

ПИТОН

Курс лекций

Лекция вторая

© Зайцев Вадим Валерьевич, 2002–2010,
spider.vz@gmail.com

Раз уж зашла речь о языках, это достойно того, чтобы поговорить подробнее:

1.3 Типы языков программирования и их эволюция

По этой теме написано книг чуть ли не больше, чем по каждому языку отдельно. Но мы попытаемся ограничиться лишь общим обзором.

1. Ассемблеры — это вербализованные машинные коды. Сколько машинных архитектур, столько и ассемблеров. Даже самая малая программа занимает много страниц на этом языке, абстракции никакой, уровень сверхнизкий. Сейчас эти языки используются только в мелких, но очень важных частях систем, которым необходимо быстроедействие.
2. Процедурные языки — языки среднего и высокого уровня, ориентированные на деление основной проблемы на несколько более мелких и решение каждой мелкой с помощью своей подпрограммы. Основные представители этого направления: **Фортран** (в настоящее время используется версия Фортран-99, и та только в программировании больших численных проектов, откуда постепенно вытесняется готовыми математическими вычислительными системами вроде Мэпл, Матлаб и других), **Кобол** (применяется в области экономики), **Алгол** (не применяется нигде, но в 60х годах имел большое теоретическое влияние на развитие теории языков программирования), **Си** (уже почти не используется), **Ада** (широко использовался Департаментом Защиты США, сейчас заменён) и **Паскаль** (пока что используется в системе Дельфи, но постепенно умирает).

Большинство используемых процедурных языков имеют ограниченные возможности работы с объектами, но не дотягивают до языков следующей категории.

3. Объектно-ориентированные языки — языки высокого уровня, ориентированные только на работу с различными объектами. Наиболее используемая в наше время группа. Основные представители: Си++ (очень широко используется во многих областях), Ада-95 (опять-таки, используется в основном Департаментом Защиты США), Ява (потомок Си++, используется всё шире с каждым днём, удобен для интернет-программирования), Смоллток (один из первых объектно-ориентированных языков программирования, живой и по сей день), КЛОС (о котором ниже) и Эйфель (программирование, ориентированное на ограничения — инварианты).
4. Языки, ориентированные на данные — языки, созданные специально для работы с одним определённым типом данных. Например, АПЛ настроен на работу с матрицами и векторами без циклов, Снобол и его преемник Икон работают со строками как с базовой структурой, СЕТЛ позволяет описывать множества почти математическим языком, Форт полностью ориентирован на стек.
5. Функциональные языки — практически разросшийся подтип языков, ориентированных на данные. Основная структура данных — связный список. Функциональными языками они названы за счёт того, что программирование на них принципиально отличается от процедурного. Функциональные языки — это ЛИСП и его потомки: более объектно-ориентированный — КЛОС и более чисто реализующий функциональную парадигму — ML.
6. Логические языки — языки, ориентированные на решение проблем без описания алгоритмов. Действительно используется только один язык — Пролог. Где? Конечно, в области искусственного интеллекта.
7. Сценарные языки, ещё называемые скриптами — это языки, для которых не существует отдельной от какого-либо программного продукта реализации, либо используемые только в связке с одной программой или типом программ. Это, конечно, прежде всего Яваскрипт, простейшее и одновременно наиболее широко используемое средство интернет-программирования. Этот язык позволяет управлять браузером — программой просмотра интернет-документов — причём его возможностей хватает подчас для реализации больших серьёзных проектов. Ещё два типично сценарных языка (выросших, однако, и приобретших отдельные реализации) — это Перл и Питон — две большие противоположности. Перл — очень сложный и мощный сиподобный язык, питон — попроще и полегче, к тому же более архаично построенный, паскале- или даже фортраноподобный. Хотя простота, конечно, не всегда означает меньшие возможности.

На этом можно наш обзор завершить. Конечно, языков программирования существуют многие тысячи, к тому же есть ещё широко используемые языки разметки, такие как HTML, XML или Т_РХ.

Мы недаром перескочили через определение языка программирования. «Некорректный вопрос», как написано в одной уважаемой книге. Вообще, формальное определение существует.

Определение. Программы суть последовательности символов, определяющие вычисление.

Определение. Языки программирования суть наборы правил, определяющих, какие последовательности символов составляют программу и какое именно вычисление описывается этой программой.

Как видно, можно дать определение, даже не используя слово *компьютер*. На деле же язык программирования используется как механизм абстрагирования, позволяя программисту описать вычисления абстрактно и перекладывая большую часть работы на транслятор.

2 Введение в язык питон

2.1 Краткая история языка

Питон — молодой сценарный язык, история которого началась только в 1990 году, когда сотрудник голландского института CWI, тогда ещё мало кому известный Гвидо ван Россум участвовал в проекте создания языка ABC. Этот язык был предназначен для замены языка Бейсик в обучении студентов основным концепциям программирования. (Язык Бейсик как-то странно и надолго закрепился в сфере обучения, хотя многие понимали, что к добру это привести не может. Например, одно из светил теории программирования Эдсгер Дейкстра говорил, что «преподавателей, которые начинают обучение программированию с бейсика, следует привлекать к уголовной ответственности»).

Параллельно с работой над основным проектом Гвидо ван Россум дома на своём Макинтоше написал интерпретатор другого простого языка; он, конечно, позаимствовал некоторое количество идей из ABC. Он назвал его «Питон» и стал распространять через Интернет.

Язык стал быстро развиваться, поскольку появилось большое количество заинтересованных и понимающих в развитии языков программирования людей. Сначала это был совсем простой язык, просто небольшой интерпретатор, некоторое количество функций, не было объектно-ориентированного программирования, но всё это быстро появилось. Уже в 1991 году появились первые средства объектно-ориентированного программирования.

Позже Гвидо ван Россум переехал из Голландии в Америку, перешёл из CWI в CNRI, потом в фирму BeOpen Labs, а сейчас работает в Digital Creations. Всё это время он продолжает развитие языка, выпуская новые версии. Причём каждая следующая версия имеет несколько серьёзных отличий от предыдущей, меняющих подчас саму философию программирования и подходы к решению различных задач.

Интерпретаторы питона существуют под все мыслимые платформы:

Windows, UNIX, Macintosh, QNX и пр. Все они распространяются бесплатно, что обеспечивает дополнительную привлекательность использования этого языка как в коммерческих, так и в свободно-распространяемых проектах.

Последняя версия питона — 2.1 — уже пятнадцатая, откомпилированная под **Windows** 16 апреля 2001 в 18:25:49. Спектр разработанного программного обеспечения (как в форме отдельных программ, так и в форме подключаемых модулей) очень разнообразен:

- **Zope** — сервер интернет-приложений, позволяющий создавать и поддерживать интернет-сайты со сложной структурой не только профессионалам, но и простым редакторам и наборщикам.
- **Jython** — реализация питона, позволяющая компилировать программы на нём в коды виртуальной ява-машины (универсального воображаемого компьютера, в команды которого компилируются программы на языке ява). Установка явы на персональный компьютер означает установку программы, позволяющей выполнять команды виртуальной ява-машины, поэтому откомпилированные программы на яве остаются машинно-независимыми (если говорить о реальных машинах, конечно). Сейчас возможность запускать мелкие программы на яве (так называемые *апплеты*) встроена почти в каждый браузер, и Jython — это начало наступления питона на яву.
- **Blender** — пакет работы с трёхмерной графикой и создания сложных фильмов, использующий питон по прямому назначению — в качестве сценарного языка. Питон позволяет легко в паре десятков строк кода сформулировать сложное движение трёхмерной фигуры.
- **Mailman** — программа поддержки списков рассылки. Имеет поддержку всех необходимых возможностей: работа со шлюзом групп новостей, формирование дайджестов, ведение архивов и т.п.
- Два математических расширения питона: **Numeric** и **Scientific**. Первое помогает работать с матрицами различными численными методами, по возможностям сравнимо с системой Матлаб. Второе представляет из себя набор модулей, реализующих тензорное исчисление, статические процедуры, трёхмерную визуализацию и пр.
- **PyXML** и **4Suite** позволяют работать на питоне с такими современными технологиями, как XML, XPath, XSLT, SAX, DOM, RDF и ODS.
- **Sketch** и **PIL** — ещё два пакета работы с графикой. Первый — это просто векторный графический редактор, написанный на питоне, а второй — пакет для работы с различными растровыми форматами.

Кроме того, питон сильно теснит остальные языки: он широко используется как сценарный язык CGI, отвоевывая место у перла; в стандартной поставке питона есть платформонезависимый модуль Tk для лёгкого

построения графического интерфейса (раньше он использовался только в связке с языком тикль (TCL, Tool Command Language, язык командования инструментами)).

2.2 Работа с интерпретатором питона

Питон — интерпретируемый язык. Это значит, что термин *программа* эквивалентен термину *исходный текст программы*. Питон может работать в двух различных режимах: интерактивном и неинтерактивном.

В интерактивном режиме питон ведёт диалог с пользователем. Реплики самого питона не блещут разнообразием — они включают >>>, ... и результаты введённых выражений. Первые две реплики — это приглашения, если последнее, что написано на экране — это >>>, можно смело начинать набирать новую команду. ... означает, что набор команды ещё не кончен несмотря на переход на новую строку (возможно, не закрыта скобка или действительно выражение ещё не закончено).

Можно начинать вводить выражения питона или последовать выводимому на экран при запуске совету посмотреть права, благодарности или лицензию:

```
Python 2.1 (#15, Apr 16 2001, 18:25:49) [MSC 32 bit (Intel)] on win32
Type "copyright", "credits" or "license" for more information.
>>> copyright
Copyright (c) 2001 Python Software Foundation.
All Rights Reserved.
Copyright (c) 2000 BeOpen.com.
All Rights Reserved.
Copyright (c) 1995-2001 Corporation for National Research Initiatives.
All Rights Reserved.
Copyright (c) 1991-1995 Stichting Mathematisch Centrum, Amsterdam.
All Rights Reserved.
>>> credits
Thanks to CWI, CNRI, BeOpen.com, Digital Creations and a cast of thousands
for supporting Python development. See www.python.org for more information.
>>>
```

Лицензию смотреть не рекомендуем — ценной информации там нет, а занимает она несколько страниц. А вот советом заглянуть на <http://www.python.org> пренебрегать не стоит, по этому адресу можно найти много полезной информации.

Неинтерактивный режим подразумевает существование программы, записанной в отдельный файл. Питон переходит в этот режим автоматически, если при запуске дать ему первым же параметром имя файла с программой (после выполнения программы управление вернётся операционной системе, а не интерпретатору!).