编程语言设计和实现
—— 课程简介

冯新宇

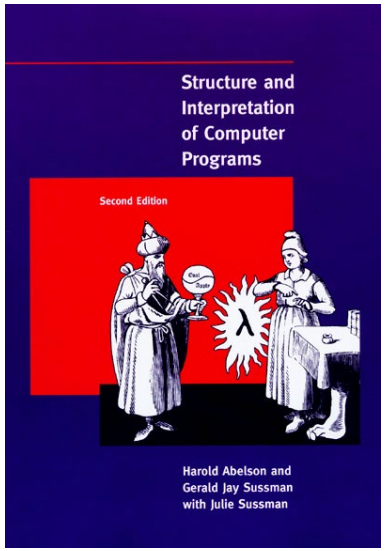
南京大学

什么是编程语言?

... the technology for coping with large-scale computer systems merges with the technology for building new computer languages, and **computer science itself becomes no more (and no less) than the discipline of constructing appropriate descriptive languages**.

... 应对大规模计算机系统的技术与构建新型计算机语言的技术相融合，**计算机科学自身也就不多不少成为一门构建恰当的描述性语言的学科**。

-- Chapter 4 of SICP



编程语言是连接现实世界和计算机世界的桥梁

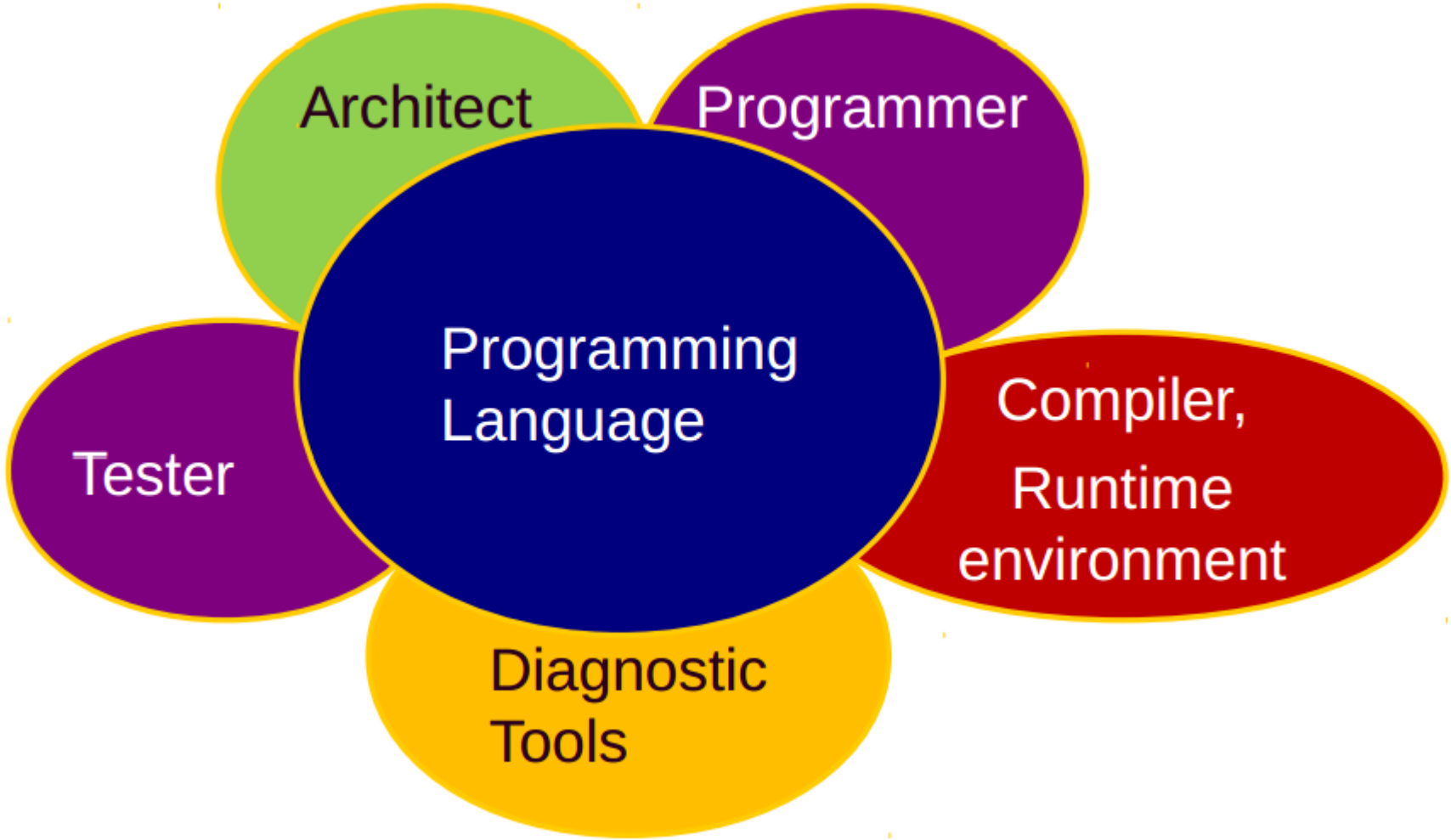
什么是编程语言?

- 人与计算机交流的语言
  - 交流的内容：计算、通信、世界的建模
  - 大模型时代，自然语言可否作为编程语言?
- 计算机科学中最基础的领域之一
  - 与操作系统、体系结构等基础领域并列
    - 首届SOSP: 1967；ISCA: 1973；POPL: 1973
- 无论是工业界还是学术界，仍然非常活跃的领域
  - 新的工业界语言层出不穷
    - Swift, Kotlin, Rust, Dart, Zig, Mojo, F#, 仓颉，...
  - DSLs: R, Matlab, tensorflow, Triton, P4, Agent DSLs ...

编程语言关注什么?

- 易用性
  - **语言设计**：语法、语义、抽象级
  - 工具支持：构建、调试、测试等
  - 提升软件开发效率：领域专用语言（domain specific languages ）
- 高性能
  - 语言实现：编译器/解释器，运行时，并行化等
- 好的软件质量（可靠性、安全性）
  - 类型安全、内存安全、并发安全
  - 动静态检查、形式化验证
- 理论基础
  - 形式化语义、形式化验证等
    - 与其他领域的交叉：逻辑、代数、计算理论、形式语言与自动机等

编程语言和周边的关系



图片来源于Stanford CS242 (2012) 课件

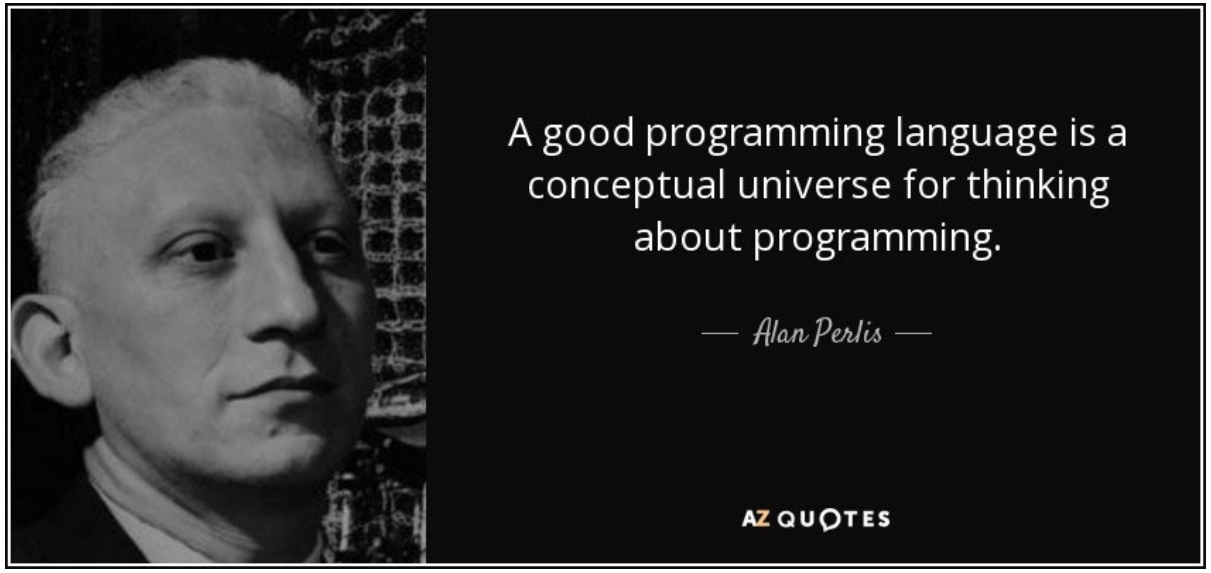
PL领域图灵奖得主

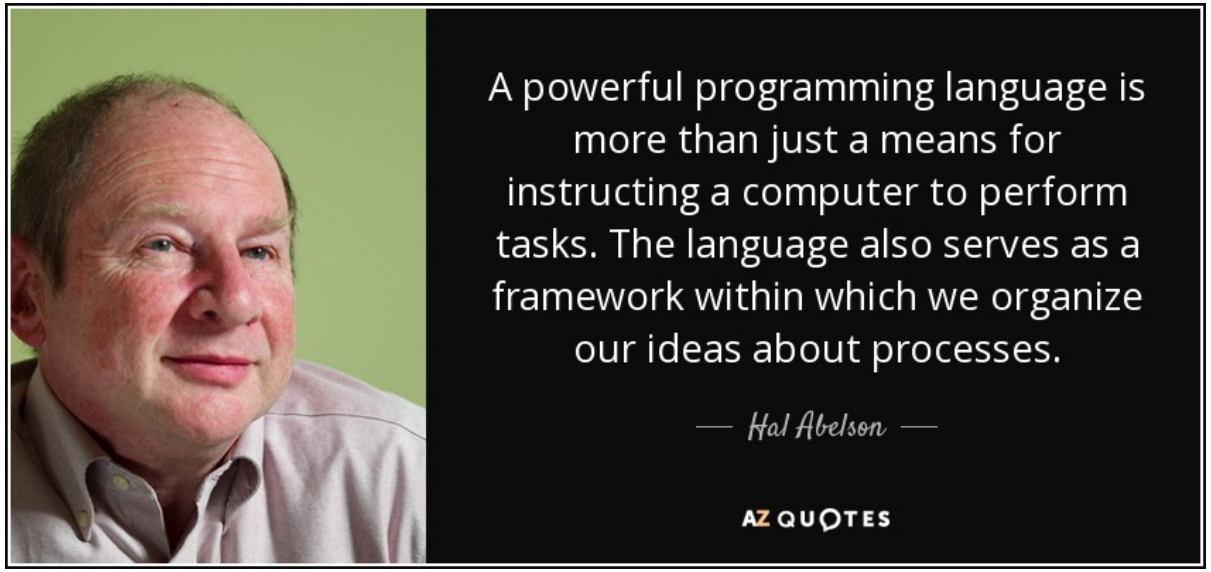
为什么选这门课？

公众号“码农翻身”：
《两年，我学会了所有的编程语言！》

为什么选这门课?

- 深入理解编程语言的基本概念
  - 深入现象看本质
  - 举一反三，“一法通万法通”
- 更优秀的编程能力
  - 写出更加高效、可靠的代码
- 设计新的语言
  - 和库、框架一样，
    语言也是一种解决问题的重要手段
  - 设计自己的编程语言并非梦想
  - 领域专用语言的需求非常广泛





课程内容

基本概念和语言设计 → 代数数据类型、OO等

编译前端实现 → IR设计、类型推断、宏 …

关键运行时和虚拟机技术 → 对象格式、GC、协程和任务管理 …

教学大纲（1）

- **第一部分：程序设计语言分类、发展历史、重要性介绍等**
- **第二部分：经典编程范式和核心概念**
  - 编程范式：命令式、面向对象、函数式、逻辑式等
  - 核心概念：作用域、高阶函数和闭包、代数数据类型、模式匹配、继承和动态派遣 ...
- **第三部分：高级语言特性**
  - 多态机制（重载、参数化多态/泛型、子类型）、类型扩展、类型系统和类型推断
  - 元编程机制：宏（动态、静态）、反射、注解等
  - 并发并行编程模型：异构编程、内存一致性模型等

教学大纲（2）

- **第四部分：程序设计语言实现原理（部分覆盖）**
  - 程序运行模式和实现机制：编译器、虚拟机（解释器、JIT）、端侧AOT、GPO等
  - 编译前端IR设计和分析优化
  - 对象模型设计
  - 内存管理机制：Sweep-GC、引用技术、Rust ownership机制等
  - 任务管理机制：用户态轻量级线程（go/java语言的线程模型）、无栈协程（Rust Tokio, C++, C#, JavaScript等语言的线程模型）
- **第五部分：高级类型系统、编程最新前沿和发展趋势等（大概率没时间讲）**
  - 线性类型系统，及其应用：Rust的内存安全机制
  - 渐进定型理论，及其应用：TypeScript以及类似语言的类型系统
  - 依赖类型理论，及其应用：Coq等定理证明工具和程序验证
  - …

课程信息

- **课程内容设置：以仓颉语言的设计和实现为蓝本**
  - 递进式不断丰富语言特性，
    最终得到一个类仓颉（子集）的语言
  - 部分内容邀请仓颉团队的工程师现场授课
  - 但不局限于仓颉：同时系统介绍除了仓颉之外的各种技术路线

- 课程实践选用仓颉作为开发语言
  - 基于仓颉语言开发课程实验

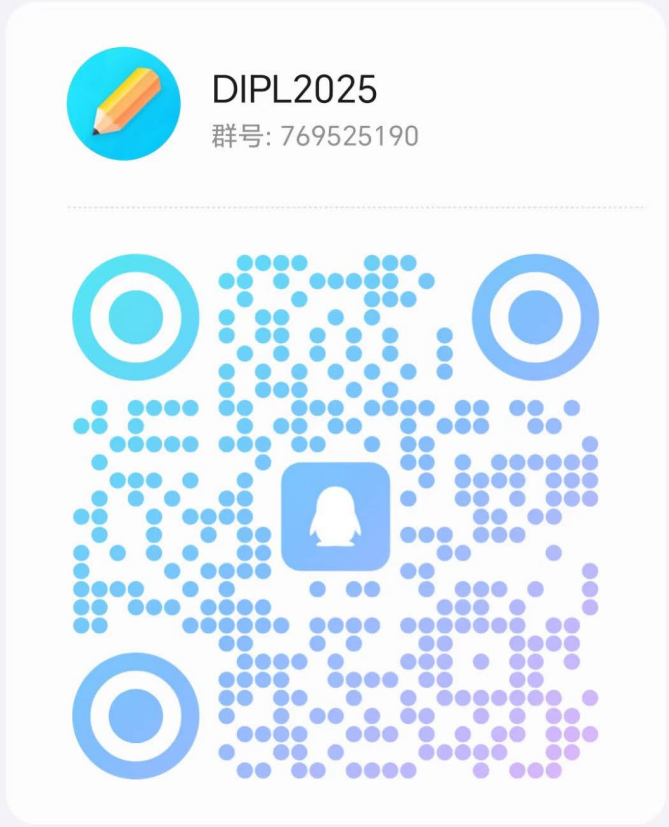
“看大轮子 + 用大轮子 + 造小轮子”

课程信息

- 课程实践
  - 编程实现各种语言机制的解释器，加深对理论课中介绍的操作语义的理解；
  - 通过一个设计型的期末项目，鼓励学生发挥创造力，设计一个小型的程序设计语言，并实现其语义检查、字节码设计和生成、以及字节码解释器

- 课程**考核**方式
  - 平时作业 + 平时Projects + 期中考试 + 期末Project

课程QQ群



群号：769525190