

Микаэль Франц

Java: критическая оценка

Источник: Мир ПК, #08/1997

О Java — "языке программирования для Internet" — сейчас говорят повсеместно. Увы, это делается в несколько шарлатанской манере; о том, что многие возможности Java отнюдь не новы и уже давно прекрасно зарекомендовали себя (в особенности в языках близких к Паскалю), чаще всего умалчивают. Ниже мы коснемся стремительного взлета Java, выявим неизвестные сейчас широкой публике связи между Java и Паскалем и, наконец, рискнем сформулировать прогноз относительно будущего Java.

Введение

Как точно подметил в одной из своих работ Никлаус Вирт [Вирт 89], мир информатики подвержен влияниям моды. О технологии Java без всякого преувеличения можно сказать, что она сейчас в моде. Рано или поздно эта мода пройдет, и тогда интерес к Java снизится до несколько более разумного уровня. Цель настоящей работы — рассмотреть феномен Java в контексте современных, в особенности европейских, исследований в области информатики, попутно развеяв миф о спонтанном возникновении новой технологии на пустом месте, а затем назвать реальные достоинства Java.

"Концепция Java", состоящая из объектно-ориентированного языка программирования и архитектуры сконструированной для него виртуальной машины — была представлена общественности в августе 1995 г. в ходе беспримерной маркетинговой акции. Благодаря ряду факторов, на которых мы чуть ниже остановимся подробнее, концепция быстро приобрела огромную популярность. Порой и самому не верится, что с момента появления Java прошло меньше двух лет, — настолько всеобъемлющим образом удалось новой концепции заполнить мир информатики.

Стремительное распространение Java

Концепция Java, т. е. комбинация языка и виртуальной машины, ведет свое происхождение от проекта фирмы Sun под названием Green (Oak), открытого в 1990 г. Первоначальной целью проекта было создать среду разработки программного обеспечения для бытовой электроники. Современными электроприборами — тостерами, стиральными машинами, мобильными телефонами — управляют все более и более изощренные программы, выполняемые встроенными компьютерами — так называемыми микроконтроллерами, и Sun поставила задачу заменить множество архитектур микроконтроллеров одной-единственной масштабируемой архитектурой (прототипом нынешней виртуальной Java-машины) с широким спектром применений. В качестве производителя микросхем Sun, естественно, планировала в дальнейшем наладить выпуск и поставку соответствующих микроконтроллеров.

Согласно информации Sun [JavaSoft], идея переориентации Java на World Wide Web возникла в 1994 г. и принадлежала Биллу Джою, одному из основателей фирмы. Осенью 1994 г. началась работа по реализации Web-браузера HotJava с возможностью выполнения апплетов. В марте 1995 г. браузер был представлен избранному кругу стратегических партнеров Sun, включая руководство корпорации Netscape Communications, чья публичная поддержка впоследствии немало способствовала успеху новой технологии. В августе 1995 г. состоялся триумфальный выпуск Java, который происходил в довольно необычной для базовых технологий форме — в виде серии пресс-конференций.

Своим поразительно быстрым распространением идея Java обязана необычайно удачному сочетанию нескольких факторов.

- Взрывоподобный рост Internet привел к усиленному распространению идей, связанных с глобальными коммуникациями. Представьте себе, что идеи — это муравьи, ползающие по поверхности воздушного шарика. Если шарик при этом будет надуваться, муравьи смогут продвинуться существенно дальше, чем позволяют их собственные силы.

- Доминирование корпорации Microsoft на рынке программного обеспечения способствует объединению ее конкурентов. Благодаря этому Java-технология получила поддержку широкого круга фирм, не без оснований рассматривавших ее как средство борьбы с монополией Microsoft.
- Пользователи постоянно ищут панацею, способную решить все их проблемы с программным обеспечением, а продавцы — новый лозунг, под которым легко сбывать программный продукт. В цикле этого бесконечного поиска и у тех, и у других в очередной раз срочно возникла потребность в чем-то новеньком. Можно провести аналогию с предыдущими волнами под лозунгами "структурный", "объектно-ориентированный", "клиент-сервер": на соответствующие технологии также возлагались большие надежды, едва ли оправдавшиеся.
- В обстановке общей неудовлетворенности языками Си и Си++ язык Java с энтузиазмом приняли на вооружение преподаватели американских университетов. Данное обстоятельство тесно связано с тем, что университеты США испытывают постоянно усиливающееся давление, заставляющее их все больше заменять фундаментальное образование краткими курсами, ориентированными на потребности рынка. В подобном климате разве что элитные учебные заведения типа MIT могли позволить себе использовать на занятиях современные, но "непромышленные" языки программирования, такие как Scheme (учебный язык, разработанный в MIT. Вариант Лиспа. — Прим. ред.). С приходом Java университеты получили наконец возможность избежать навязываемого им Си/Си++, не опасаясь обвинений во "враждебном отношении к промышленности". В университете шт. Калифорния в Ирвине, где преподает автор этих строк, учебный план уже к осени 1996 г. был практически полностью перестроен под Java. Все прошло на удивление спокойно, почти без споров, хотя до того преподавательский коллектив долгое время был расколот "языковой дискуссией" на два непримиримых лагеря — сторонников Паскаля и сторонников Си.

И наконец, необходимо отметить, что без непрерывной рекламной кампании, проводимой Sun, вероятно, никогда не была бы достигнута та "критическая масса", которая сделала Java ведущей технологией для реализации программных вставок в Web-страницы у многих и многих фирм. Во всяком случае, подавляющее превосходство Java на рынке привело к тому, что лишь немногие из специалистов имеют хоть какое-то представление о существовании конкурирующих технологий, таких как, например, Inferno, разработанная в Bell Laboratories [Lucent]. Очевидно, в Sun усвоили урок, преподанный историей борьбы базовых технологий за рынок (например, VHS против Betamax, DOS против Macintosh): даже с технически менее совершенным продуктом можно добиться успеха, если вывести его на рынок своевременно и энергично.

Что в Java действительно ново?

При ближайшем рассмотрении довольно быстро обнаруживается, что революционных новшеств в Java совсем немного. Правильнее сказать, что Java объединяет ряд давно известных идей, причем поразительно многие из них восходят к работам Никлауса Вирта и его исследовательской группы в Швейцарском федеральном технологическом институте (ETH) в Цюрихе.

- Переносимость Java основана на наличии виртуальной машины [Линдхольм и др.], позволяющей легко имитировать большое число архитектур. Идея виртуальной машины была очень популярна уже более двадцати лет назад, хотя впоследствии о ней забыли. Тогда речь шла о Pascal-P — созданной в ETH реализации Паскаля [Нори и др.], которая сыграла решающую роль в распространении этого языка. Интересно, что виртуальные машины для Паскаля и Java весьма схожи по архитектуре: в обеих используются однобайтовые инструкции без адресов (операнды помещаются в стек).
- Не нова также идея строгой типизации с поддержкой динамических типов и системы "сборки мусора", причем роль первооткрывателя здесь тоже сыграл цюрихский ETH. Многие из аргументов, с помощью которых команда разработчиков Java обосновывает отказ от использования указателей, были сформулированы за пять лет до выхода Java в среде "сообщества модульных языков" (modular language community) при обсуждении разработанного Виртом языка программирования Оберон [Вирт 88]. С давних пор в виртовских языках [Вирт 71, 82, 88] принимаются и другие меры по усилению надежности, такие как проверка индекса при добавлении элементов в массив, — однако до сих пор программисты рассматривали их скорее как досадное ограничение, чем как преимущество. По-видимому, рекламная кампания Sun привела к определенному перелому в общественном мнении по данному вопросу.

- Мысль о том, что язык со строгой типизацией делает возможным построение программ, переносимых с одной архитектуры на другую, в 1990-93 гг. нашла в ЕТН воплощение в целом семействе [Брандис и др.] реализаций Оберон-систем [Вирт-Гуткнехт 89, 92]. Они отличались не только единым интерфейсом прикладных программ для всех аппаратно-программных сред [Франц 93], но и единой машинно-независимой архитектурой документов [Шиперский], уже в то время предусматривавшей встраивание в текст "плавающих" активных объектов. Созданный в развитие идеи пакет Oberon System 3 [Гуткнехт] идет еще дальше, заменяя присущее современным Java-браузерам концептуально неудовлетворительное разграничение между текстом и объектами на единую интегрированную объектную модель.
- Программы на Java могут состоять из нескольких независимо компилируемых единиц, которые "динамически" собираются воедино только в момент загрузки [Франц 97]. В Java поддерживается "истинная" раздельная компиляция, при которой компилятор производит проверку границ между модулями. Программистам, имеющим дело в основном с Си/Си++, это, возможно, покажется значительным усовершенствованием, поскольку там для раздельной обработки применяется крайне примитивный механизм включения непосредственно в текст так называемых "заголовочных файлов". Однако в мире модульного программирования возможности, эквивалентные имеющимся в Java, на протяжении уже более чем десяти лет являются стандартом. Краеугольный камень здесь также заложил Никлаус Вирт, создав язык Модула-2 [Вирт 82].
- Даже преподносимая сейчас в качестве главной инновации Java идея не зависящих от платформы мобильных программ в сочетании с генерацией кода в процессе выполнения еще раньше появилась в ЕТН. Выполненная под руководством Вирта диссертация автора этих строк [Франц 94] посвящена описанию динамически расширяемой системы, в которой исполняемый объектный код "на лету" порождается из промежуточного машинно-независимого формата. В этой диссертации, опубликованной на английском языке в феврале 1994 г., т. е. более чем за год до самой первой презентации Java, уже появляется слово `applet` и описывается ряд сценариев, весьма сходных с существующими сегодня на Java. В настоящее время автор работает над дальнейшим развитием своей системы, обладающей несколькими преимуществами по сравнению с Java [Франц-Кистлер]. В связи с этим может быть интересно то обстоятельство, что в марте 1994 г. автором был прочитан в Калифорнии ряд докладов по теме диссертации, причем один из них — в Sun Laboratories, Inc. Кроме того, упомянутый выше Билл Джой, который переориентировал проект Java на WWW, стал одним из первых обладателей лицензии на Оберон-систему ЕТН, и в конце 1994 — начале 1995 г. он неоднократно связывался с ЕТН; в процессе контактов выяснилось, что он читал мою диссертацию.

Что получится из Java?

Как уже упоминалось в начале статьи, интерес к Java практически неизбежно должен снизиться. Хотя то и дело звучат предложения переписать на Java программное "наследие старины", сама эта задача, при том, что исходным языком является Си или Си++, не вполне тривиальна из-за отсутствия в Java указателей. И вполне реальна опасность, что как раз ради поддержки такого проекта кто-нибудь решит создать специальную версию Java "с указателями", что, естественно, одним махом сведет на нет все преимущества технологии.

Вдобавок фрагментация мира Java грозит распадом языка на ряд несовместимых диалектов, как это, к сожалению, уже не раз случалось с так называемыми промышленными стандартами. А корпорация Microsoft, против которой, как мы все помним, был (по крайней мере отчасти) изначально направлен Java-альянс, буквально выдернула ковер из-под ног у конкурентов. В Microsoft сейчас работают не только над созданием собственной среды программирования на Java, коварно названной J++ (возможно, в отдаленном будущем она не останется полностью совместимой с Java), но, видимо, и над собственными расширениями виртуальной Java-машины.

Выживание Java в качестве учебного языка программирования в высшей школе весьма вероятно, и это должно весьма приветствоваться. Во многих высших учебных заведениях, как в родном университете автора, в прошлом году воспользовались благоприятным моментом, чтобы вставить Java в учебные планы взамен чудища по имени Си/Си++. Вернуть ситуацию в прежнее состояние будет не так-то просто даже в случае, если промышленные проекты и дальше будут реализовываться на Си/Си++. Профессора смогут с чистой совестью заявить, что применяют на занятиях более современный и нацеленный в будущее язык, а благодаря сходству синтаксиса Java с Си/Си++ должны быть удовлетворены и "промышленники".

Как ни смешно, маркетинговая машина Sun сделала для нас, преподавателей информатики, то, чего мы безуспешно добивались не один десяток лет: даже законченные "хакеры" приобщились теперь к добродетелям типизации и абстрагирования данных.

Заключение

В качестве профессора американского университета я должен приветствовать Java уже по той причине, что этот язык позволяет безболезненно отказаться в преподавании от дидактически несостоятельных Си и Си++. Java представляет собой приемлемый компромисс: современный язык программирования, разработанный в традиции, близкой к Паскалю, Модуле-2 и Оберону, но с сознательной попыткой замаскировать это синтаксисом, сходным с синтаксисом Си. В качестве практика я, таким образом, приветствую язык Java везде, куда он планомерно входит как альтернатива Си или Си++. Однако те, кто уже обрел столь же мощный, но эстетически более совершенный язык программирования, такой как Оберон или Ада-95, спокойно могут держаться за него, не боясь оказаться в невыгодной "языковой ситуации".

Литература

1. [Брандис и др.] M. Brandis, R. Crelier, M. Franz, J. Templ. The Oberon System Family // Software Practice and Experience, 1995, vol.25, #12, p. 1331-1366.
2. [Вирт 71] N. Wirth. The Programming Language Pascal // Acta Informatica, 1971, vol.1, #1, p. 35-63.
3. [Вирт 82] N. Wirth. Programming in Modula-2. Heidelberg: Springer-Verlag, 1982.
4. [Вирт 88] N. Wirth. The Programming Language Oberon // Software Practice and Experience, 1988, vol.18, #7, p. 671-690.
5. [Вирт 89] N. Wirth. Modula-2 and Object-Oriented Programming // Proceedings of the First International Modula-2 Conference. Bled, Yugoslavia, 1989. p. 7-13.
6. [Вирт-Гуткнехт 89] N. Wirth, J. Gutknecht. The Oberon System // Software Practice and Experience, 1989, vol.19, #9, p. 857-893.
7. [Вирт-Гуткнехт 92] N. Wirth, J. Gutknecht. Project Oberon: The Design of an Operating System and Compiler. Addison-Wesley, 1992.
8. [Гуткнехт] J. Gutknecht. Oberon System 3: Vision of a Future Software Technology // Software Concepts and Tools, 1994, vol.15, #1, p. 26-33.
9. [Линдхольм и др.] T. Lindholm, F. Yellin, B. Joy, K. Walrath. The Java Virtual Machine Specification. Addison-Wesley, 1996.
10. [Нори и др.] K. V. Nori, U. Amman, K. Jensen, H. H. Naegeli, Ch. Jacobi. Pascal-P Implementation Notes // Pascal: The Language and its Implementation / Ed. by D.W. Barron. Chichester: Wiley, 1981.
11. [Франц 93] M. Franz. Emulating an Operating System on Top of Another // Software Practice and Experience, 1993, vol.23, #6, p.677-692.
12. [Франц 94] M. Franz. Code-Generation On-the-Fly: A Key to Portable Software. Diss. #10497, ETH Zurich; Zurich, 1994 (ISBN 3-7281-2115-0).
13. [Франц 97] M. Franz. Dynamic Linking of Software Components // IEEE Computer, March 1997.
14. [Франц-Кистлер] M. Franz und T. Kistler. Slim Binaries // Communications of the ACM, принято к публикации.

15. [Шиперский] С. А. Szyperski. Write-ing Applications: Designing an Extensible Text Editor as an Application Framework // Proceedings of the Seventh International Conference on Technology of Object-Oriented Languages and Systems (TOOLS'92). Dortmund, 1992, p. 247-261.
16. [JavaSoft] JavaSoft, Inc. What Is Java? <http://www.javasoft.com/nav/whatis>.
17. [Lucent] Lucent Technologies Inc. Inferno. <http://plan9.bell-labs.com/inferno/>.

Михаэль Франц — преподаватель Отделения информации и информатики университета шт. Калифорния в Ирвине. В прошлом — старший научный сотрудник руководимой Никлаусом Виртом исследовательской группы ETH (Цюрих), один из первых участников проекта Oberon и автор созданной в ETH реализации языка Оберон для Macintosh. Области научных интересов — языки программирования и их реализация; расширяемые системы (программная архитектура таких систем, компонентное ПО, переносимое ПО, распространяемое по сети). E-mail: franz@uci.edu, <http://www.ics.uci.edu/~franz>.

Информация к размышлению

Статья Михаэля Франца, любезно предоставленная им для публикации в "Мире ПК", — одна из редких критических работ, где предпринимается попытка осмыслить причины и последствия нынешней эйфории, связанной с технологией Java. К сожалению, автор лишь вскользь упомянул о своих собственных исследованиях в этой области, поэтому читатель, не знакомый с его работами, возможно, сочтет статью неконструктивной.

Михаэль Франц был основным разработчиком системы Oberon System 3 для платформы Macintosh — ОС принципиально нового типа, позволяющей расширение за счет отложенного связывания и реконфигурации программных модулей, а также за счет активного использования так называемых долговременных объектов (persistent objects) для видоизменения пользовательского интерфейса. В системе Oberon любая процедура без параметров в реализованном программистом модуле автоматически становится полноценной командой ОС(!), а текстовый фрагмент любого окна напрямую может быть использован для выполнения произвольных активных действий (это куда более мощные средства, чем ставший для многих уже привычным механизм гипертекста). Проект Oberon почти на десять лет опередил работы Sun Labs по Java и привел к появлению языка Оберон и ориентированной на него аппаратной архитектуры Ceres.

Упомянутая в статье диссертация Франца посвящена анализу недостатков виртуальных машин и подробному изложению особенностей динамической генерации объектного кода для произвольного целевого процессора. Суть решения, предложенного Францем, состоит в том, что исполняемая программа не просто запускается с диска штатным загрузчиком ОС, а сначала "на лету" компилируется из компактного промежуточного представления (причем с проведением полноценной оптимизации!), затем в оперативной памяти компонуется с другими "полуфабрикатными" и объектными модулями приложения, после чего получает управление. Эта схема была реализована в виде протокола OMI (Oberon Module Interchange). Благодаря компактности промежуточного кода время загрузки программ увеличилось не очень сильно (примерно на 30%). Интересно, что схема Франца имеет значительные преимущества по сравнению с куда более поздними реализациями медленных виртуальных Java-машин и так называемых JIT-компиляторов прежде всего за счет не линейной (как в Java), а древовидной структуры промежуточного кода, построенного на принципе семантического словаря.

В Ирвине Франц возглавил коллектив, который к середине прошлого года реализовал OMI в варианте для Internet, получившем название Juice. Загрузчик компактных полуфабрикатных апплетов, кодогенератор (back-end компилятора Oberon) и библиотека основных функций (в том числе экранных) поместилась в дополнительном модуле для Microsoft Internet Explorer и Netscape Navigator (его объем менее 400 Кбайт). Juice существует в редакциях для платформ Wintel и Macintosh, распространяется свободно (<http://www.ics.uci.edu/~juice>).

Руслан Богатырев