

Dynamic Languages – Übungen Blatt 9

Abgabedatum: 29. June 2010

Aufgabe 1 – Methods

So far, when sending a message the receiver cannot be obtained by the method, only the implicit parent, which is the object where the method was defined. To fix this, the method should get access to its receiver: Every time a method is called it should get the **self** attribute set to its receiver. This is tested in the file `test_method.py`.

Aufgabe 2 – Builtins

Primitives are usually not used by the user directly. Instead they are put into methods which are then used. For example, the `$int_add` primitive would typically be called by the `add` method of integer objects. This `add` method lives on the integer trait `inttrait` which exists in the builtin module. This module can contains all sort of useful objects, methods and traits. To make its content easily accessible, every other module should get this module as its `__parent__`.

The content of the builtin module is defined by some code in the language itself, which can optionally be passed to the `Interpreter` constructor. If this argument is not given, a default builtin module should be constructed. There are tests about the intended behaviour in `test_builtin.py`.

Aufgabe 3 – Default Builtins

Start implementing the default builtin module. It should contain at least an `inttrait` object with some sensible methods (like `add`, `sub`, `mul`, and `div`). Write tests for those methods. Think of some other sensible things to add into the builtin module, write tests for them and implement them.

Aufgabe 4 – Primality Check

Implement a simple primality checker in your programming language. It should be implemented together with tests runnable by your interpreter. Add primitives and `inttrait` methods as needed.