



Kostenloses eBook

LERNEN

pharo

Free unaffiliated eBook created from
Stack Overflow contributors.

#pharo

Inhaltsverzeichnis

Über	1
Kapitel 1: Erste Schritte mit Pharo	2
Bemerkungen.....	2
Examples.....	2
Installation oder Setup.....	2
Kapitel 2: Klasse erstellen	3
Einführung.....	3
Examples.....	3
Klasse in einem Systembrowser hinzufügen.....	3
Trivialer Klassenerstellungscode.....	4
Anonyme Klasse.....	5
Kapitel 3: Strand	6
Einführung.....	6
Examples.....	6
Schlagbar.....	6
Credits	8



You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [pharo](#)

It is an unofficial and free pharo ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official pharo.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

Kapitel 1: Erste Schritte mit Pharo

Bemerkungen

Pharo bietet eine hervorragende Möglichkeit, der **Komplexität** in der Softwareentwicklung entgegenzuwirken. Mit seiner schnellen Rückmeldung und der sofortigen Kompilierung der aktuell bearbeiteten Methode, der Objekte vom Compiler bis zum Runtime-Stack und Tools, die vom einzelnen Entwickler anpassbar sind, unterstützt er die Programmierung von Fehlern. Es verfügt über hervorragende Refactoring-Tools und macht es leicht, domänenspezifische Sprachen zu erstellen. Sie legt Wert auf Innovation und Effizienz gegenüber der Laufzeiteffizienz.

Examples

Installation oder Setup

Einfach zu installieren

Gehen Sie zu <http://pharo.org/download> und wählen Sie einen passenden Download aus, und führen Sie ihn aus.

Einzelheiten

Es gibt viele Möglichkeiten, Pharo zu installieren. Pharo selbst besteht aus einer VM und einem Bild. Außerdem benötigt es seine Quellen und Plugins und hat einige Abhängigkeiten:

- Es handelt sich um eine plattformübergreifende Umgebung, die unter OS-X (und iOS), Windows und mehreren Unix-Varianten (ua Ubuntu und Android) ausgeführt wird.
- Es läuft auf einer virtuellen Maschine, die auf mehreren Prozessorarchitekturen (Intel, ARM) laufen kann. Die virtuelle Maschine wird mit Squeak, Cuis und Newspeak geteilt. Mit Pharo 5 wurde eine neue und viel schnellere VM mit einem anderen Bildformat und FFI eingeführt.
- Es gibt 32-Bit- und 64-Bit-Varianten.
- Neben dem Standardimage gibt es den PharoLauncher, der in unsere CI-Infrastruktur integriert ist und das Herunterladen und Ausführen aller Arten von Bildern unterstützt, darunter vorkonfigurierte Küsten-, Magritte- und Elchbilder, ältere Versionen und die neuesten Entwicklungsversionen von Pharo.

Erste Schritte mit Pharo online lesen: <https://riptutorial.com/de/pharo/topic/8799/erste-schritte-mit-pharo>

Kapitel 2: Klasse erstellen

Einführung

Das Erstellen einer Klasse in Pharo ist so einfach wie das Senden einer `subclass:` Nachricht an ein anderes Klassenobjekt. Die meisten Klassen werden jedoch in einem Systembrowser (derzeit Nautilus) erstellt und geändert.

Examples

Klasse in einem Systembrowser hinzufügen

Die gebräuchlichste Methode zum Hinzufügen (und Bearbeiten) von Klassen über den Systembrowser Im Systembrowser von Nautilus haben Sie nichts ausgewählt oder ein Paket (erste Spalte) oder eine Klasse (zweite Spalte) ausgewählt. Je nach Auswahl zeigt der Code-Editor eine etwas andere Klassenvorlage an:

Auswahl	Vorlage
Keiner	Leere Klassenvorlage
Paket	Eine Klassenvorlage mit vorgefülltem Paketnamen (basierend auf dem ausgewählten Paket)
Klasse	Eine tatsächliche Definition der ausgewählten Klasse

Das folgende Bild zeigt das Nautilus-Fenster mit einem ausgewählten Paket:

Navigation pane showing a project structure and a class list.

Buttons: Scoped, Variables

Type: Pkg1|^Pkg2|Pk.*Core\$

- BlueInk-Extras
- BlueInk-Tests
- CodeExport
- CodeImport**
- CodeImportCommandLi
- Collections-Abstract
- Collections-Arithmetic
- Collections-Atomic
- Collections-DoubleLinke
- Collections-DoubleLinke
- Collections-Native
- Collections-Sequenceab
- Collections-Stack

- ChunkFileFormatParser
- ChunkReadStream
- CodeChunk
- BehaviorOwnedChunk
- ClassCommentChunk
- ClassOrganizationChunk
- MethodChunk
- DoltChunk
- StyleChunk
- CodeImporter
- FileCompilerRequestor

Buttons: Hier., Class, ?

```
Object subclass: #NameOfSubclass
  instanceVariableNames: ''
  classVariableNames: ''
  package: 'CodeImport'
```

. Wenn Sie keine bestimmte Superklasse in Ihrem Kopf haben, wird empfohlen, die Unterklasse von `Object` Dies führt jedoch zu einer tautologischen Verwirrung im vorherigen Beispiel. Angenommen, Sie möchten eine Klasse `PriorityStack` mit der Klasse `Stack` als Superklasse erstellen, dann müssen Sie Folgendes auswerten:

```
Stack subclass: #PriorityStack
```

Anonyme Klasse

Sie können Klassen ohne Namen erstellen, die nicht im System installiert sind, indem Sie `newAnonymousSubclass` an eine Klasse senden.

Zum Beispiel

```
anonymousSet := Set newAnonymousSubclass
```

ordnet der `anonymousSet` Variablen eine anonyme Unterklasse von `Set` zu. Dann können Sie Methoden in dieser Klasse kompilieren und instanzieren oder mit einer echten Klasse austauschen.

Nützlich für Testressourcen zum Proxying

Klasse erstellen online lesen: <https://riptutorial.com/de/pharo/topic/8834/klasse-erstellen>

Kapitel 3: Strand

Einführung

Seaside ist ein Web-Framework für Pharo und andere Smalltalks. Es ist ideal für komplexe Anwendungen mit einem umfangreichen Domänenmodell.

Examples

Schlagbar

Eine Seaside-Komponente (Unterklasse von `WComponent`) muss `#renderContentOn:` überschreiben. Es ist eine Smalltalk-Klasse, die alle üblichen Methoden zum Strukturieren einer Anwendung verwenden kann. Hier wird auf drei verschiedene Arten delegiert.

```
JQDroppableFunctionalTest>>renderContentOn: html
  self renderInstructionsOn: html.
  self renderInventoryOn: html.
  self renderSelectedOn: html
```

Als Parameter wird ein HTML-Canvas-Objekt abgerufen, das Nachrichten versteht, die für den Aufbau von HTML und Javascript relevant sind. Es verwendet eine fließende Schnittstelle, wobei `#with:` die letzte Nachricht ist, die an den aktuellen Canvas-Kontext `#with:` wird.

```
JQDroppableFunctionalTest>>renderInventoryOn: html
  html div class: 'inventory ui-corner-all'; with: [
    self colors do: [ :each |
      html div
        class: each;
        passenger: each;
        script: (html jQuery new draggable
          revert: true) ] ]

JQDroppableFunctionalTest>>renderSelectedOn: html
  html div
    class: 'selected ui-corner-all';
    script: (html jQuery new droppable
      onDrop: (html jQuery this load
        callbackDroppable: [ :event |
          self selected add: (event at: #draggable) ]);
      html: [ :r | self renderSelectedItemsOn: r ]));
    with: [ self renderSelectedItemsOn: html ]

JQDroppableFunctionalTest>>renderSelectedItemsOn: html
  self selected do: [ :each |
    html div
      class: each;
      passenger: each;
      script: (html jQuery new draggable
        onStop: (html jQuery this effect
          percent: 1; puff;
```



```
onComplete: (html jQuery this parent load html: [ :r |  
  self selected remove: each ifAbsent: [ ].  
  self renderSelectedItemOn: r ]))) ]
```

Strand online lesen: <https://riptutorial.com/de/pharo/topic/8805/strand>

Credits

S. No	Kapitel	Contributors
1	Erste Schritte mit Pharo	Community , Stephan Eggermont
2	Klasse erstellen	Uko
3	Strand	Stephan Eggermont