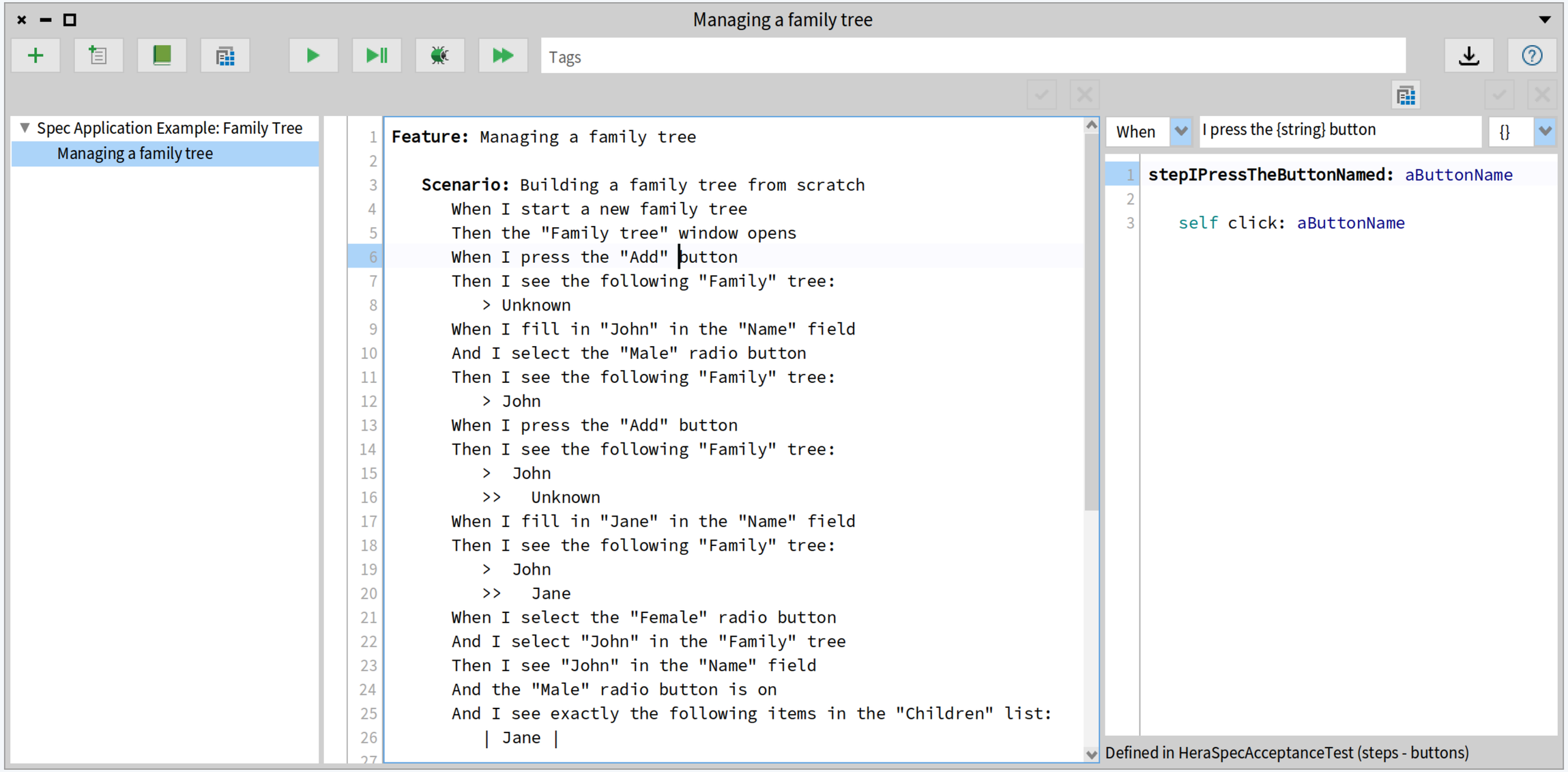
Feature Browser



Feature Runner



Describing Spec Applications



DSL for Describing Spec Applications

Refering to presenters

... "<name>" <presenter type> ...

Terminology in When and Then steps

When I <verb> the "<name>" <presenter type>
When I <verb> ... in the "<name>" <presenter type>
Then I see ...
Then ... is <state>
Then ... is <state> in the "<name>" <presenter type>
Then the {string} window opens
Then the {string} window closes

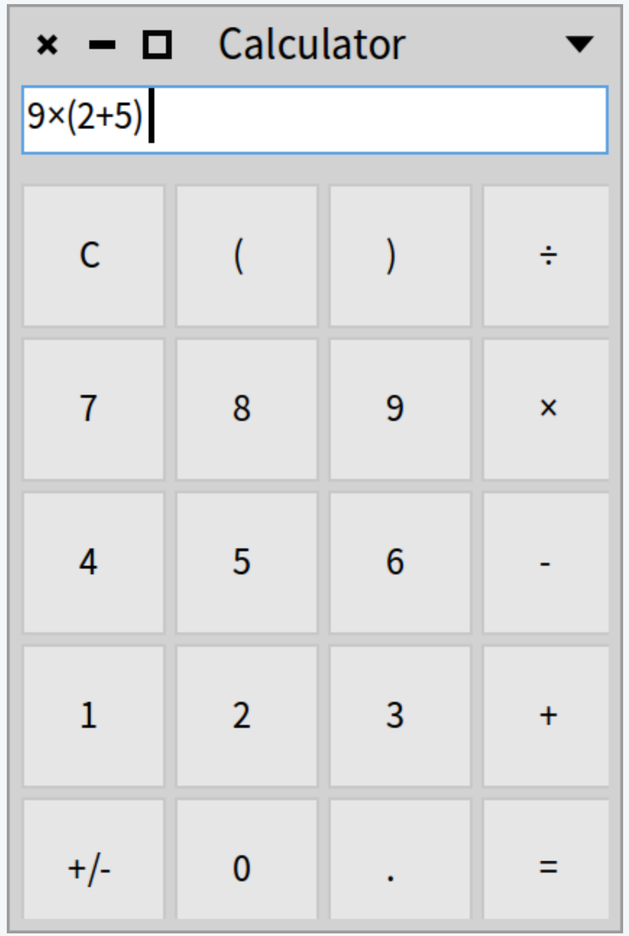
Examples

When I press the "Play" button
When I collapse "Jane" in the "Family" tree

Then I see the following "Family" tree:
Then I see "John" in the "Name" field
Then the "Search results" list is empty
Then the "Case insensitive" checkbox is checked
Then "William" is selected in the "Children" list

Then the "Finder" window opens
Then the “Family tree" window closes

Finding Presenters in Spec Applications



Steps fetch presenters with

```
self presenterAt: aPresenterName
```

Top-level presenters must implement

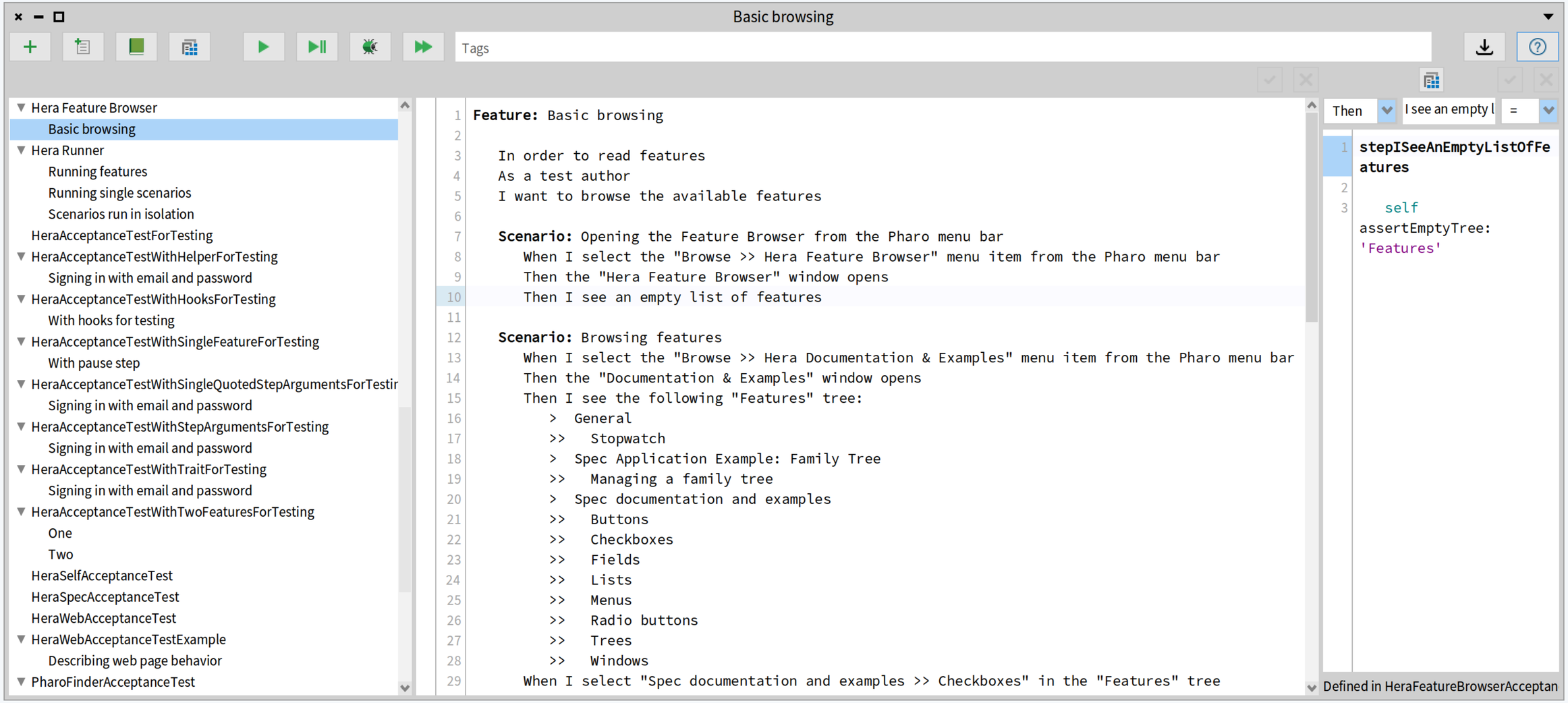
```
heraPresenterAt: aPresenterName ifAbsent: absentBlock
```

CalculatorPresenter example

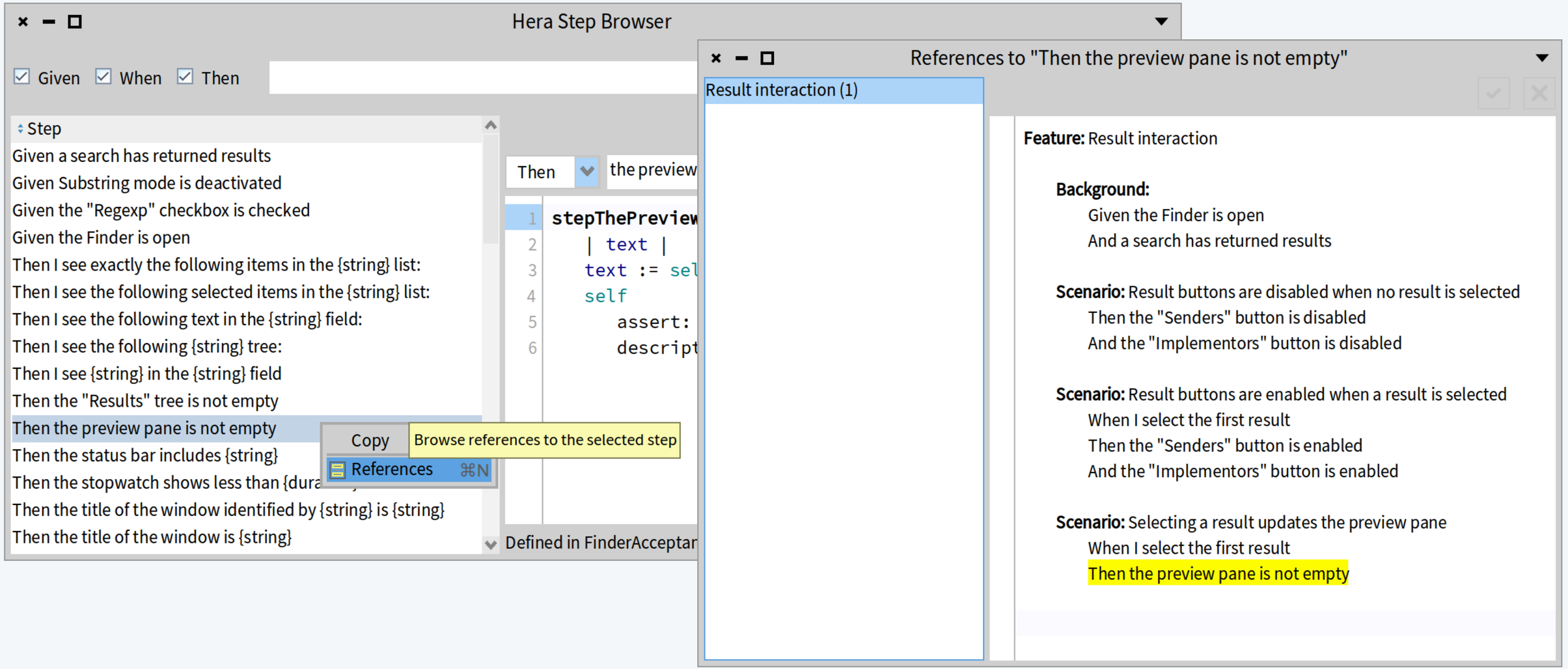
```
heraPresenterAt: aPresenterName ifAbsent: absentBlock

    aPresenterName = 'display' ifTrue: [ ^ display ].
    ^ buttons at: aPresenterName ifAbsent: absentBlock
```

Describing Hera



Hera References Browser



Part 03

Hera in Industry

Automating regression test cases at Mediagenix

Hera @ Mediagenix

CONTEXT	Mediagenix builds a Broadcast Management System	Big VisualWorks code base	Manual regression testing by the QA team
SOLUTION	Ported Hera to VisualWorks	DSL with 130+ reusable steps to interact with the application	Finding widgets based on the meta model
ADOPTION	8 squads are using Hera	Gherkin features are run on the CI system	Performance measurement scenarios in Hera

Part 04

Experiments with AI-assisted Test Automation

AI writes Gherkin scenarios with Hera

AI-assisted Gherkin Scenario Development

Me

Add a Hera acceptance test class. Add a Gherkin feature with scenarios for the calculator.

Claude

```
Feature: Calculator

    Background:
        Given the calculator is open

    Scenario: Using parentheses to override precedence
        When I press the "(" button
        And I press the "2" button
        And I press the "+" button
        And I press the "3" button
        And I press the ")" button
        And I press the "×" button
        And I press the "4" button
        And I press the "=" button
        Then the calculator displays "20"
```

And more

AI-assisted Gherkin Scenario Development

Me

The Finder application is implemented by the StFinderPresenter class. Analyse the class and propose some Gherkin scenarios for Finder functionality.

Claude

```
Feature: Finder

    Background:
        Given the Finder is open

    Scenario: Searching for a selector by name
        Given the "Search mode" drop list shows “Selectors"
        When I type "printString" in the "Query" input field
        And I click the "Search" button
        Then the "Results" tree is not empty

    Scenario: Searching by source code
        When I select "Source" in the "Search mode" drop list
        And I type "printString" in the "Query" input field
        And I click the "Search" button
        Then the "Results" tree is not empty
```

And more

Part 05

What’s Next

Future plans

The Future

1. **Update the Feature Browser on system changes**
   Update after feature and step method changes by the Pharo tools and AI through an MCP server
2. **Add more reusable steps for describing Spec applications**
   Cover more widgets
3. **Describe Pharo tools with Gherkin scenarios**
   Build regression test suite for Pharo tools built with Spec

Thank you!

Questions?

Koen De Hondt · koen@all-objects-all-the-time.st

github.com/koendehondt/hera-for-pharo

all-objects-all-the-time.st

all: objects **all:** theTime