

Hera: A Year in Production

Growing a BDD Framework in Pharo

Koen De Hondt

ESUG 2026 · Plovdiv, Bulgaria



all: objects all: theTime



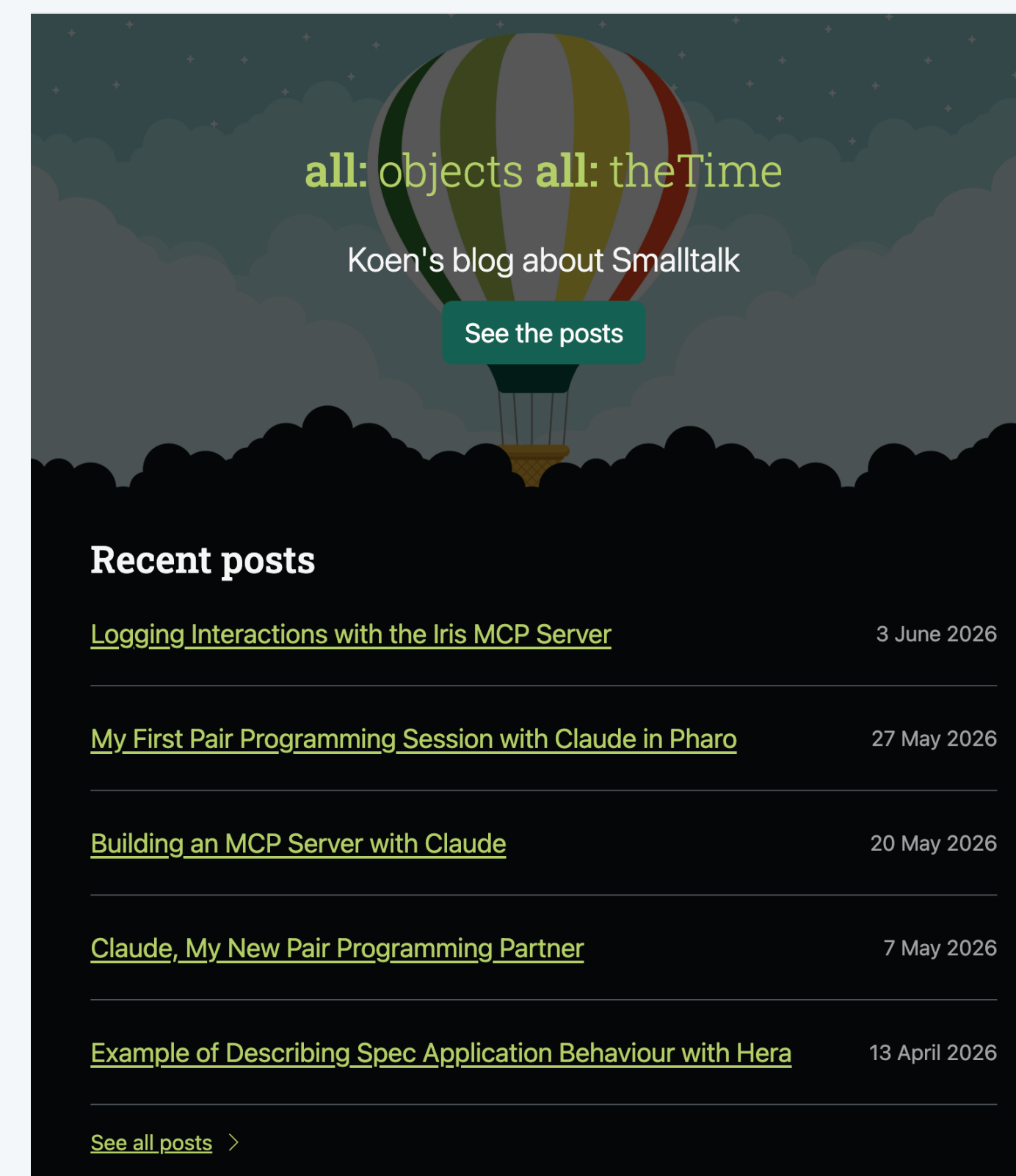
About Me

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

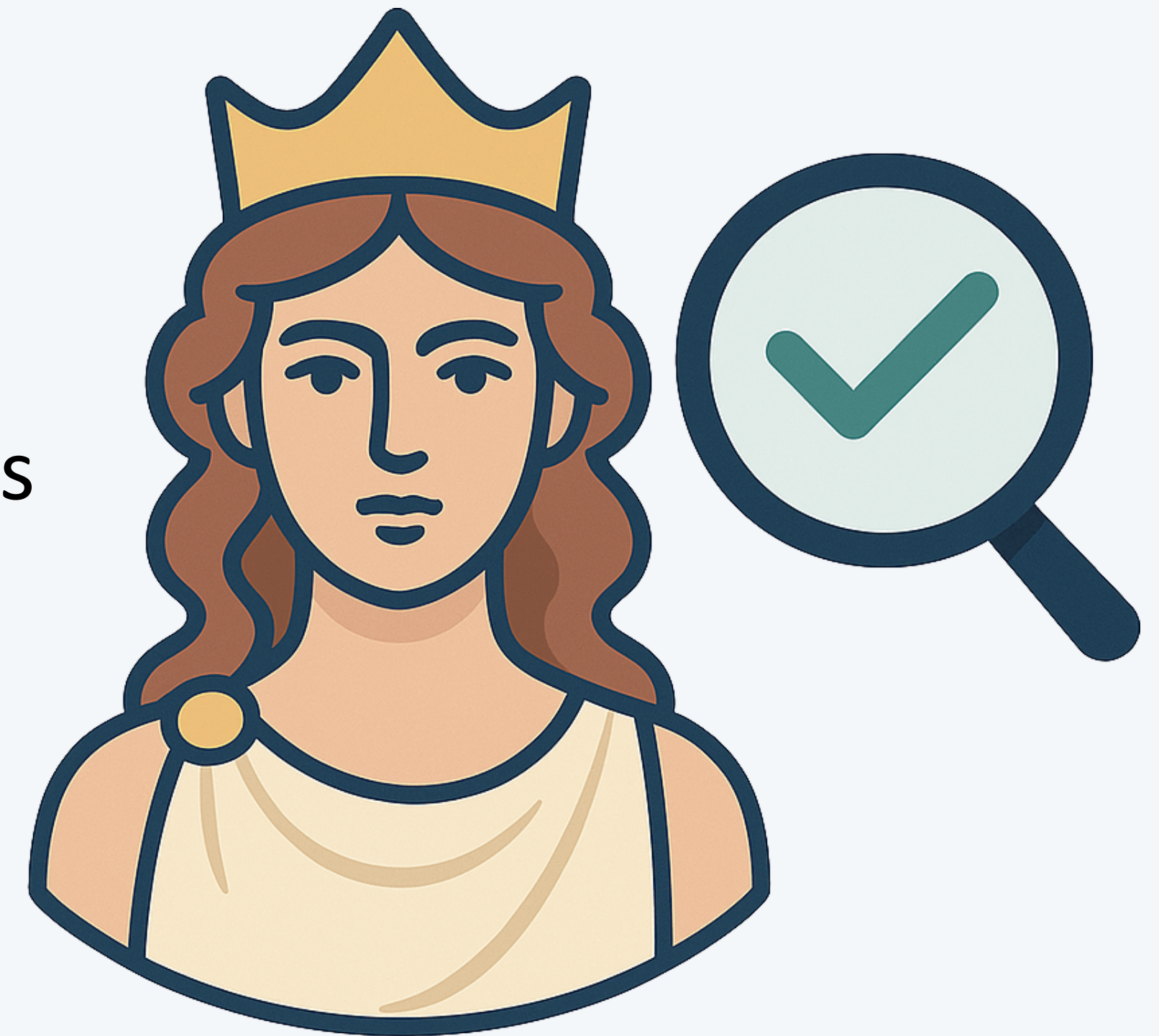
Gherkin features and steps are methods

Open source

Developed in Pharo

<https://all-objects-all-the-time.st/#/projects/hera>

<https://github.com/koendehondt/hera-for-pharo>



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"

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

Structured Step Arguments

Doc string

For multi-line text

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

Then I receive this receipt:

""""

Apples3,34 €

Green beans4,67 €

Cola8.40 €

Total16.41 €

""""

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 "x" 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 "x" 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 "x" 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 "x" 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 "x" 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

+

+

▶

▶

🐛

▶

Tags

↓

⬇

?

▼ FinderAcceptanceTest

Clearing behaviour

Result interaction

Search modes

Search options

Status bar

1

2

3

4

5

6

7

8

Feature: Search modes

Background:

Given the Finder is open

Scenario: Default

Then the "Search" button is enabled when the "Case" button is enabled

Then

the {string} drop list shows {string}

{}

1

1

stepTheDropListNamed: aDropListName shows:

expectedItemName

Finished 15/15

15

15

0

0

8 seconds (2026-06-07 14:36)

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}

{}

1

2

3

4

stepTheDropListNamed: aDropListName

shows: expectedItemName

| dropList |

dropList := self presenterAt:

aDropListName.

self

Defined in FinderAcceptanceTest (steps)

Feature Runner

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

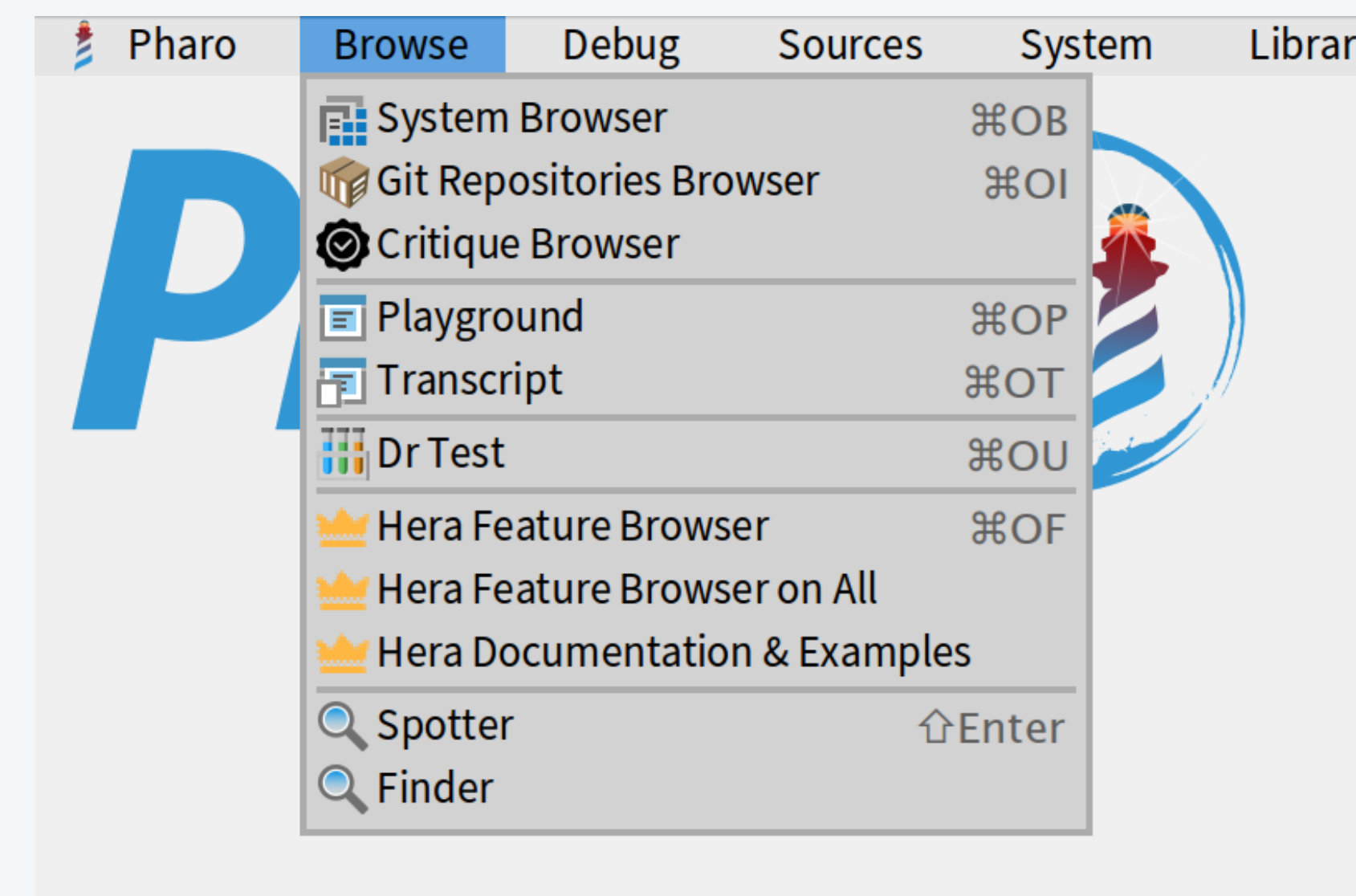
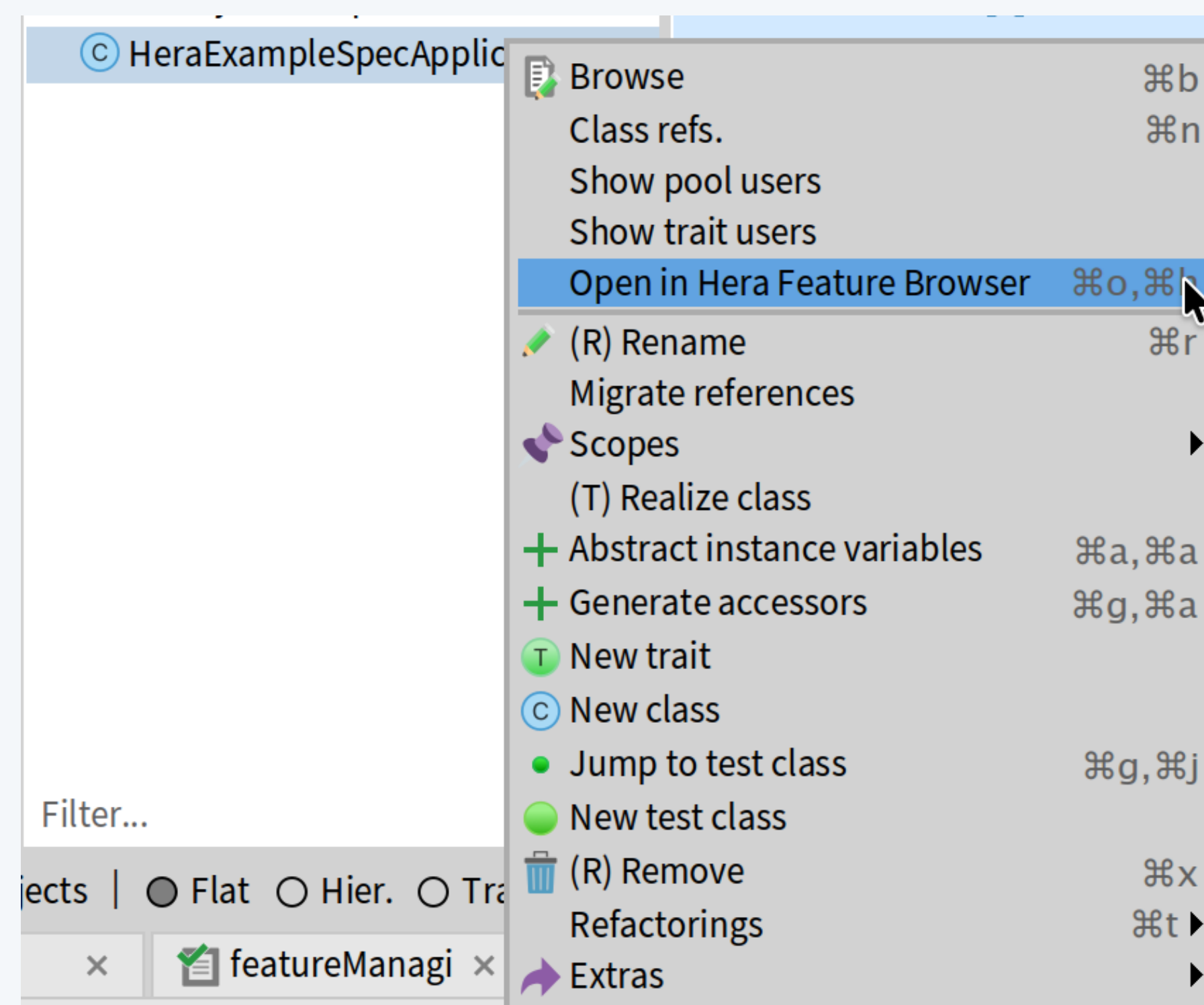
Each tree item starts with >
Multiple >>> indicate depth

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

HeraWebAcceptanceTest

HeraWebAcceptanceTestExample

Describing web page behavior

Old

Hera Feature Browser

1 Feature: Describing web page behavior

2

3 Scenario: Search Pharo on Wikipedia

4

5

6

7

8

9

When I open {}

1 stepIOpenURL: url

2

3 driver get: url

Managing a family tree

Tags

▼ Spec Application Example: Family Tree

Managing a family tree

1 Feature: Managing a family tree

2

3 Scenario: Building a family tree from scratch

4 When I start a new family tree

5 Then the "Family tree" window opens

6 When I press the "Add" button

7 Then I see the following "Family" tree:

8 > Unknown

9 When I fill in "John" in the "Name" field

10 And I select the "Male" radio button

11 Then I see the following "Family" tree:

12 > John

13 When I press the "Add" button

14 Then I see the following "Family" tree:

15 > John

16 >> Unknown

17 When I fill in "Jane" in the "Name" field

18 Then I see the following "Family" tree:

19 > John

20 >> Jane

21 When I select the "Female" radio button

22 And I select "John" in the "Family" tree

23 Then I see "John" in the "Name" field

24 And the "Male" radio button is on

25 And I see exactly the following items in the "Children" list:

26 | Jane |

27

When I press the {string} button {}

1 stepIPressTheButtonNamed: aButtonName

2

3 self click: aButtonName

Defined in HeraSpecAcceptanceTest (steps - buttons)

Feature Runner

× - □

Hera Runner

▶

↶

↷

↺

1

1

0

0

3 seconds (2025-07-04 21:43)

Feature: Describing web page behavior

Scenario: Search Pharo on Wikipedia
When I open `https://en.wikipedia.org/`
And I enter "Pharo" in the "Search Wikipedia" field
Then the current URL is `https://en.wikipedia.org/wiki/Pharo`
And I see a page with main header "Pharo"

Total: 1. Passed: 1. Failed: 0. Error: 0.

Old

× - □

Finished 13/13

▶

↶

↷

↺

13

13

0

0

4 seconds (2026-07-02 23:23)

- 1 feature (5s)
- Calculator** (5s)
 - ✓ The window opens with the correct title (95ms)
 - ✓ The display shows zero by default (88ms)
 - ✓ Pressing a digit appends it to the display (131ms)
 - ✓ Entering a multi-digit number (226ms)
 - ✓ Adding two numbers (276ms)
 - ✓ Subtracting two numbers (279ms)
 - ✓ Multiplying two numbers (291ms)
 - ✓ Dividing two numbers (297ms)
 - ✓ Operator precedence: multiplication before addition (426ms)
 - ✓ Clearing the display (326ms)
 - ✓ Entering a decimal number (508ms)
 - ✓ Toggling the sign of a number (231ms)
 - ✓ Using parentheses to override precedence (563ms)

Scenario: Entering a multi-digit number
Given the calculator is open [Background]
When I press the "1" button
And I press the "2" button
And I press the "3" button
Then the calculator displays "123"

Scenario: Adding two numbers
Given the calculator is open [Background]
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: Subtracting two numbers
Given the calculator is open [Background]

New

Describing Spec Applications

Managing a family tree

+

+

■

■

▶

▶

⏏

▶

Tags

↓

?

✓

✕

■

✓

✕

▼ Spec Application Example: Family Tree

Managing a family tree

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

Feature: Managing a family tree

Scenario: Building a family tree from scratch

When I start a new family tree

Then the "Family tree" window opens

When I press the "Add" button

Then I see the following "Family" tree:

> Unknown

When I fill in "John" in the "Name" field

And I select the "Male" radio button

Then I see the following "Family" tree:

> John

When I press the "Add" button

Then I see the following "Family" tree:

> John

>> Unknown

When I fill in "Jane" in the "Name" field

Then I see the following "Family" tree:

> John

>> Jane

When I select the "Female" radio button

And I select "John" in the "Family" tree

Then I see "John" in the "Name" field

And the "Male" radio button is on

And I see exactly the following items in the "Children" list:

| Jane |

When

▼

I press the {string} button

{}

▼

1

2

3

stepIPressTheButtonNamed: aButtonName

self click: aButtonName

Defined in HeraSpecAcceptanceTest (steps - buttons)

DSL for Describing Spec Applications

Referring 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

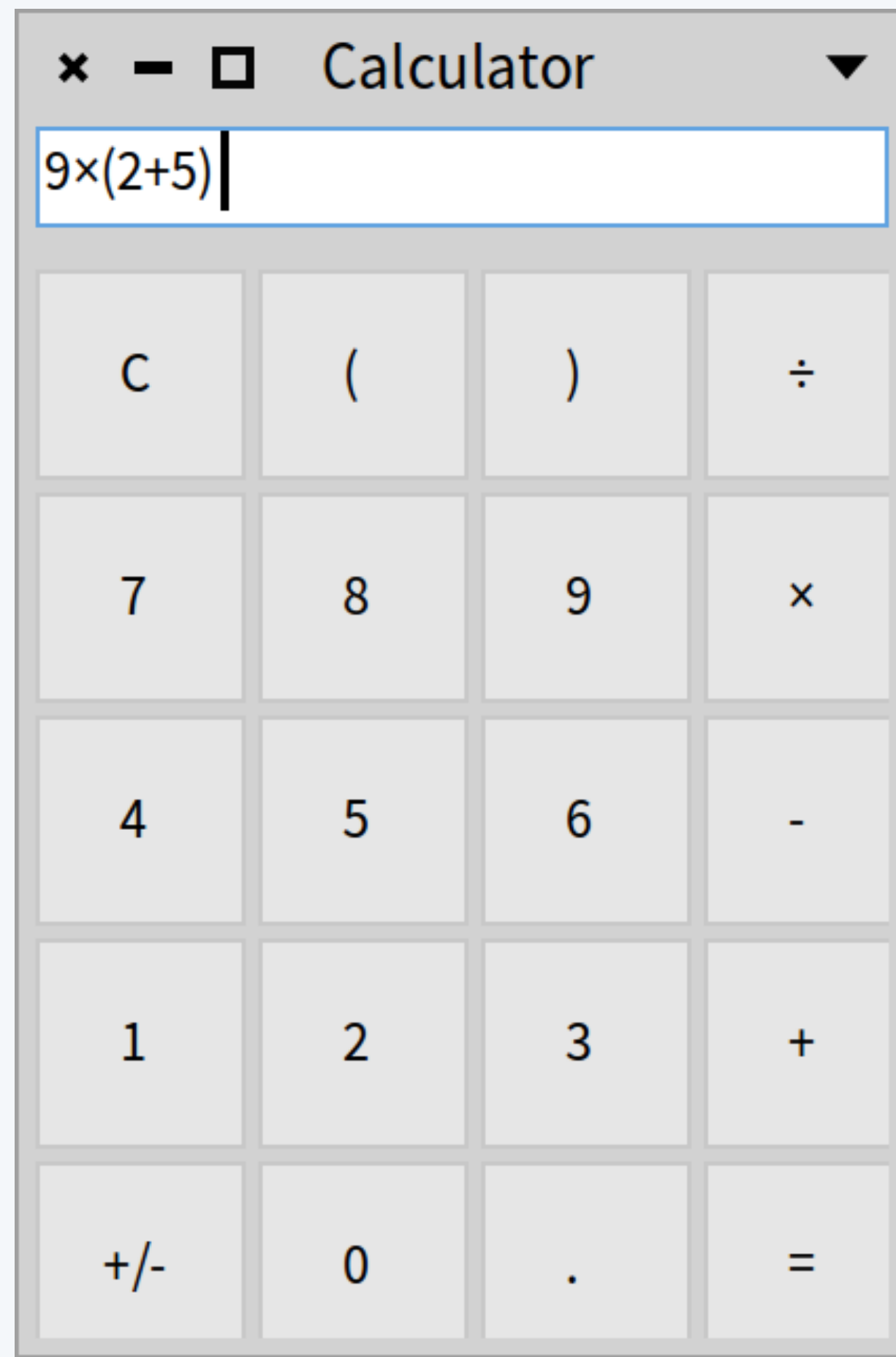
Examples

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

The screenshot shows the Hera IDE interface. On the left is the 'Hera Feature Browser' sidebar with a tree view containing categories like 'Basic browsing', 'Hera Runner', and various acceptance tests. The main area is divided into two panes. The top pane, titled 'Basic browsing', contains a list of features with line numbers 1 through 29. The bottom pane shows the code for the selected feature, 'stepISeeAnEmptyListOfFeatures', which includes a comment and a code snippet using 'self' and 'assertEmptyTree:'. The rightmost pane shows the 'Then' clause of the step, 'I see an empty list of features', with a dropdown menu set to 'I see an empty list of features'.

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 "x" 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

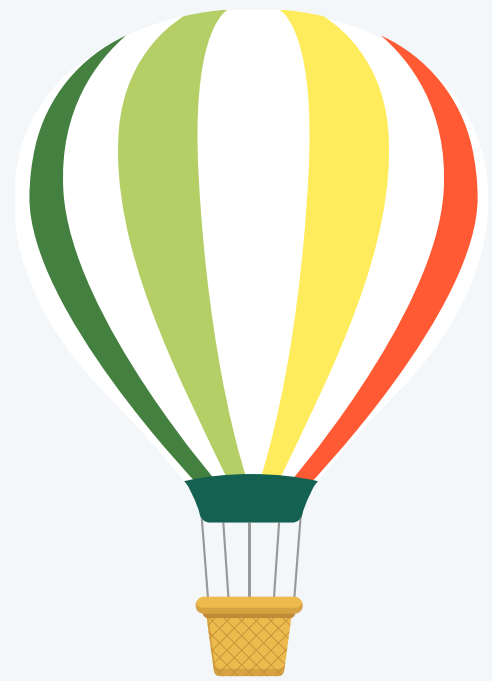
Thank you!

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