

Pythy 语言开发小组

小组章程

目录

一．成员名单及联系方式.....	2
二．项目概要	2
三．小组组织方式	3
四．小组工作方式	4
五．小组管理措施	5
有效性.....	5

一．成员名单及联系方式

姓名	学号	邮箱
胡宬	2016301500363	2038412353@qq.com
罗溥晗	2017300030006	2017300030006@whu.edu.cn
莫会民	2017300030054	747799598@qq.com
章博文（组长）	2017300030010	bwzhang@whu.edu.cn

二．项目概要

本小组的项目是实现编程语言的解释器以及一个配套的集成开发环境(IDE), 小组成员为胡宬、罗溥涵、莫会民、章博文, 指导老师为武汉大学计算机学院伍春香副教授。

在编程语言的选取上, 本小组拟定实现一个弱类型的命令式脚本语言 **Pythy** (Python Hongyi), 其语法与 Python 相似。将会使用 Java 语言来编写, 在特性上会初步实现以下几个方面:

- (1) 基本数据类型 (整数、实数)
- (2) 基本运算 (算术运算、逻辑运算)
- (3) 赋值语句
- (4) 条件判断、循环语句。此处对 True Value 和 False Value 的判断将会和 Python 保持一致
- (5) 函数、高阶函数、柯里化等特性 (依照实际开发进度决定)
- (6) IO 语句或 IO 函数
- (7) 在不实现面向对象的情况下, 实现运算符重载 (依照开发进度决定)

根据实际项目的需要, 本小组将整个项目分为三个模块:

- (1) 编译部分, 包括词法、语法、语义的分析, 最后为执行部分提供抽象语法树、控制流图 (如果有必要)。使用 Java 语言的编译器开发工具 JavaCC 编写。
- (2) 执行部分, 实现编程语言的运行环境, 根据 AST 逐条运行代码。使用 Java 面向对象方法编写。
- (3) IDE 部分, 实现一个能够编辑并且查看运行结果的集成开发环境。使用 Java 的 GUI 库 Swing 或 C++ 的 Qt 或基于 Web 开发。

在此基础上, 下一个部分是小组的组织形式。

三．小组组织方式

依照软件工程书中关于软件小组的组织的论述,本小组的组织方式主要遵循敏捷过程小组方法,同时结合民主小组方法,组织方式如下:

- (1) 一个模块的设计与代码编写,将会由两个人共同完成。一个成员主要负责代码的编写,另一个成员主要负责文档的编写、测试程序以及和其他模块沟通接口。如果整个开发周期很长,可以考虑这两个成员的角色定期互换。
- (2) 对于成员的分配方式,本组拟定,保证每一个模块都有一个相对有经验的组员,这样可以保证整体的效率,并且同一个模块内的成员可以在相互的学习过程中有更多的收获。
- (3) 民主小组方法的思想体现在,一个模块的两个成员共同接受奖惩。因此一个成员完成了自己的工作后,有义务去帮助另一个成员完成工作。比如说,一个模块的checkpoint代表着两个人的工作必须在checkpoint之前都完成,不然就都接受惩罚。

之所以采用这样的组织方式,出于各个成员此前的经验教训,大家之前经历过的很多分工合作,大多数其实只落实到了“分工”,在“合作”方面其实做的不太好。分到了任务以后,缺乏很有效的沟通机制,最后导致两个现象:

- (1) 当遇到特别棘手的问题,由于大多数时候只能自己解决,因为别的同学也忙于自己的部分,最后导致自身产生挫败感,效率一度非常低下。
- (2) 当自己的部分顺利完成后,便觉得“大功告成了”,然后对于其他成员的困难,由于自己不了解也不参与,往往给予不了很多帮助。同时自己的那部分也不一定能和其他组员的部分很好地配合。

四．小组工作方式

本章节主要介绍工作中的迭代流程以及总体的时间线。

迭代流程

本项目将遵循“沟通+开发+测试”三步进行迭代式开发，三个步骤具体实施如下：

沟通阶段。在各个成员都思考之后，组织一次全员的会议，使得所有人能够明确本阶段的任务，需要通过什么样的测试，并大致讨论模块之间的接口。之后，由两个模块各派出一个代表，由这两个人共同商讨具体的设计方案与接口，然后回去和自己的成员交代清楚。我们认为第二次讨论的作用非常大，因为在全员会议中往往只能讨论出一些大致的内容，而小规模会议能够讨论出更加明确的结果。

开发阶段。这个阶段中，负责文档、沟通、测试的成员先负责完成大致的接口（类的大致定义、函数签名等），以及相应的测试程序。前者负责和编码的成员沟通落实该部分的思想，后者开始编写代码，前者负责写文档以及构思下一阶段的内容。编码者在编写代码中产生问题，首先和自己组的成员沟通，如果发现还有问题，再和另一组的联络员沟通，直到问题解决。

测试阶段。模块通过内部的测试后，两部分合并测试，合并测试的过程可以由全员参与，通过后上传 github 备份。

时间线

正如上文所述，本项目将依据“沟通+开发+测试”三步进行迭代式的开发，每一个迭代的周期，完成一部分的工作。

对于各个模块任务的优先级，我们前期要着重于编译模块和执行模块的开发，可以根据上一章工作组织的方式，按照两两分为编译小组和执行小组，并行地完成工作。而对于 Ide 的开发，将会被归到后期（前期开发中，有余力的成员可以学习 GUI 相关的框架）。

因此，根据项目实际的情况，大略将时间线分为以下几步：

1. 整体规划阶段：组织会议，目标是使每一个组员都大概了解编程语言怎么运作，怎么写，对整个项目有一个整体的感知，之后开始讨论总体的框架。希望通过这个讨论能产生的成果：即该编程语言的测试样例（testBench）。
2. 快速构建阶段：本阶段是一个从 0 到 1 的过程，目标将最简单的语句“x=1”跑通。这样的意义是，编译和执行的两个模块可以获得一个参考，打消“我们写的代码会不会和旁边对不上”的念头，同时也通过最简单的例子了解另一个模块的运作。之后便可以在此基础上完成更多的功能。
3. 重复迭代阶段：将会在多个迭代中，逐个实现该编程语言的特性，可以按照 数据类型、表达式、控制循环语句、函数…的顺序进行开发，直到基本特性都实现。
4. IDE 开发阶段+编程语言优化阶段：在此阶段，将会重新分配两个人小组，一组完成 IDE 的开发，另一组负责向编程语言中加特性，具体的任务分配将根据实际情况决定。
5. 收工阶段：完成最终的文档，以及整个项目的打包，文档的整合，开源。

五．小组管理措施

由于本小组人数较少，且大家都是同学，经组内初步讨论，决定采用比较民主公平的分数评价制来管理小组内的日常事务并予以奖惩。具体细则暂定如下：

1. 每人初始拥有 100 分，组长将在最后提交报告时根据每人最后的得分判断相对的贡献大小。
2. 每周召开组会，核对项目进度并讨论遇到的问题，规划下一步的工作。组会迟到者-5 分，在讨论时提出重要想法建议者+5 分。
3. 设定项目提交 ddl，推迟完成者每迟交一天-5 分，提前一天+5 分。
4. 在疫情期间，为了方便成员的交流，文件的共享通过 github 完成。考虑到项目比较小，用比较简单的方式进行同步，即每个人在项目下建一个自己名字的文件夹，将自己完成的、需要共享的代码或文档放到这个文件夹下，并且用一个.md 或.txt 文件交待一下各个文件是干什么的即可。

有效性

本章程经本组所有成员投票通过，自 2020 年 2 月 23 日起生效，在该项目生命周期内皆具有效力。