

РЕФЕРАТ

ИСТОРИЯ РАЗВИТИЯ ПОПУЛЯРНЫХ ЯЗЫКОВ ПРОГРАММИРОВАНИЯ.

Выполнили: Калакуцкий А.В., гр.: 4125
Мартьянов С.С., гр.: 4125
Толстиков Н.С. гр.: 4120

1 Аннотация

Мы живем в очень интересное время. Время, когда у практически у каждого есть мобильный телефон в кармане, персональный компьютер на рабочем столе, телевизор на кухне и практически неограниченное «облачное» хранилище информации. Время, когда со всех сторон нас окружают машины. А что же превращает эти машины из груды железа в вещи, к которым мы так привыкли?

Эту важную роль на себя берет программное обеспечение. ПО играет роль души, которая делает разнообразные устройства так близки нам.

Для того что бы сделать программное обеспечение наиболее удачным, нужен хороший инструмент. Инструментом для создания ПО является язык программирования.

На сайте The Language List¹ сейчас представлено около 2500 языков, но сколько из них реально используется и почему только несколько их них получили широкое распространение в среде программистов?

В нашей работе мы поставили цель, выяснить какие факторы привели к появлению и распространению современных наиболее популярных и интересных нам языков программирования. Для достижения цели нужно ответить на вопросы: как развивались языки программирования, какие области программного обеспечения они охватывали, какие парадигмы программирования поддерживали и какие факторы повлияли на популяризацию данного языка программирования. Для ответов на данный вопрос нужно решить следующие задачи. Первое, какие языки имеют наибольший успех среди программистов. Второе, составить хронологическую модель развития языков. И третье рассмотреть каждый язык в отдельности, выделить его основные характеристики и определить факторы, из-за которых он получил свою репутацию.

2 Рейтинги популярности языков программирования.

Существует множество рейтингов и способов оценки языков программирования. Мы в своей работе рассмотрим три из них:

ohloh.net Language Comparison

TIOBE Programming Community Index

The RedMonk Programming Language Rankings

2.1 TIOBE Programming Community Index

<http://www.tiobe.com/index.php/content/company/GeneralInfo.html>

Компания TIOBE занимается оценкой качества программных систем на основе оценки этих систем на соответствие своим стандартам программирования.

http://www.tiobe.com/index.php/content/paperinfo/tpci/tpci_definition.htm

Для попадания в рейтинг TIOBE язык программирования должен отвечать двум требованиям:

На википедии должна быть страничка посвященная этому языку.

Язык программирования должен быть полным по Тьюрингу. Соответственно такие языки как HTML, XML, SQL не учитываются данным рейтингом. Но различные расширения языка SQL, например PL/SQL, входят в рейтинг.

Рейтинг рассчитывается на основе подсчета количества результатов в выдаче поисковых систем при запросе «название языка» programming". Данное число домножается на специальный коэффициент, установленный для каждой используемой поисковой системы:

¹Киннерсли Б. "Language List" официальный сайт Канзаского университета
URL:<http://people.ku.edu/~nkinners/LangList/Extras/langlist.htm> дата обращения: 18.09.2012

Google: 30%
Blogger: 30%
Wikipedia: 15%
YouTube: 9%
Baidu: 6%
Yahoo!: 3%
Bing: 3%
Amazon: 3%

Количество результатов в выдаче определяет рейтинг языка. Рейтинги нормализуются по сумме первых пятидесяти языков. Т.о. первые 50 языков в сумме будут иметь рейтинг 100%. После для каждого результата подсчитывается количество ложных совпадений, таких как выдачи по запросу "Basic programming" страницы "Improve your basic programming skills in Java". Подсчитывается процент таких промахов для первых ста страниц выдачи, дальше этот процент вычитается из общего количества результатов.

Так же вычисляются статусы языков на основе изменения их рейтинга, но нас это мало волнует.

<http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html>

- 1 c
- 2 Java
- 3 Objective-c
- 4 c++
- 5 c#
- 6 PHP
- 7 (Visual)Basic
- 8 Python
- 9 Perl
- 10 JavaScript

2.2 ohloh.net Language Comparison Page

Ohloh — публичный ресурс посвященный свободному и открытому программному обеспечению. Он предоставляет возможности для общения разработчиков, аналитику и поиск по проектам. <http://meta.ohloh.net/us/> На странице проекта можно вывести и просмотреть график отражающий популярность языков. График показывает результаты для языков выбранных пользователем. Значения на графике — это сумма все коммитов за месяц, которые содержат хотя бы одну строчку на запрашиваемом языке. Если в коммите несколько языков, то он зачитывается в пользу всех. Информация собирается уже более 20 лет, и не включает текущий месяц, т.к. он еще не закончился.

meta.ohloh.net/compare_languages/ самые популярные языки

- 1 c
- 2 java
- 3 c++

- 4 python
- 5 javascript
- 6 php
- 7 ruby
- 8 c#
- 9 perl
- 10 objective-c

я специально исключил неполные по тьюрингу языки (html, sql, css) чтобы добиться соответствия результатам различных рейтингов.

2.3 The RedMonk Programming Language Rankings

<http://redmonk.com/>

<http://redmonk.com/about/>

RedMonk — аналитическая компания сфокусированная на разработчиках [1].

Их методика составления рейтинга весьма и весьма проста, впервые она была предложена Дрю Конвеем (Drew Conway) [2] в 2010 году. С тех пор уже второй год компания RedMonk обновляет статистику.

рейтинг представляет из себя график осями которого является: популярность на GitHub и популярность на Stack Overflow. Для расчета популярности на Stack Overflow ведется подсчет тэгов. А на GitHub есть свой собственная система подсчета количества проектов на каждом из языков.

В результате получается картинка:

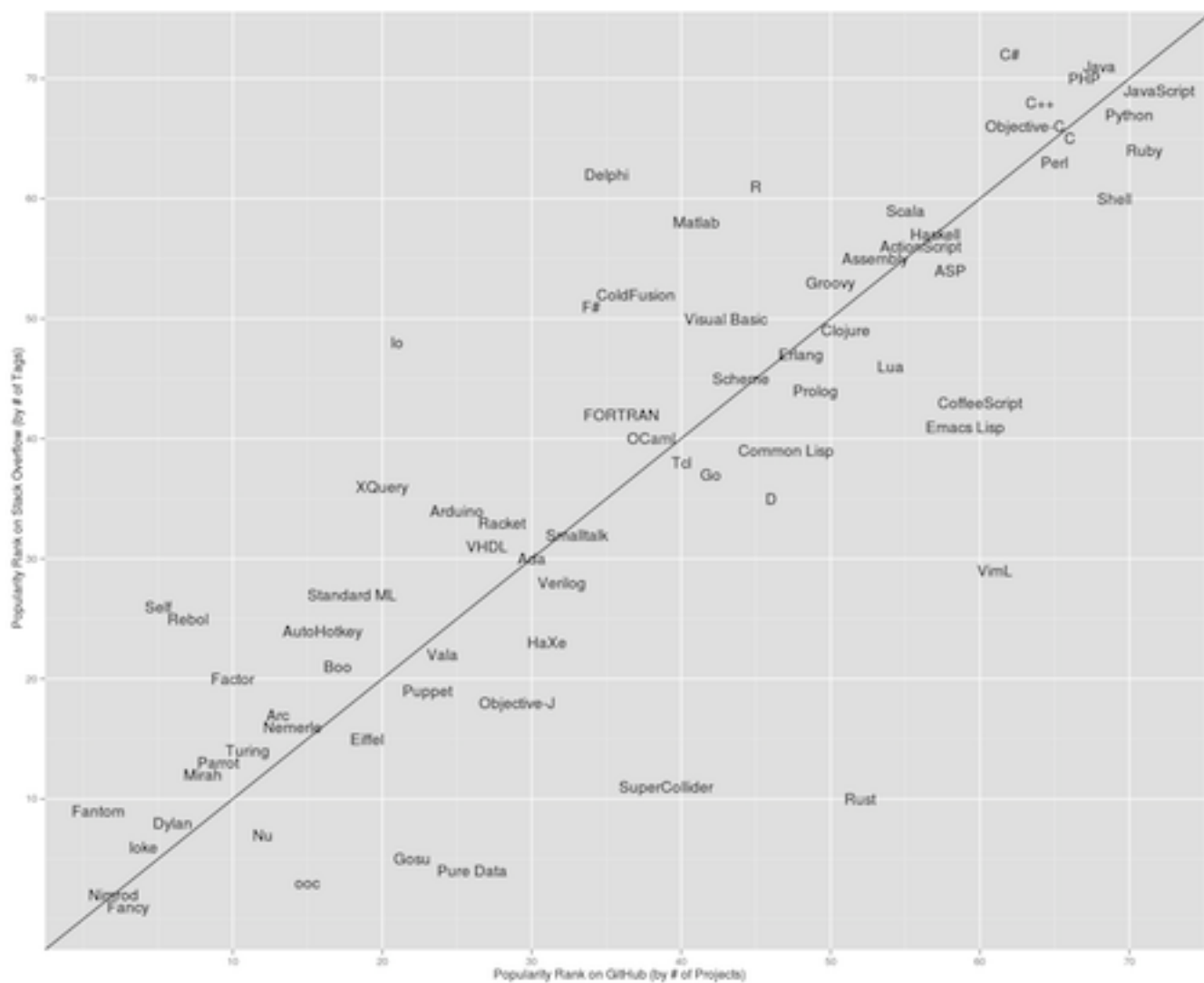


Рис. 1: The RedMonk Programming Language Rankings.

Т.к. график двумерный, то нельзя однозначно решить какой из языков лучше. Поэтому просто выберем языки входящие в правую верхнюю группу.

- C#
- Java
- JavaScript
- C
- C++
- Perl
- Python
- Ruby
- Objective-C
- PHP

2.4 Выбор популярных языков

Можно заметить, что не смотря на выбор рейтинга популярные языки программирования везде остаются примерно одни и те же лишь меняя своё место. Так что самыми популярными можно назвать:

- C
- Java
- C++
- C#
- Objective-C
- JavaScript
- Python
- Ruby
- Perl
- PHP

3 Ранняя история языков программирования.

Программирование как явление появилось за долго до появления ЭВМ.

Первым в истории программируемым устройством стал ткацкий станок Жаккарда. [3]

Француз Жозеф Мари Жаккард (1752-1834) изобрел первую программируемую машину с циклами в 1801 году. Эта машина не использовалась для вычислений. Это был ткацкий станок для узорчатых материй. Узор задавался при помощи набора перфокарт. Первым перфокарты для ткацких станков применил Jacques de Vaucanson (1709-1782) в 1745 году, но его машине нужно было скармливать карты по одной за раз. Станок Жаккарда в свою очередь мог выполнять циклы из заранее установленных перфокарт, создавая повторяющийся узор. В конечном итоге была создана колода из 24000 карт.

Машина Бэббиджа [4]

??

??

Программы загружались в неё при помощи жаккардовских первокарт.

Бэббидж предусмотрел подключение к своей машине нескольких считывателей перфокарт.

Причем некоторые из них могли контролировать работу других. Причем можно было как пропускать несколько карт, так и считывать карты в обратном порядке. Это позволяло пропускать или повторять участки программы, что уже соответствовало трем основным типам выражений: последовательный, итеративный и условный.

??

машина так и не была создана при жизни и вот только недавно... ???

<http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=1253887>

Ада Лавлейс, дочь лорда Байрона (того самого, знаменитого поэта), с детства увлекавшаяся математикой, общалась и находилась в переписке с самим Бэббиджем. По его просьбе она перевела и дополнила комментариями труд "Описание Аналитической машины" Луи Менабера, итальянского инженера. Работа Менабера представляет собой конспект лекции, которые

прочел Бэббидж в университете в Турине.

С текстом перевода можно ознакомиться на сайте <http://www.fourmilab.ch/babbage/sketch.html>

Среди прочего Ада включила в текст комментарий программу для вычисления чисел Бернулли, ввела определения цикла и смогла предвидеть множество областей применения вычислительной техники в том числе цифровую музыку.

$$B_n = \frac{-1}{n+1} \sum_{k=1}^n C_{n+1}^{k+1} B_{n-k}, \quad n \in \mathbb{N}$$

В период 1989 по 1991 год к двухсотлетию со дня рождения Чарльза Бэббиджа на основе его оригинальных работ в лондонском Музее науки была собрана работающая копия разностной машины. В 2000 году в том же музее заработал принтер, также придуманный Бэббиджем для своей машины. После устранения обнаруженных в старых чертежах небольших конструктивных неточностей, обе конструкции заработали безупречно.

<http://news.bbc.co.uk/2/hi/science/nature/710950.stm>

На этих машинах смогли воспроизвести программы предложенные графиней Лавлейс и они оказались рабочими.

Поэтому первым программистом в мире является женщина.

??

Вплоть до сороковых годов двадцатого века больше не делалось попыток создать столь сложные устройства. Появлялись механические вычислители-калькуляторы, но они не были тьюринг-полными.

Термин ???

В 1936 году английский математик Алан Тьюринг формализовал понятие алгоритма и вычислимости на основе некой абстрактной машины, названной в последствии машиной тьюринга.

Это было огромной вехой в истории развития информатики и теории программирования. А полностью по Тьюрингу или тьюринг-полнотой, называют свойство языка программирования, означающее его способность решать любые задачи. [5]

Alan Turing, who lived from 1912-1954, is credited for describing the essential characteristics of the first practical computer when in 1936 he published his paper "On Computable Numbers, with an application to the Entscheidungs problem".

Alan Turing

While not using the practical technology of the era, Alan Turing developed the idea of a "Universal Machine" capable of executing any describable algorithm, and forming the basis for the concept of "computability". Perhaps more importantly Turing's ideas differed from those of others who were solving arithmetic problems by introducing the concept of "symbol processing".

????????????????????

????????????????????

????????????????????

????????????????????

[6]

В 1942 году в Америке был создан ENIAC (Electronic Numerical Integrator And Computer — Электронный числовой интегратор и вычислитель). На смену механике пришли электронные сигналы. Правда процесс программирования такого компьютера заключался в его переконфигурировании. В течение нескольких дней из него получался специализированный компьютер для решения нужной задачи.

Хотя еще в годы Второй Мировой в Германии был разработан электромеханический Z4, программы которого хранились на перфокартах. Его автор Конрад Цузе придумал для него специальный язык программирования высокого уровня Планкалькюль. Но опубликованы работы с описаниями этого языка были довольно поздно, лишь в 1972 году, и уже не влияли

на развитие языков программирования.

Американцы же учли свои ошибки. Уже работая над созданием ENIAC они продумывали основы новой архитектуры, которая была впервые реализована в компьютере EDVAC. [7] Сейчас эта архитектура известна как архитектура фон Неймана. В её основе лежат принципы

- хранимой программы
????????????????????
- Принцип жесткости архитектуры
????????????????????
- ?????????????????????

Формулировка этих принципов оказала огромное влияние на развитие вычислительной техники и языков программирования.

Первым высокоуровневым языком программирования для электронных вычислительных машин стал шорткод (Short Code - короткий код), созданный Джоном В. Мочли в июле 1949 года. Интерпритатор был реализован В. Ф. Шмитом и запущен на UNIVAC I Serial 1 в 1950. Программа на этом языке уже не была набором машинных инструкций, а математических выражений. Программисту приходилось в ручную переписывать арифметические выражения в набор специальных символов, а затем в двоичный код, воспринимаемый интерпритатором. [?]

4 История C.

от 50x до C.

Чем он завоевал популярность

Где C сейчас применяется, чем он хорош.

5 C++

от C до C++

6 Objective-C

не знаю на сколько он отличается от C++. Но с ним все просто: история, затем где он применяется и кратко о том почему он становится популярен в последнее время.

7 Java C#

разные темы, но они оба в байт коде и материал по обоим есть.

8 JavaScript

9 Perl

Долгое время основным средством взаимодействия человека с компьютером оставалась командная строка. Это сейчас мы уже привыкли к оконным интерфейсам, а раньше все команды системе пользователю приходилось вбивать в ручную. Часть операционной системы,

которая отвечает за обработку команд пользователя называется интерпретатором командной строки или командными оболочками.

Ранние интерпретаторы такие как оболочка Томпсона (Thompson Shell) [?], появившаяся в 1971 году, первая командная оболочка операционной системы UNIX, или командный интерпретатор системы MULTICS www.multicians.org/history.html представленной в 1965, могли только выполнять команды пользователя и не имели средств автоматизации.

Командная оболочка Берна (Bourne shell) стала оболочкой по-умолчанию в седьмой версии операционной системы UNIX, релиз которой состоялся в 1979 году. <http://www.in-ulm.de/~mascheck/b>

Эта оболочка могла выполнять скрипты, которые были не просто последовательностью команд, а настоящими программами. Её синтаксис был крайне сложен

“Nobody really knows what the Bourne shell’s grammar is. Even examination of the source code is little help.” [?] ²

Но за то этот синтаксис позволял вставлять в скрипты команды переходов, циклы, управлять потоками выполнения, переменные, файловый ввод/вывод. По сути это язык оболочки берна стал языком программирования.

Но крайне сложный синтаксис отпугивал многих.

Примерно в то же время появляется интерпретируемый язык AWK, названный так по первым буквам фамилий авторов Alfred Aho, Peter Weinberger, and Brian Kernighan. ³ Этот язык предназначался для построчного разбора и обработки входного потока (например, текстового файла) по заданным шаблонам. Он стал де-факто стандартом и до сих пор его новые версии NAWK и GAWK используются почти во всех UNIX-подобных системах.

Лингвист по образованию, Ларри Уолл (Larry Wall) в 1 февраля 1988 года опубликовал в новостной конференции comp.sources.misc новость под заголовком “Perl, a “replacement” for awk and sed“ ⁴

содержащее довольно смелое высказывание о том, что теперь sed и awk можно считать устаревшими.

Perl is a interpreted language optimized for scanning arbitrary text files, extracting information from those text files, and printing reports based on that information. It’s also a good language for many system management tasks. The language is intended to be practical (easy to use, efficient, complete) rather than beautiful (tiny, elegant, minimal). It combines (in the author’s opinion, anyway) some of the best features of C, sed, awk, and sh, so people familiar with those languages should have little difficulty with it. (Language historians will also note some vestiges of csh, Pascal, and even BASIC-PLUS.) Expression syntax corresponds quite closely to C expression syntax. If you have a problem that would ordinarily use sed or awk or sh, but it exceeds their capabilities or must run a little faster, and you don’t want to write the silly thing in C, then perl may be for you. There are also translators to turn your sed and awk scripts into perl scripts.

Perl был разработан Ларри Уоллом (Larry Wall) в 1986 году. Ларри Уолл являлся в то время системным администратором одного проекта UNIX, связанного с созданием многоуровневой безопасной сети, которая объединяла несколько компьютеров, разнесенных на большие расстояния. Работа была выполнена, но потребовалось создание отчетов на основе большого числа файлов с многочисленными перекрестными ссылками между ними. Первоначально Ларри предполагал использовать для этих целей фильтр awk, но оказалось, что последний не мог управлять открытием и закрытием большого числа файлов на основе содержащейся в них же самой информации об их расположении. Его первой мыслью было написать специальную системную утилиту, решающую поставленную задачу, но, вспомнив, что до этого

²Tom Duff, “Rc-A Shell for Plan 9 and UNIX systems”, Proc. of the Summer 1990 EUUG Conf., London, July, 1990, pp. 21-33

³http://www.computerworld.com.au/article/216844/a-z_programming_languages_awk/?fp=4194304&fpid=1

⁴http://groups.google.com/group/comp.sources.unix/tree/browse_frm/month/1988-02?_done=/group/comp.sources.unix/browse_frm/month/1988-02?&

ему уже пришлось написать несколько утилит для решения задач, не «берущихся» стандартными средствами UNIX, он принял кардинальное решение — разработать язык программирования, который сочетал бы в себе возможности обработки текстовых файлов (sed), генерации отчетов (awk), решения системных задач (shell) и низкоуровневое программирование, доступное на языке C. Результатом этого решения и явился язык Perl, интерпретатор для которого был написан на C. По утверждению самого Ларри Уолла, при создании языка Perl им двигала лень — в том смысле, что для решения стоявшей перед ним задачи следовало бы написать большое количество программ на разных языках, входящих в состав инструментальных средств UNIX, а это достаточно утомительное занятие. Новый язык программирования сочетал в себе возможности системного администрирования и обработки файлов — две основные задачи, решаемые обычно при программировании в системе UNIX. Причем следует отметить, что язык Perl появился из практических соображений, а не из-за желания создать еще одно «красивое» средство для работы в UNIX, поэтому-то он и получил широкое распространение среди системных администраторов, когда Ларри Уолл предоставил его широкому кругу пользователей. После создания языка Perl появилась возможность решать задачи с помощью одного инструмента и не тратить время на изучение нескольких языков среды программирования UNIX. Первая версия языка не содержала многих возможностей, вошедших в последнюю версию Perl, с которой читатель познакомится в нашей книге и которая идентифицируется как версия 5.6.1.626 и считается устойчивой. Первоначально язык включал

простой поиск по строковым образцам (шаблонам) в файлах;
дескрипторы файлов;
скалярные переменные;
форматы.

Вся документация умещалась на 15 страницах, но Perl решал задачи быстрее, чем sed или awk, и скоро стал использоваться не только для решения задач системного администрирования.

В дальнейшем сам Ларри Уолл позаимствовал у Генри Спенсера (Henry Spencer) («бога регулярных выражений», как о нем написано в одной из книг) пакет для работы с регулярными выражениями и модифицировал его для использования в языке Perl. Другие функциональные возможности были разработаны не только Ларри Уоллом, но и его друзьями и коллегами, и также включены в состав языка. Опубликование в Интернете привело к появлению сообщества единомышленников, которые не только использовали для решения своих задач язык, но и развивали его. Perl и по настоящее время продолжает интенсивно развиваться за счет разработки пакетов, реализующих новые функциональные возможности языка, отражающие стремительное развитие современных информационных технологий. В табл. 1.1 представлена динамика появления новых версий языка Perl, начиная с самой первой.

Язык активно развивался, но до 1991 года, когда в свет вышла книга Программирование на Перле, не существовало приличной документации. Эта книга стала фактически стандартом языка. С её выходом текущую версию языка тут же называли 4.0, чтобы было проще ссылаться на него в книге.⁵

Язык начал активно развиваться и набирать популярность с выходом пятой версии. Появилась возможность расширять стандартные возможности при помощи модулей написанных другими программистами, это стало удобно за счет появления On October 26, 1995, the Comprehensive Perl Archive Network (CPAN). Сейчас на серверах CPAN хранится около 170000 модулей <http://www.cpan.org/>. Это, а так же то, что язык, его интерпретатор и все инструменты для работы с ним распространялись на основе лицензии GNU GPL (General Public License).

Существует множество крупных веб-порталов написанных на языке PERL. Например из-

⁵<http://perldoc.perl.org/perlhist.html>

вестный онлайн-магазин Amazon⁶

????????????????????

веб-фрэймворки на перле:

Catalyst

Dancer

Gantry

Mason

Maypole

Mojolicious

Poet

WebGUI

На каждом множество сайтов.

Perl как язык разработки скриптов имеет ряд преимуществ. Перечислим их в порядке значимости:

- независимость от программно-аппаратной платформы;
- мощные средства разбора строк;
- простота работы с переменными окружения ;
- простота работы со входными и выходными стандартными потоками;
- возможность чтения заданного числа символов из входного потока;
- хешированные таблицы;
- возможность организации конвейеров;
- библиотеки TCP/IP-обмена;
- множество стандартных библиотек прикладных программ.

10 PHP

<http://php.net/manual/ru/history.php.php> PHP, как всем известно, на самом деле преемник продукта под названием RHP/FI. Созданное в 1994 году Расмусом Лердорфом, самое первое воплощение RHP было простым набором CGI-скриптов, написанных на языке программирования Си. Изначально используя их для отслеживания посещений своего веб-резюме, он назвал этот набор скриптов "Personal Homepages Tools" ("Инструменты для персональных домашних страниц"), но более часто упоминалось название "RHP Tools". Со временем требовалось все больше улучшений функциональности, и Расмус переписал RHP Tools, создав более крупную и богатую реализацию. Эта новая реализация была способна взаимодействовать с базами данных и многое другое, что создавало фреймворк, с помощью которого пользователи могли создавать простые динамические веб-приложения, такие как гостевые книги. В июне 1995 года Расмус » открыл исходный код RHP Tools общественности, что позволило разработчикам использовать его по своему усмотрению. Это также дало возможность пользователям исправлять ошибки в коде и улучшать его.

В сентябре того же года, Расмус расширил RHP и на короткое время убрал из названия RHP.

Теперь в виде инструмента FI (сокращение от "Интерпретатор Форм"), новая реализация включала в себя некоторые основные функциональные возможности того RHP, который мы знаем сегодня. Она имела Perl-подобные переменные, автоматическую интерпретацию форм

⁶<http://www.amazon.com/exec/obidos/tg/feature/-/7137/>

и встраиваемый в HTML синтаксис. Синтаксис языка был похож на Perl, хотя и был гораздо более ограниченным, простым, и в некоторой степени противоречивым. Для того, чтобы вставлять код в HTML-файл, разработчикам пришлось использовать HTML комментарии. Хотя этот метод был не совсем хорошо принят, FI по-прежнему набирал популярность в качестве CGI-инструмента, но все-таки не в качестве языка. Однако, перемены начались в следующем месяце, когда в октябре 1995 года Расмус выпустил полностью переписанный код. С вернувшимся именем PHP, но уже сокращенным от "Personal Home Page Construction Kit" это был первый релиз, который мог похвастаться расширенным интерфейсом сценариев. Язык намеренно напоминал Си по структуре, что делало его легким для восприятия разработчиками, знакомыми с Си, Perl и подобными языками. Будучи все еще ограниченными UNIX и POSIX-совместимыми системами, был изучен вопрос для реализации языка в Windows NT.

Код получил еще одно существенное преобразование в апреле 1996 года. Объединив названия предыдущих версий, Расмус представил PHP/FI. Реализации второго поколения начали по-настоящему развивать PHP из набора инструментов в самостоятельный язык программирования. PHP включал в себя встроенную поддержку для DBM, mSQL и Postgres95 баз данных, cookies, поддержку определяемых пользователем функций и многое другое. В июне PHP/FI была присвоена версия 2.0. Интересно, однако, что существовала только одна версия PHP 2.0. Когда она, наконец, в ноябре 1997 года преодолела статус бета-версии, парсер языка был уже полностью переписан.

Несмотря на короткую историю разработки, PHP/FI продолжал набирать популярность в молодом мире веб-разработки. В 1997 и 1998, PHP/FI стал культом для нескольких тысяч пользователей по всему миру. Исследования Netcraft в мае 1998 года показали, что почти 60 тысяч доменов передавали заголовки, содержащие "PHP". Это число равнялось примерно 1/3. Несмотря на эти впечатляющие цифры, развитие PHP/FI было ограничено: несмотря на нескольких второстепенных участников, в целом он по-прежнему разрабатывался одним человеком.

PHP 3

PHP 3.0 был первой версией, напоминающей PHP, каким мы знаем его сегодня. Посчитав PHP/FI 2.0 все еще неэффективным и недостаточно функциональным для использования в коммерческих приложениях, разрабатываемых для их университетского проекта, Энди Гутманс и Зив Сураски из Тель-Авива начали еще раз заново переписывать парсер в 1997 году. Связавшись с Расмусом, они обсудили различные аспекты текущей реализации и их новой разработки PHP. Для улучшения движка и использования уже существующей базы пользователей PHP/FI, Энди, Расмус и Зив решили сотрудничать в развитии нового, независимого языка программирования. Этот совершенно новый язык был выпущен под новым именем, без упоминания о персональном использовании, как в PHP/FI 2.0. Он был назван просто "PHP аббревиатура, означающая рекурсивный акроним - PHP: Hypertext Preprocessor.

Одной из сильнейших сторон PHP 3.0 была возможность расширения ядра. Кроме обеспечения пользователей надежной инфраструктурой из множества различных баз данных, протоколов и API, расширяемость PHP 3.0 привлекла к нему множество сторонних разработчиков, желающих добавить к языку свои расширения. Возможно, это и был главный ключ к успеху, но стоит добавить, что немаловажным шагом оказалась поддержка ООП синтаксиса и намного более мощного и последовательного синтаксиса самого языка.

В июне 1998 года, со множеством новых разработчиков со всего мира присоединившихся к проекту, PHP 3.0 был представлен новой командой разработчиков, как официальный преемник PHP/FI. Активное развитие PHP/FI 2.0, фактически прекратившееся в ноябре прошлого года, теперь официально закончилось. Примерно через девять месяцев открытого публичного тестирования, при объявлении официального выпуска PHP 3.0, он уже был установлен на более чем 70000 доменах по всему миру, и уже не ограничивается POSIX-совместимыми операционными системами. Относительно небольшая доля доменов с установленным PHP была размещена на серверах под управлением Windows 95, 98 и NT, а также Macintosh. На пике своего развития, PHP 3.0 был установлен приблизительно на 10 PHP 4

К зиме 1998 года, практически сразу после официального выхода PHP 3.0, Энди Гутманс и Зив Сураски начали переработку ядра PHP. В задачи входило увеличение производительности сложных приложений и улучшение модульности кодовой базы PHP. PHP 3.0 дал возможность подобным приложениям успешно работать с набором баз данных и поддерживать большое количество различных API и протоколов, но PHP 3.0 не имел качественной поддержки модулей и приложения работали неэффективно.

Новый движок, названный 'Zend Engine' (от имен создателей: Zeev и Andi), успешно справлялся с поставленными задачами и впервые был представлен в середине 1999 года. PHP 4.0, основанный на этом движке и принесший с собой набор дополнительных функций, официально вышел в мае 2000 года, почти через два года после выхода своего предшественника. Кроме значительного улучшения производительности, PHP 4.0 имел еще несколько ключевых нововведений, таких как поддержка намного большего числа веб-серверов, поддержка HTTP сессий, буферизация вывода, более безопасные способы обработки вводимой пользователем информации и несколько новых языковых конструкций. PHP 5

После долгой разработки и нескольких пре-релизов в июле 2004 был выпущен PHP 5. В основном он управляется ядром Zend Engine 2.0 с новой объектной моделью и множеством различных других нововведений.

Команда разработчиков PHP включает в себя десятки разработчиков, а также десятки других организаций, работающих над связанными с PHP и его поддержкой проектами, такими как PEAR, PECL и документацией, а также базовую инфраструктуру сети более чем из ста серверов на шести из семи континентах мира. Основываясь на статистике прошлых лет, можно с уверенностью предположить, что PHP теперь установлен на десятки или даже, возможно, сотни миллионов доменов по всему миру.

<http://docs.php.net/manual/ru/features.php>

В области программирования для сети Интернет PHP — один из популярных скриптовых языков (наряду с JSP, Perl и языками, используемыми в ASP.NET) благодаря своей простоте, скорости выполнения, богатой функциональности, кроссплатформенности и распространению исходных кодов на основе лицензии PHP.

Популярность в области построения веб-сайтов определяется наличием большого набора встроенных средств для разработки веб-приложений[7]. Основные из них:

- автоматическое извлечение POST и GET-параметров, а также переменных окружения веб-сервера в предопределённые массивы;
- взаимодействие с большим количеством различных систем управления базами данных (MySQL, MySQLi, SQLite, PostgreSQL, Oracle (OCI8), Oracle, Microsoft SQL Server, Sybase, ODBC, mSQL, IBM DB2, Cloudscape и Apache Derby, Informix, Ovrinos SQL, Lotus Notes, DB++, DBM, dBase, DBX, FrontBase, FilePro, Ingres II, SESAM, Firebird / InterBase, Paradox File Access, MaxDB, Интефейс PDO);
- автоматизированная отправка HTTP-заголовков;
- работа с HTTP-авторизацией;
- работа с cookies и сессиями;
- работа с локальными и удалёнными файлами, сокетами;
- обработка файлов, загружаемых на сервер;
- работа с XForms.

В настоящее время PHP используется сотнями тысяч разработчиков. К крупнейшим сайтам, использующим PHP, относятся Facebook, Wikipedia и др.

<http://laziness-impatience-hubris.blogspot.ru/2009/07/php-perl-web-frameworks.html>

Причины популярности PHP на фоне Perl Web Frameworks

??

11 Python

abc

<http://www.idhub.com/abc/>

Leo Geurts, Lambert Meertens, Steven Pemberton

Everybody knows about designing web sites and computer systems so that they are easy to use, but what would happen if you designed a programming language with the same goals?

The answer is ABC; a language that was designed on a ‘clean sheet’ with the goal of delivering an easy-to-use language ideal for teaching and learning about programming.

ABC is more readable than other languages; no obscure commands. It is more concise than other languages; equivalent programs written in C are five times as long.

It is more powerful than other languages; its perfect choice of high-level data types make it an idea tool for serious programming. ABC was one of the inspirations for the Python programming language and Python shares key ideas and features of ABC.

This book provides a hands-on introduction to the language including many programming examples along with a users guide and a complete definition of the language. The language is available free to run on PCs, Macs and UNIX.

python

http://www.computerworld.com.au/article/255835/a-z_programming_languages_python/ Язык созданный в 1989 году голландцем Гвидо ван Россумом (Guido van Rossum) и названный в честь английского юмористического шоу "Летающий цирк Монти Пайтона". Является непосредственным потомком языка ABC. Гвидо работал в команде занимавшейся разработкой ABC и хотел создать что-то подобное, исправив ошибки и проблемы этого языка. Главная проблема, которую хотел решить Гвидо — продуктивность. Тогда в 1989 году он работал с группой инженеров над разработкой новой ОС и часто обращал внимание на то, как использование C усложняет дело. В начале 1991 года он уже сделал его общедоступным.

<http://www.python.org/dev/peps/pep-0020/> Философия Питона

- Красивое лучше, чем уродливое.
- Явное лучше, чем неявное.
- Простое лучше, чем сложное.
- Сложное лучше, чем запутанное.
- Плоское лучше, чем вложенное.
- Разреженное лучше, чем плотное.
- Читаемость имеет значение.
- Особые случаи не настолько особые, чтобы нарушать правила.
- При этом практичность важнее безупречности.
- Ошибки никогда не должны замалчиваться.
- Если не замалчиваются явно.
- Встретив двусмысленность, отбрось искушение угадать.
- Должен существовать один — и, желательно, только один — очевидный способ сделать это.
- Хотя он поначалу может быть и не очевиден, если вы не голландец⁷.

⁷Намек на национальность Гвидо ван Россума

- Сейчас лучше, чем никогда.
- Хотя никогда зачастую лучше, чем прямо сейчас.
- Если реализацию сложно объяснить — идея плоха.
- Если реализацию легко объяснить — идея, возможно, хороша.
- Пространства имён — отличная штука! Будем делать их побольше!

Нет стандартов языка. Стандартом считается реализация интерпритатора Puthon на C Cpython. Но есть и другие реализации Python: ironpython jython pyru

В качестве документации может рассматриваться PEP (Python Enhancement Proposal - предложения по расширению питона).⁸ Это постоянно развивающаяся документация, в соответствии которой происходит развитие языка.

Популярные Веб-фрэймворки Django, Zope

Язык ABC не поддавался расширению. Чтобы избежать этого Гвидо сделал язык yton максимально расширяемым, с очень большой сестемной библиотекой. Много библиотек.

Язык используется везде. Вообще везде. В научных расчетах, в прототипировании, в разработке встроенного ПО, как серверный язык для веб-приложений, экспертные системы.

Активно используется и поддерживается Гуглом.

12 Ruby

Язык Руби был создан Юхиро Мацумото (Yukihiro Matsumoto), в декабре 1994 года вышла первая альфа-версия языка.

Мацумото говорил: "Мне нужен был скриптовый язык более мощный чем Перл и более объектно-ориентированный чем Питон. Поэтому я решил сделать своя язык."⁹

В 1996 году сформировалось сообщество, до этого момента Мацумото, выложивший все свои наработки в открытый доступ как большой фанат открытого ПО, работал один. Ruby is named after the precious gemstone, it's not an abbreviation of anything. When I started the language project, I was joking with a friend that the project must be code-named after a gemstone's name (ala Perl). So my friend came up with "ruby". It's a short name for a beautiful and highly valued stone. So I picked up that name, and it eventually became the official name of the language.

Later, I found out that pearl is the birthstone for June, and ruby is the birthstone for July. I believe Ruby is an appropriate name for the next language after Perl.

У этого языка тоже есть своя философия заложенная своим основателем. Звучит она японски кратко "принцип наименьшего удивления". Мацумото хотел, чтобы программист чувствовал себя комфортно, чтобы язык делал именно то, что ожидает от него программист. Язык стал весьма популярен в Японии, но за её пределами стал популяризовываться только 1997 году, когда документация была переведена на английский.

Из сравнительных графиков охloh¹⁰ видно как за последние годы perl теряет популярность, я ruby только набирает её.

Менеджер библиотек RubyGems аналогичный во многом перловскому CPAN.

Медленный, но верный рост популярности руби обязан в основном приложению Ruby-on-Rails¹¹

13 Заключение.

⁸<http://www.python.org/dev/peps/pep-0001/>

⁹<http://linuxdevcenter.com/pub/a/linux/2001/11/29/ruby.html>

¹⁰<http://www.ohloh.net/languages/compare?measure=commits&percent=true&l0=ruby&l1=perl&l2=-1&l3=php&l4=-1&commit=Update>

¹¹<http://www.devarticles.com/c/a/Ruby-on-Rails/Web-Development-Ruby-on-Rails/>

Содержание

1	Аннотация	1
2	Рейтинги популярности языков программирования.	1
2.1	TIOBE Programming Community Index	1
2.2	ohloh.net Language Comparison Page	2
2.3	The RedMonk Programming Language Rankings	3
2.4	Выбор популярных языков	5
3	Ранняя история языков программирования.	5
4	История C.	7
5	C++	7
6	Objective-C	7
7	Java C#	7
8	JavaScript	7
9	Perl	7
10	PHP	10
11	Python	13
12	Ruby	14
13	Заключение.	14

πSπSπSπSπSπS πSπSπSπSπSπSπSπSπSπSπSπSπS- πS πSπSπSπSπSπSπSπSπS

1. Stephen O'Grady. The redmonk programming language rankings: February 2012. Website, 2012. <http://redmonk.com/sogrady/2012/02/08/language-rankings-2-2012/>.
2. Drew Conway. Ranking the popularity of programming languages. Website, 2010. <http://www.dataists.com/2010/12/ranking-the-popularity-of-programming-languages/>.
3. E.G. Swedin and D.L. Ferro. *Computers: The Life Story Of A Technology*. Greenwood Technographies. Greenwood Press, 2005.
4. John Fuegi and Jo Francis. Lovelace & babbage and the creation of the 1843 'notes'. *IEEE Ann. Hist. Comput.*, 25(4):16–26, October 2003.
5. Alan M. Turing. On computable numbers, with an application to the entscheidungsproblem. *Proceedings of the London Mathematical Society*, 42:230–265, 1936.
6. Юрий Полунов. Автора !!! Website, 2006. http://www.pcweek.ru/themes/detail.php?ID=72606&phrase_id=204737.
7. John von Neumann. First draft of a report on the edvac. *IEEE Ann. Hist. Comput.*, 15(4):27–75, October 1993.