

程序设计语言







• 来自百度

- 程序设计语言用于书写计算机程序的语言。语言的基础是一组记号和一组规则。根据规则由记号构成的记号串的总体就是语言。在程序设计语言中,这些记号串就是程序。(一维)
- 程序设计语言有3个方面的因素,即语法、语义和语用。
 - 语法表示程序的结构或形式,亦即表示构成语言的各个记号之间 的组合规律,但不涉及这些记号的特定含义,也不涉及使用者。
 - 语义表示程序的含义,亦即表示按照各种方法所表示的各个记号的特定含义,但不涉及使用者。





语言的发展

自20世纪60 年代以来,世界上公布的程序设计语言已有上千种之多,但是只有很小一部分得到了广泛的应用。从发展历程来看,程序设计语言可以分为4代。

- ▶ 第一代机器语言
- ▶ 第二代汇编语言
- ▶ 第三代高级语言
- ▶ 第四代非过程化语言





• 第一代机器语言

机器语言是由二进制 0、1 代码指令构成,不同的 CPU 具有不同的指令系统。机器语言程序难编写、难修改、难维护,需要用户直接对存储空间进行分配,编程效率极低。这种语言已经被渐渐淘汰了。

• 第二代汇编语言

- 汇编语言指令是机器指令的符号化,与机器指令存在着直接的对应关系,所以汇编语言同样存在着难学难用、容易出错、维护困难等缺点。
- 但是汇编语言也有自己的优点:可直接访问系统接口,汇编程序翻译成的机器语言程序的效率高。从软件工程角度来看,只有在高级语言不能满足设计要求,或不具备支持某种特定功能的技术性能(如特殊的输入输出)时,汇编语言才被使用。





• 第三代高级语言

- 高级语言是面向用户的、基本上独立于计算机种类和结构的语言。其最大的优点是:形式上接近于算术语言和自然语言,概念上接近于人们通常使用的概念。高级语言的一个命令可以代替几条、几十条甚至几百条汇编语言的指令。因此,高级语言易学易用,通用性强,应用广泛。
- 高级语言种类繁多,可以从应用特点和对客观系统的描述两个方面对其进一步分类。





第三代高级语言

一、从应用角度分类

从应用角度来看,高级语言可以分为基础语言、结构化语言和专用语言。

(1) 基础语言

基础语言也称<mark>通用语言</mark>。它历史悠久,流传很广,有大量的已开发的软件库,拥有众多的用户,为人们所熟悉和接受。属于这类语言的有 FORTRAN、COBOL、BASIC、ALGOL等。

- FORTRAN 语言是目前国际上广为流行、也是使用得最早的一种高级语言,从 20 世纪90 年代起,在工程与科学计算中一直占有重要地位,备受科技人员的欢 迎。
- BASIC 语言是在 20 世纪 60 年代初为适应分时系统而研制的一种交互式语言,可用于一般的数值计算与事务处理。BASIC 语言结构简单,易学易用,并且具有交互能力,成为许多初学者学习程序设计的入门语言。





第三代高级语言

(2) 结构化语言

20世纪70年代以来,结构化程序设计和软件工程的思想日益为人们所接受和欣赏。出现了一些很有影响的结构化语言,这些结构化语言直接支持结构化的控制结构,具有很强的过程结构和数据结构能力。PASCAL、C、Ada语言就是它们的突出代表。

- PASCAL 语言是第一个系统地体现结构化程序设计概念的现代高级语言,软件 开发的最初目标是把它作为结构化程序设计的教学工具。由于它模块清晰、控 制结构完备、有丰富的数据类型和数据结构、语言表达能力强、移植容易,不 仅被国内外许多高等院校定为教学语言,而且在科学计算、数据处理及系统软 件开发中都有较广泛的应用。
- C语言功能丰富,表达能力强,有丰富的运算符和数据类型,使用灵活方便,应用面广,移植能力强,编译质量高,目标程序效率高,具有高级语言的优点。同时,C语言还具有低级语言的许多特点,如允许直接访问物理地址,能进行位操作,能实现汇编语言的大部分功能,可以直接对硬件进行操作等。用 C语言编译程序产生的目标程序,其质量可以与汇编语言产生的目标程序相媲美,具有"可移植的汇编语言"的美称,成为编写应用软件、操作系统和编译程序的重要语言之一。

但是,也带来了很多副作用



(3) 专用语言

是为某种特殊应用而专门设计的语言,通常具有特殊的语法形式。一般来说,这种语言的应用范围狭窄,移植性和可维护性不如结构化程序设计语言。随着时间的发展,被使用的专业语言已有数百种,应用比较广泛的有 APL 语言、Forth 语言、LISP 语言。

– APL是A Programming Language或Array Processing Language的缩写。肯尼斯·艾佛森1962年设计这个语言时他正在哈佛大学工作。为此他1979年获得图灵奖。





二、从客观系统的描述分类

(1) 面向过程语言

以"数据结构+算法"程序设计范式构成的程序设计语言,称为面向过程语言。前面介绍的程序设计语言大多为面向过程语言。

(2) 面向对象语言

以"对象+消息"程序设计范式构成的程序设计语言,称为面向对象语言。比较流行的面向对象语言有 Delphi、Visual Basic、Java、C++等。

- Delphi 语言具有可视化开发环境 , 提供面向对象的编程方法,可以设计各种具有 Windows 内格的应用程序(如数据库应用系统、通信软件和三维虚拟现实等),也可 以开发多媒体应用系统。
- Visual Basic 语言简称 VB,是为开发应用程序而提供的开发环境与工具。具有很好的图形用户界面,采用面向对象和事件驱动的新机制,把过程化和结构化编程集合在一起。它在应用程序开发中的图形化构思,无需编写任何程序,就可以方便地创建应用程序界面,且与 Windows 界面非常相似,甚至是一致的。
- Java 语言是一种面向对象的、不依赖于特定平台的程序设计语言,简单、可靠、可编译、可扩展、多线程、结构中立、类型显示说明、动态存储管理、易于理解,是一种理想的、用于开发 Internet 应用软件的程序设计语言。





主要语言

- 1. <u>习语言</u> 即中文版的C语言,确切的说是支持中文代码的C语言
- 2. <u>O语言</u> O语言是一款中文<u>计算机语言</u>(或称套装: O汇编语言、O中间语言、O高级语言)
- 3. APT(Automatically Pro-grammed Tools)——自动<u>数控</u>程序。第一个专用语言,用于数控机床加工,1956。
- 4. FORTRAN(FORmula TRANslation)——公式翻译程序设计语言。第一个广泛使用的<u>高级语言</u>,为广大科学和<u>工程技术人员</u>使用计算机创造了条件,1956。
- 5. <u>FLOW-MATIC</u>。第一个适用于商用<u>数据处理</u>的语言,其语法与英语语法类似,1956。
- 6. IPL-V(Information Processing Language V)——信息处理语言。第一个表处理语言,可看成是一种适用于表处理的假想计算机上的汇编语言,1958。
- 7. COMIT(COmpiler Massachusetts Institute foTechnology)——<u>马萨诸塞州理工学院编译</u>程序。第一个现实的串处理和模式匹配语言,1957。
- 8. COBOL(COmmon Business Oriented Language)——面向商业的通用语言。使用最广泛的商用语言,它是适用于<u>数据处理</u>的高级程序设计语言,1960。
- 9. <u>ALGOL60</u>(ALGOrithmic language60)——<u>算法语言</u>60。程序设计语言由技艺转向科学的重要标志,其特点是局部性、<u>动态</u>性、<u>递归</u>性和严谨性,1960。
- 10. <u>表处理</u>语言(**LIS**t **Proceessing**——**LISP**。引进<u>函数式程序设计</u>概念和<u>表处理</u>设施,在<u>人</u>工智能的领域内广泛使用,1960。





主要语言

- 11. JOVIAL(Jules Own Version of IAL)——国际<u>算法语言</u>的朱尔斯文本。第一个具有处理科学计算、输入-输出逻辑信息、数据存储和处理等综合功能的语言。多数JOVIAL编译程序都是用JOVIAL书写的,1960。
- 12. GPSS(General-purpose Systems Simulator)——通用系统模拟语言。第一个使模拟成为实用工具的语言,1961。
- 13. <u>JOSS</u> (Johnniac Open-Shop System ——。第一个交互式语言,它有很多方言,曾使分时成为实用,1964。
- 14. FORMAC(FORmula MAnipulation Compiler)——公式翻译程序设计语言公式处理<u>编</u> <u>译程序</u>。第一个广泛用于需要形式代数处理的数学问题领域内的语言,1964。
- 15. SIMULA(SIMUlation LAnguage)——模拟语言。主要用于模拟的语言,是 ALGOL60的扩充,1966。SIMULA67是1967年SIMULA的改进。其中引进的"类"概念 ,是现代程序设计语言中"模块"概念的先声。
- 17. PASCAL (Philips Automatic Sequence CALcul-ator)——菲利浦自动顺序<u>计算机语言</u>。 在ALGOL60的基础上发展起来的重要语言,其最大特点是简明性与结构化,1971。





主要语言

- 18. PROLOG(PROgrammingin LOGic)。一种处理逻辑问题的语言。它已经广泛应用于 关系数据库、<u>数理逻辑</u>、抽象问题求解、<u>自然语言</u>理解等多种领域中,1973。
- 19. ADA。一种现代<u>模块化语言</u>。属于ALGOLPASCAL语言族,但有较大变动。其主要特征是强类型化和模块化,便于实现个别编译,提供类属设施,提供<u>异常处理</u>,适于嵌入式应用,1979。
- 20. Python 最好的字符串处理<u>脚本语言</u>。
- 21. perl 广泛应用于unix/linux系统管理的<u>脚本语言</u>。
- 22. ruby 日本人设计的一种被广泛学习使用的<u>动态语言</u>。
- 23. C结构化程序设计语言的经典,它能完成你想要的一切。
- 24. C++ 构建大型软件,要求速度与性能,可以使用它。
- 25. JAVA SUN公司开发的一种基于JVM<u>虚拟机</u>的面向对象的语言,被广泛应用于移动设备。

PHP(Hypertext Preprocessor,中文名: "超文本预处理器")是一种通用开源脚本语言。语法吸收了C语言、Java和Perl的特点,利于学习,主要适用于Web开发领域。PHP独特的语法混合了C、Java、Perl以及PHP自创的语法。它可以比CGI或者Perl更快速地执行动态网页。



程序设计语言的选择

1、项目的应用领域

- (1)科学工程计算。需要大量的标准库函数,以便处理复杂的数值计算,可供选用的语言有FORTRAN语言、C语言等。
- (2)数据处理与数据库应用。SQL为IBM公司开发的数据库查询语言,4GL称为第4代语言。
- (3)实时处理。实时处理软件一般对性能的要求很高,可选用的语言有汇编语言、Ada语言等。
- (4)系统软件。如果编写操作系统、编译系统等系统软件时,可选用 汇编语言、C语言、Pascal语言和Ada语言。
- (5)人工智能。如果要完成知识库系统、专家系统、决策支持系统、推理工程、语言识别、模式识别等人工智能领域内的系统,应选择Prolog、Lisp语言。





程序设计语言的选择

2、软件开发的方法

有时编程语言的选择依赖于开发的方法,如果要用快速原型模型来开发,要求能快速实现原型,因此宜采用4GL。如果是面向对象方法,宜采用面向对象的语言编程。

3、软件执行的环境

良好的编程环境不但能有效提高软件生产率,同时能减少错误,有效提高软件质量。

4、算法和数据结构的复杂性

科学计算、实时处理和人工智能领域中的问题算法较复杂,而数据处理、数据库应用、系统软件领域的问题,数据结构.比较复杂,因此选择语言时可考虑是否有完成复杂算法的能力,或者有构造复杂数据结构的能力。

5、软件开发人员的知识

编写语言的选择与软件开发人员的知识水平及心理因素有关,开发人员应仔细地分析软件项目的类型,敢于学习新知识,掌握新技术。



Popularity, Interoperability, and Impact of Programming Languages in 100,000 Open Source Projects

翻译标题

十万个开源项目中编程语言的普及性互操作性及其影响

摘要

Tegawend' e F. Bissyand' e1, Ferdian Thung2, David Lo2, Lingxiao
Jiang2 and Laurent R' eveill`ere1
Laboratoire Bordelais de Recherche en Informatique, France
Singapore Management University, Singapore

37th Annual International Computer Software & Applications Conference (COMPSAC 2013), Kyoto : Japan (2013)

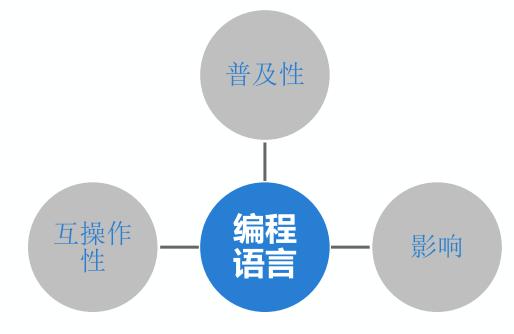


引言

"

对研究背景、内容、方法的介绍,以及总体概述

- 编程语言便于人类向计算 1 机描述指令、计算机理解 人类的指令
- 出现了成百上千种的编程 语言,它们有着不同性质 和特点,存在着各自优缺 点。



- 我们有必要研究各语言的普及性、互操作性(两个或多个语言在同一项目中使用时的交互程度)、影响。这可以帮助开发人员选择语言来使用和学习,并帮助管理者评估开发人员的编程技能池。
- 本研究基于十万个开源项目,通过对这些项目中出现的编程语言的某些指标进行统计与对比展开对语言普及性、互操作性和影响分析。本文在第四章中围绕以下几个问题展开对编程语言上述性质的分析。

问题1:

根据现实中软件项目的采用率,各种编程语言的普及性如何?

问题2:

有多少项目是使用两种(含)以上的编程语言编写的,每种语言之间的互操作性如何?

问题3:

所使用的编程语言和项目成功率之间有何相关性?

问题4:

所使用的编程语言和报告的问题数之间有何相关性?

问题5:

所使用的编程语言和开发团队规模有何相关性?

5

研究中,我们从GitHub获得100,000个GIT存贮库,构成数据集。我们通过分析每个GIT存储库找到软件代码,并计算各种语言所书写的代码行数,还获取了关于项目成功率、报告问题数和团队规模等各种信息。





准备工作 介绍编程语言,描述GitHub及其 提供的项目

编程语言

- 1. 编程语言定义人类与机器交流和互动的语法和语义。
- 2. 每种编程语言都属于某类别,根据语言的功能和面向对象的特点,编译和解释设计,脚本和执行目的等分类
- 3.分析什么样的编程语言受欢迎,一个语言对一个项目 发展的影响,这也是本文的主题。

GitHub

- 1. GitHub是一个被开发人员广泛使用的开源的项目托管网站,通过它提供的扩展API,可以看到大概13,000亿个内容公开的存储库。
- 2. GitHub使用git作为其源代码版本修订的改版控制系统;它提供分支并且集成了常用的软件开发工具。
- 3.通过GitHub的API,我们不仅获得GitHub社区或者git的项目的信息,还可以获得一些平台外开发却在GitHub上存在储存镜像的项目以及在GitHub上托管的项目信息

)

研究方法

"

我们以GitHub上涵盖了多个领域的十万个项目为样本,做了探究和考察,研究了项目的开发团队、应用领域和项目中所使用的各种编程语言。发现GitHub上的项目非常适合于我们的研究。

为了从不同的角度广泛调查不同编程语言的普及性和其影响,我们从这些项目中抽取了以下信息:

- 1. 项目通用信息。本文运用了守护着和分支的概念。守护者指关注项目的开发人员,是用衡量一个项目中兴趣与活动的度量值。它指明一个项目受开发者们关注的程度。而分支,衡量了开发者们在项目的发展过程中参与程度。
- 2. 源代码行数:我们下载了每个项目的源代码库,使用了SLOCCount工具并对扩展了它的功能,来计算项目中使用的各种编程语言代码真实行数。
- 3. 开发者贡献:研究考虑到项目的贡献者。开发者所参与的git的储存库中有他的信息。这对那些不在GitHub上开发却在GitHub上有镜像的项目很重要
- 4. 问题报告:问题报告也被视作软件开发过程中一个重要的产物。"问题"可能是指缺陷报告或功能需求,它标识出程序员社区对此项目的关注点。

四"

经验性评估

为了保证绝大多数项目从规模上适合我们研究,我们通过预先准备好的数据计算了每个项目的代码量(LOC)。结果如图1所示:

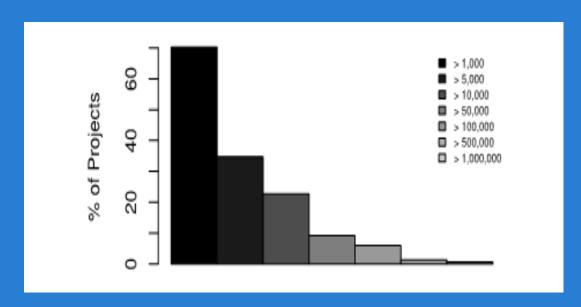


图1,不同代码量的项目所占比例

多于70%的项目有1,000以上 LOC,约35%的项目含5,000以上 LOC,而20%多的项目含10,000以上LOC。

这个分布表明我们数 据集中相当多的项目 都有一定的实际规模

0



Rank	Language	# LOC	% LOC
1	ansi c	1,615,634,331	60.83 %
2	javascript	296,893,761	11.18 %
3	c++	217,566,364	8.19 %
4	php	167,458,938	6.31 %
5	java	99,308,060	3.74 %
6	ruby	59,967,003	2.26 %
7	python	53,850,088	2.03 %
8	c#	31,560,343	1.19 %
9	lisp	27,614,150	1.04 %
10	sh	22,605,731	0.85 %
11	objective c	16,570,836	0.62 %
12	perl	16,413,762	0.62 %
13	pascal	11,766,801	0.44 %
14	erlang	7,335,480	0.28 %
15	yacc	1,646,328	0.06 %
16	ml	1,550,750	0.06 %
17	fortran	1,468,246	0.06 %
18	tel	1,351,073	0.05 %
19	haskell	1,117,902	0.04 %
20	jsp	972,759	0.04 %
21	ada	878,412	0.03 %
22	f90	588,801	0.02 %
23	lex	573,349	0.02 %
24	vhdl	485,806	0.02 %
25	expect	318,554	0.01 %
26	awk	199,513	0.01 %
27	cobol	135,257	0.01 %
28	csh	74,608	0.00 %
29	sed	26,549	0.00 %
30	modula3	985	0.00 %

表1依据项目中的每种编程语言代码量对各种编程语言进行排名。

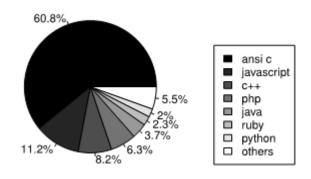


图2,语言普及性——代码行数

图 2展示了按代码量排在 前十位的语言和它们的代码量所占的比例。 BUAA SEI



Rank	Language	# Projects	% Projects
1	javascript	27,873	27.87 %
2	ruby	19,857	19.86 %
3	python	15,224	15.22 %
4	sh	14,444	14.44 %
5	php	11,023	11.02 %
6	java	10,646	10.65 %
7	ansi c	10,142	10.14 %
8	c++	6,865	6.87 %
9	perl	5,741	5.74 %
10	objective c	3,721	3.72 %
11	c#	3,082	3.08 %
12	lisp	2,005	2.01 %
13	pascal	1,165	1.17 %
14	haskell	1,071	1.07 %
15	jsp	958	0.96 %
16	yacc	887	0.89 %
17	erlang	652	0.65 %
18	awk	601	0.60 %
19	lex	567	0.57 %
20	sed	409	0.41 %
21	tcl	255	0.26 %
22	csh	242	0.24 %
23	ml	208	0.21 %
24	ada	195	0.20 %
25	fortran	134	0.13 %
26	expect	102	0.10 %
27	f90	77	0.08 %
28	vhdl	45	0.05 %
29	modula3	13	0.01 %
30	cobol	10	0.01 %

表2展示了使用某种语言的软件项目数。这也可以作为一种评价语言受欢迎程度的标准。

JavaScript以27,873个项目遥遥领先于其它编程语言,占我们调查的项目总量的27.87%。下面也是几种脚本语言,分别占了项目的19.86% (Ruby)、15.22% (Python)和14.44% (Shell)。紧跟着是通用编程语言,如Java、C、C++和C#,分别占项目的10.65%、10.14%、6.87%和3.08%。

Haskell函数型语言包含于1071个项目,占了1.07%的比例。其排名高于与它极为相似的函数型语言ML。排在最后的是VHDL、Modula3和Cobol,所有项目中只有1%与它们相关。





Rank	Language	# Projects
1	ruby	12,642
2	python	10,165
3	javascript	9,616
4	java	8,861
5	php	7,886
6	ansic	6,307
7	c++	4,477
8	sh	3,311
9	objective c	2,982
10	perl	2,472

表3:语言普及性——项目数排前十的 主要编程语言

注:如果一种语言在项目中的代码量 最多,我们就把它认为是该项目的主 要编程语言。 表3展示了某种语言被作为 主要编程语言使用的项目的 数量(前十位)。

对比表3和表4的内容,可看出脚本语言大多作为支撑语言使用。虽然JavaScript出现在28000个项目中,但将它作为主要语言的项目只有9000个(占含JavaScript代码项目总数的34.50%)。Shell脚本语言出现在超过14000个项目中

O手I O. . 左 EOW NI L 始 T 日 由

可编译的编程语言,

BUAA SEI 例如Java



Rank	Language	# Developers	% Developers
1	ansi c	48,373	22.08%
2	ruby	34,878	15.92%
3	python	29,764	13.58%
4	java	27,567	12.58%
5	c++	26,708	12.19%
6	php	22,790	10.40%
7	javascript	19,651	8.97%
8	sh	11,345	5.18%
9	perl	7,587	3.46%
10	c#	7,499	3.42%
11	objective c	6,405	2.92%
12	lisp	3,644	1.66%
13	haskell	2,026	0.92%
14	erlang	1,757	0.80%
15	pascal	1,161	0.53%
16	ml	408	0.19%
17	tcl	213	0.10%
18	jsp	151	0.07%
19	fortran	75	0.03%
20	yacc	67	0.03%
21	ada	48	0.02%
22	vhdl	48	0.02%
23	f90	28	0.01%
24	sed	8	0.00%
25	expect	7	0.00%
26	1ex	5	0.00%
27	awk	4	0.00%
28	csh	3	0.00%
29	modula3	3	0.00%
30	cobol	1	0.00%

图3体现了在项目中使用某种语言的 开发者数目和比例。表4中是详细统 计。C语言在开发人员中最受欢迎。 接下来是Ruby、Python、Java和C++ 。JavaScript和C#只被8.97% 和 3.42% 的开发者使用。

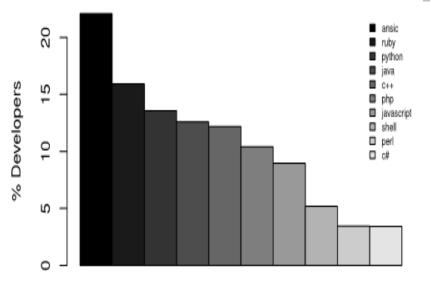


图3: 语言普及性 使用各种编程语言的 开发者数目和所占比例

表4:语言普及性——使用各种编程语言的 开发者数目和所占比例



经过上述图表的分析,我们可以对研究问题1即编程语言的 普及性得出**结论:**

结论:标准C、JavaScript和C++在我们的LOC排行中占据前列。我们数据集中绝大部分的项目包含JavaScript、Ruby和Python语言。最终,使用标准C、Ruby和Python的项目开发者最多。结果表明,我们应该好好学习这些语言。





在同一个项目中,开发人员通常使用不同的编程语言来实现不同的功能。第二个研究问题调查了在多语言项目中广泛使用的编程语言,以分析编程语言的互操作性。我们假设出现在同一个项目中的两种语言是"可交互的"。以此来探究编程语言之间的互操作性。



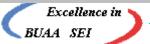


研究问题 2: 多语言项目

Rank	Language	# Multi Language Projects	% Multi Language Projects
1	javascript	18,353	21.81%
2	sh	12,456	14.80%
3	ruby	9,782	11.62%
4	python	7,757	9.22%
5	ansi c	7,646	9.09%
6	php	6,160	7.32%
7	c++	4,577	5.44%
8	java	4,163	4.95%
9	perl	3,841	4.56%
10	objective c	1,593	1.89%
11	c#	1,391	1.65%
12	lisp	1,239	1.47%
13	pascal	958	1.14%
14	jsp	925	1.10%
15	yacc	884	1.05%
16	awk	598	0.71%
17	lex	567	0.67%
18	haskell	464	0.55%
19	sed	405	0.48%
20	erlang	394	0.47%
21	tel	243	0.29%
22	csh	241	0.29%
23	ada	187	0.22%
24	ml	153	0.18%
25	fortran	129	0.15%
26	exp	101	0.12%
27	f90	71	0.08%
28	vhdl	33	0.04%
29	modula3	10	0.01%
30	cobol	9	0.01%

表6通过多编程语言项目中使用某 语言的项目数目来探究语言的互 操作性。排行第一的是JavaScript 在超过18000个(21.81%)多语 言项目中被采用。接下来是Shell (14.80%)、Ruby (11.62%)和 Python (9.22%)。C语言、PHP、 C++和Java分别占了所有多语言项 目的9.09%、7.32%、5.44%和 4.95%。Haskell 和 ML 是交互性 最差的,在项目中分别占0.55%和 0.18%。交互性最少的语言是 VHDL、Modula3和Cobol,它们 只在低于1%的多语言项目中出现

表5:语言互操作性—多编程语言项目 中使用某语言的项目数目及其百分比





研究问题 2: 多语言项目

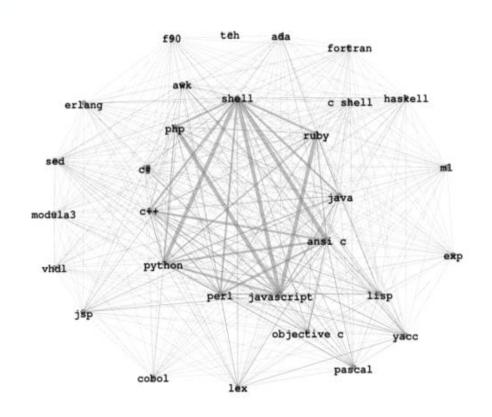


图4:语言互操作性——多编程语言项目中使用某语言的项目数目及其百分比

通过观察语言之间交互的细节 ,我们得到图4,该关系图给出 了一个大致的视图, 说明语言 之间在项目中是如何交互的。 图中,两个节点之间的边越粗 , 同时包含着两种语言的项目 越多。我们观察到, 所有语言 均有交互,只是交互度不同而 已。Shell是与其他语言交互最 多的语言。Objective C表现出 与其他语言更低的交互性。 Haskell和ML和其他语言的关 系非常之弱。



CO B 研究问题 2: 多语言项目

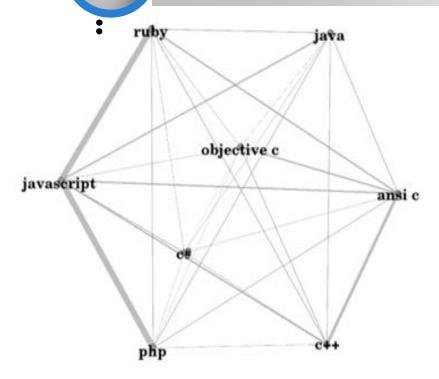


图5:语言交互性——主流语言之间的 关系和其交互性

图5研究了主流语言子集的互操作性。图中显示JavaScript与另外两个常用的网络编程语言——Ruby和php,有很好的交互性。

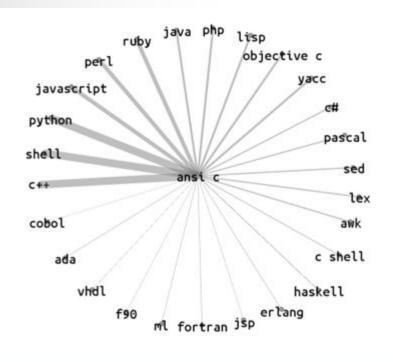


图6: 语言交互性——标准C和其他语言之间的交互性

图6中我们探索了标准C和其他所有语言之间的关系,得知C和其他很多语言的互操作性都很好。我们注意到,C语言和C++的交互性最好。

经过在研究问题2中图表的分析,对多语言项目中各编程语言的互操作性得出结论:

结论: JavaScript、Shell和Ruby会与其他绝大多数的语言一起使用。标准C和它的衍生语言们,语法构成非常接近,他们之间关联性也很强。网络编程语言经常在项目中同时使用

0





在第三个研究问题中,我们调查了成功的项目所采用的语言。而我们通过该项目受人关注的程度来衡量一个项目成功与否。本小节通过守护者与分支两方面来解析这一点。





研究问题 3: 编程语言与成功率

1、守护者

Rank	Language	Mean	Median	Significance
1		91.6	-	
2	objective c	65.8	5 2	+
3	javascript		2	+
	ruby	60.5	5	+
4 5	erlang	52.6		+
	sh	28.8	1	+
6	python	26.8	1	-
7	php	24.8	2	-
8	yacc	24.3	1	-
9	c#	23.6	1	+
10	c++	23.1	1	
11	ansi c	19.7	1	+
12	lisp	17.3	1	+
13	java	16.8	1	-
14	ml	15.5	1	-
15	tcl	15.0	2	-
16	perl	12.2	1	-
17	haskell	10.0	1	-
18	fortran	9.9	1	
19	f90	9.5	1	_
20	ada	8.9	1	+
21	pascal	7.2	1	_
22	lex	3.0	2	+
23	jsp	2.8	1	_
24	awk	2.2	2	_
25	csh	2.0	3	
26	vhdl	1.9	í	
27	sed	1.8	1	_
28	exp	1.0	i	+
29	modula3	1.0	1	+
30	cobol	1.0	1	+

表6呈现了使用了某种编程语言代码的项目的守护者的平均数量。此时,我们认为多语言项目中只有一种主语言会影响守护者。

Objective-C为主语言的项目是守护者最多的,平均值91。中值数也最大,为5。Erlang守护者的中值数也是5,但使用Objective-C的项目数是Erlang的六倍多。JavaScript和基于Ruby的守护者平均数量也多。主流编译语言中,C#和C++排在最前,它们平均值为23。以C和Java作为主语言的项目分别有19和16个守护者。Modula3和Cobol又一次排行垫底。

表6:语言与项目的成功——使用不同语言 的项目的关注者平均值和中值数目



研究问题 3: 编程语言与成功率

2、分支

Rank	I	Mann	Median	Ci-mi6
Kank	Language	Mean	Median	Significance
1	objective c	12.1	2	+
2	ruby	10.9	1	+
3	erlang	9.9	1	+
4	javascript	8.9	1	+
5	c#	6.4	1	-
6	php	6.1	1	+
7	sh	5.7	1	+
8	python	5.6	1	+
9	c++	5.2	1	+
10	java	5.1	1	+
11	lisp	4.4	1	-
12	ansi c	4.2	1	-
13	ada	3.7	1	-
14	pascal	3.6	1	-
15	tcl	3.0	1	-
16	perl	2.9	1	-
17	ml	2.8	1	-
18	yacc	2.6	1	-
19	fortran	2.4	1	-
20	haskell	2.4	1	-
21	lex	2.3	2	-
22	f90	1.6	1	-
23	csh	1.5	2	-
24	jsp	1.4	1	-
25	vhdl	1.4	1	-
26	exp	1.0	1	-
27	sed	1.0	1	+
28	awk	1.0	1	+
29	modula3	1.0	1	+
30	cobol	1.0	1	

表7:语言与项目的成功——使用不同语言的项目的分支数平均值和中值数目

分支有助于指出哪些没有固定组织的开发人员自愿参与项目开发。表7呈现了每个项目的每种语言的分支数据。

Objective C仍然以平均12个分支和2的中值排行第一。Lex和C-Shell语言与Objective-C中值相同,但是它们实际是用于相对较少的项目中(对比表3)。JavaScript和Ruby同样排得很靠前。总的来说,用分支的方法所得结果与用关注者方法测得的结果趋向相同。

我们使用Mann-Whitney-Wilcoxon (MWW),来评估表6和表7中某语言与紧邻着它的下一个语言的项目分布之间的统计学意义。



经过在研究问题2中图表的分析,对多语言项目中各编程语言的互操作性得出结论:

结论:使用Objective C和Ruby的项目更加能引起人们的兴趣。一个关于使用Objective C的项目的快速抽样调查表明,他们中有很多是用来做iOS(iPhone)应用的。





为了调查编程语言和问题的关系,我们计算了每个项目所用每种语言的问题数的平均值和中值。其中我们只考虑那些在GitHub上提交了至少一个问题的项目。





研究问题 4: 编程语言与问题报告

Rank	Language	Mean	Median	Significance
1	c++	64.4	6	-
2	ml	55.3	3	-
3	c#	50.6	9	
4	fortran	46.7	8	-
5	pascal	46.3	3	-
6	php	41.8	7	-
7	ruby	40.3	8	+
8	java	37.8	6	-
9	ansi c	36.7	10	-
10	python	35.7	7	-
11	sh	31.1	6	-
12	erlang	30.7	10	-
13	lisp	30.5	5	-
14	haskell	30.1	4	+
15	javascript	29.3	8	+
16	objective c	23.3	6	+
17	perl	21.9	3	-
18	tel	20.8	7	-
19	yacc	15.0	10	-
20	ada	13.0	25	-
21	jsp	5.3	2	-
22	f90	3.0	1	-
23	vhdl	2.0	2	
24	csh		-	
25	sed	-	-	
26	awk	-	-	
27	lex	-	-	
28	exp	-	-	
29	cobol		-	
30	modula3	-	-	

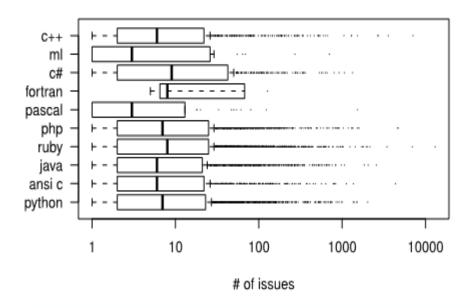


图7:编程语言和使用这种语言的项目报错数的平均值和中值

图7用箱线图的形式列出了问题数前十的语言分布。抛开Pascal、Fortran和ML外其他语言的箱线基本相似。这表明编程语言性质和它的报错数之间并没有明显联系。

表8:编程语言和使用这种语言的项目报错数的平均值和中值

依据编程语言报错数的图和表及其统计学省的显著性检验 ,我们可以得出结论:

结论:项目错误报告数目与该软件所应用的编程语言并无关联。





最后,我们根据开发团队使用的语言,调查了其开发团队的组成,计算使用每种编程语言的项目中开发人员数量的平均值和中值。 从而确定哪些语言是更"合作友好的"。





研究问题 5: 编程语言与团队规模

Rank	Language	Mean	Median	Significance
1	ansi c	46.5	1	-
2	c++	9.4	1	-
3	tel	7.7	2	-
4	erlang	6.1	2	+
5	ruby	5.3	1	+
6	sh	4.8	1	-
7	java	4.7	1	-
8	perl	4.7	1	-
9	c#	4.5	1	-
10	php	4.3	1	+
11	python	4.1	1	+
12	lisp	4.1	1	-
13	ml	4.1	1	-
14	haskell	3.7	1	-
15	pascal	3.5	1	-
16	fortran	3.2	1	-
17	ada	3.0	1	-
18	objective c	2.9	1	+
19	javascript	2.8	1	
20	yacc	2.4	2	-
21	exp	2.3	2	-
22	vhdl	2.0	1	-
23	jsp	1.8	1	-
24	f90	1.8	1	-
25	lex	1.7	1	-
26	sed	1.6	1	-
27	csh	1.5	2	-
28	modula3	1.3	1	-
29	awk	1.0	1	+
30	cobol	1.0	1	

表9:语言和团队规模。以不同编程语言来做的项目团队规模的平均值和中值

。C语言以每个项目平均46个 开发人员位居第一,但中间值 是1。C++排名第二,就在项目 中出现情况而言,另一个流行 的语言Ruby排名第五。

Tcl、Erlang、Yacc、Expect 和C-Shell的中间值都很高,为 2,但是它们用在一小部分项目 中(参照表3),从统计学角度 看,该发现不太可靠。Ruby、 Python和JavaScript项目的团队 规模分布差异明显,平均每个 项目大约分别有5个(排名第5)、4个(排名第10)和3个(排名第19)开发人员。

CO/E

研究问题 5: 编程语言与团队规模

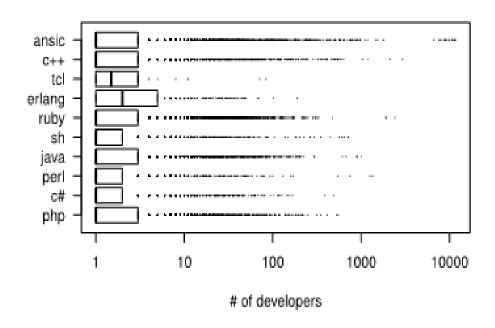


图8: 使用某种编程语言的项目的团队规模 前十位

在图8中,箱线图显示,团队规模分布和语言可能没有明显的关联。这些显著性测试也证实了一个事实,对于一小部分语言,尽可能少的平均值差异在统计学上是显著的。



依据编程语言团队规模的相关图和表分析,得出以下结论

结论:拥有最大的平均团队规模的语言中,前三位是标准C、C++和TCL。标准C的高平均值得益于我们数据集中有大量的、高分布开发的项目,例如Linux,这些项目中的编码使用标准C

Excellence in BUAA SEI

五

"

对研究正确性的威胁因素

- 数据集来源:我们的实验研究结果是基于在GitHub上找到的开源项目,这样产生的结果可能并不能概括开发者生产的软件项目的体系。我们还没有考虑仍在使用的传统项目,以及在GitHub上没有积极发展的项目。
- 数据集规模:该研究仅限于100000个项目。虽然样本相当大,但是它不能等同于数以百万计的在万维网上可以获得到代码的软件程序。

然而需要指出的是,我们在100000个项目上的出现语言研究结果是相似于GitHub提供的排名。这表明,我们的样本数据集对GitHub项目体系是有代表性的。

相关工作

已经有许多工作开始研究编程语言的流行度。列出编程语言相对容易,而就流行程度来排名各个语言是更具挑战性的。

有两个依据语言流行度的对编程语言进行排行的榜单:

TIOBE软件搜索公司的TIOBE编程社区索引

TIOBE的榜单并非是根据使用该语言编写的代码量来衡量哪个是最好的编程语言。他们基于一些热门搜索引擎的数据,呈现了以编程语言的热度为指标的排名。

Drew Conway的编程语言年流行度排名

Drew Conway是第一个从GitHub获取信息以调查编程语言普及性的人。他直接依赖于GitHub上以用到每种语言的项目数为基础的流行度排名。

相比于以上两份排名,在我们的研究中,考虑到了一种语言相比于其他语言在编程中的真实使用情况。



```
作业:查以上两个排行榜,前10名的语言,哪些是编译型语言,哪些不是,为什么?
```

