The SKilL Language V1.0

Timm Felden

TR 2017/01

Abstract

This document defines the Tyr, a research language for type-oriented programming. Type oriented programming is a paradigm that extends on object-oriented programming. In type-oriented languages, types are first order values like integers and objects. An existing but primitive form of type orientation is the Java reflection API.

Acknowledgements

For Pony!

Contents

Part I
Core Language

var/val: fields type var -> type field (in vtable)
defs: def -> virtual static def -> static type (ada non-poly pointer) type def -> type method
Typen: Any (top) void (<: Any) bool Integer int byte long UnsignedInteger FloatingPoint float double pointer
class Object <: pointer String <: Object IterableOnce <: String Iterable <: IterableOnce Option <: Iterable Seq <: Iterable Array <: Iterable

Part II
Compilation

modules, source paths, modules scopes, default scopes,
module naming convention: <organization>.<project> tyr.lang tyr.system tyr.collection skill.common

Part III
Collections

IterableOnce(T : Type) - static def for (p, b) - def foreach (f : LocalLambda[-> T])
Iterator <: IterableOnce - empty() - move() : bool - get() - for (p, b) = if(!empty) do

1 Introduction

Part IV
Appendix