

Оберон в Вологодском педагогическом университете

С. З. Свердлов

Вологодский государственный педагогический университет

Введение

Впервые о языке Оберон мы в университете узнали в начале 90-х, когда появились сообщения о лекции профессора Н. Вирта в Московском Университете. Однако описания языка у нас не было. Представление о языке, хотя и неполное, удалось составить после выхода в 1993 году книги В. Ш. Кауфмана «Языки программирования. Концепции и принципы».

Уже первое знакомство позволило понять, что этот язык близок к идеалу для многих применений, в том числе для обучения. Благодаря своей простоте, строгости и неизбыточности он предоставляет начинающему программисту возможность выработать хороший стиль, освоив структурное, модульное и объектно-ориентированное программирование.

Поиски полного описания Оберона в российских библиотеках не дали результата. Только осенью 1995 года с помощью профессора Н. Вирта и московских коллег (мы в то время располагали лишь электронной почтой) удалось получить необходимые материалы с FTP-сервера ЕТН. Позже спецификация Оберона-2 была переведена на русский язык и с любезного разрешения проф. Х. Мёссенбёка и проф. Н. Вирта опубликована в Интернете.

Оберон в учебном процессе

Множество свойств Оберона делают его исключительно подходящим для овладения программированием. Одно из первых ощущений, которое возникает при знакомстве с языком — обозримость. Оберон можно знать целиком и в совершенстве. Ничего подобного нельзя ощутить, например, при использовании диалектов Паскаля компании Borland. Многие современные языки бывают так сложны и запутаны, что почти невозможно требовать от студентов их строгого знания. В этих случаях освоение языка часто строится на переписывании готовых примеров, что никак не способствует воспитанию креативного и конструктивного мышления.

Исключительно полезен лаконизм языка. Например, отсутствие оператора FOR. Замечено, что студенты первого курса склонны употреблять FOR там, где более уместны WHILE или REPEAT. При использовании Оберона проблема исчезает. А когда цикла FOR, а тем более GOTO нет, волей-неволей научишься применять WHILE, REPEAT и LOOP!

Другой пример — строки. Многие программы на тему об обработке строк, написанные первокурсниками на Турбо Паскале, неадекватны, поскольку в они зачастую невольно используют встроенные процедуры и функции для обработки строк без понимания механизмов их работы. В Обероне строка — это просто массив символов и отсюда все следует. Никакие специальные правила не нужны — полная ясность.

Это лишь частные примеры. Общую архитектуру языка, ясную модульную структуру невозможно переоценить.

Оберон в курсе «Языки программирования и методы трансляции»

Этот двухсеместровый курс преследует две основные цели. Создать студентам условия для овладения несколькими языками и освоить конструирование компиляторов. Предполагается, что каждый студент должен овладеть как минимум тремя языками. Основными языками являются Паскаль и Оберон. В качестве третьего языка обычно рекомендуются Си, Ява и Си#. Кроме того, в рамках этого курса изучается объектно-ориентированное программирование.

Для демонстрации техники ООП (наследование – расширение типов, полиморфизм – проверка и охрана типа, связанные процедуры) на Обероне-2 разрабатывается универсальный модуль, реализующий очередь и рассматриваются примеры его применения для создания и обслуживания очередей из разнотипных элементов. Ниже приведен интерфейс этого модуля.

```
DEFINITION Queue;
```

```
TYPE
```

```
  tNode = POINTER TO tNodeRec;  
  tNodeRec = RECORD  
  END;  
  tQueue = RECORD  
  END;
```

```
PROCEDURE Init( VAR Q : tQueue );  
PROCEDURE Put( VAR Q : tQueue; n : tNode );  
PROCEDURE Get( VAR Q : tQueue; VAR n : tNode );  
PROCEDURE NotEmpty( Q : tQueue ) : BOOLEAN;  
PROCEDURE ( VAR Q : tQueue ) Init;  
PROCEDURE ( VAR Q : tQueue ) Add( n : tNode );  
PROCEDURE ( VAR Q : tQueue ) Get( VAR n : tNode );  
PROCEDURE ( VAR Q : tQueue ) NotEmpty( ) : BOOLEAN;
```

```
END Queue.
```

В учебных целях используются как процедуры в стиле Оберон (1), так и связанные процедуры.

Язык программирования «О»

Основу второй части курса составляет разработка компилятора для учебного языка, названного «О» – упрощенного Оберона. Язык «О» включает константы, переменные и выражения типа INTEGER, логические выражения, операторы присваивания, if, while, вызов стандартных процедур ABS, MAX, MIN, DEC, ODD, HALT, INC, In.Int, In.Open, Out.Int, Out.Ln. Модулям In и Out придан статус предопределенных. Вот пример программы на «О»:

```
MODULE Primes;  
IMPORT In, Out;  
VAR  
  d, i, N, c : INTEGER;  
BEGIN  
  c := 0;
```

```
In.Int(N);
i := 2;
WHILE i <= N DO
  d := 2;
  WHILE i MOD d # 0 DO
    INC(d)
  END;
  IF d=i THEN
    INC(c);
    (* Out.Int(d,10) *)
  END;
  INC(i);
END;
Out.Int(c, 0); Out.Ln;
END Primes.
```

О-компилятор генерирует код для простой стековой 0-адресной виртуальной машины. Ниже приводится код, полученный для программы Primes.

0) 4	16) IFGT	32) ADD	48) 8190
1) ENTER	17) 8191	33) STORE	49) DUP
2) 8188	18) 2	34) 20	50) LOAD
3) 0	19) STORE	35) GOTO	51) 1
4) STORE	20) 8190	36) 8191	52) ADD
5) 8189	21) LOAD	37) LOAD	53) STORE
6) IN	22) 8191	38) 8190	54) 11
7) STORE	23) LOAD	39) LOAD	55) GOTO
8) 8190	24) MOD	40) 48	56) 8188
9) 2	25) 0	41) IFNE	57) LOAD
10) STORE	26) 36	42) 8188	58) 0
11) 8190	27) IFEQ	43) DUP	59) OUT
12) LOAD	28) 8191	44) LOAD	60) OUTLN
13) 8189	29) DUP	45) 1	61) STOP
14) LOAD	30) LOAD	46) ADD	
15) 56	31) 1	47) STORE	

После реализации «О» обсуждаются его расширения: процедуры и процедуры-функции, массивы. Студенты получают задания расширить «О», дополнив его операторами REPEAT, FOR, LOOP и EXIT, CASE, типами SHORTINT, REAL и CHAR, константными выражениями. Предлагается разработать генератор кода для виртуального RISC-компьютера, написать конверторы «О» в Паскаль, Яву, Си#, ассемблер x86, PL-ассемблер.

Разработка сканера для О-компилятора проводится в форме конкурса. Побеждает автор самого быстрого сканера. Предлагаются задания по разработке сканеров для реальных языков программирования: Паскаль, Модула-2, Оберон-2, Ада, Си, Си++, Ява, Си#. Более сложные задания предусматривают реализацию подмножеств языков программирования. Студентам предоставляется возможность самим выбрать язык, на котором они будут писать программу. Часто они выбирают Оберон. Так, на Оберон был написан компилятор Small C, несколько вариантов самокомпилятора подмножества Оберона.

Роль Оберона в курсе языков программирования очень велика. Знание Оберона на деле означает владение студентами основными идеями модульного и объектно-ориентированного программирования. Этот язык играет роль ориентира, служит постоянным примером внятного воплощения основных концепций, не позволяя знающим его быть обманутыми в мире современного программирования.

JOB – компилятор Оберона-2 в байт-код JVM

Разработка компилятора Оберона-2 началась в августе 1995 года. Важнейшим стимулом для начала работы служило совершенство самого языка. Я испытывал настоящий азарт при мысли о возможности реализовать Оберон. Лаконизм языка и чистота его дизайна позволяли надеяться на завершение работы в разумные сроки. Кроме того разработка компилятора служила отличной подготовкой к курсу «Языки программирования и методы трансляции», который предстояло читать на вновь образованном отделении прикладной математики нашего университета.

Вначале компилятор был задуман как кроссплатформная система на основе П-кода. В то же время в печати стали появляться первые сообщения о языке Ява. Уже из первых сведений о байт-коде Ява стало ясно, что компилятор мог бы использовать этот формат. При этом появлялась возможность использовать библиотеки Ява. Создание подобных библиотек собственными силами представлялось нереальным.

Особенности компилятора JOB

Компилятор JOB является консольным приложением для платформы Win32. В результате компиляции модуля на языке Оберон-2 компилятор создает один или несколько файлов классов, готовых для исполнения JVM.

В комплект JOB входит набор модулей совместимых с Oakwood Guidelines: In, Out, Files, Strings, Math, MathL, Xyplane. Все эти модули кроме Files написаны на Оберон и компилируются с помощью JOB. Модуль Files написан на языке Java.

JOB позволяет создавать как приложения, так и апплеты. Однако для приложений не реализована концепция команд. По правилам Явы выполнение приложения всегда начинается с метода main:

```
MODULE hello;

IMPORT javaLang, Out;

PROCEDURE main*( VAR args: ARRAY OF javaLang.PString );
BEGIN
    Out.String("hello, world!"); Out.Ln;
END main;

END hello.
```

Пример простейшего апплета:

```
(* First JOB compiled applet "Black Square"
   Designed (1913) by Kasimir Malevich
   Programmed (2.03.98) by s.sverdlov

   <applet code="BlackSquare_App"
       width=200 height=200>
   </applet>
*)
MODULE BlackSquare;

IMPORT app:=javaapplet, awt:=javaawt;

TYPE
    App* = RECORD (app.Applet) END;
```

```
PROCEDURE ( VAR a : App ) paint*( g : awt.PGraphics );
BEGIN
    g.fillRect(30,30,140,140);
END paint;

END BlackSquare.
```

Нужно заметить, что создание апплетов возможно благодаря наличию в Оберон-2 связанных процедур.

Компилятор JOB был опубликован в Интернете в июле 1998 года (<http://www.uni-vologda.ac.ru/job/index.html>). Он оказался первым компилятором с Оберона-2 в байт-код JVM, а демонстрационный апплет Tetris (<http://www.uni-vologda.ac.ru/job/tetris/index.html>) – первым в Интернете Ява-апплетом, написанным на Обероне-2.

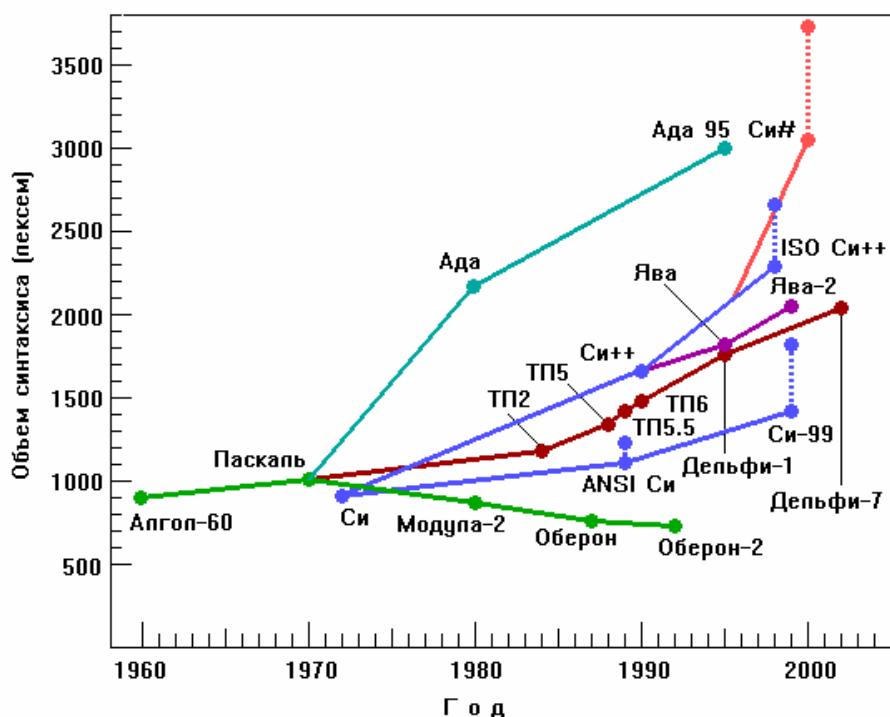
J.Class Optimizer

Поскольку JOB в некоторых случаях формирует файлы классов, содержащие избыточную информацию, была создана специальная программа J.Class Optimizer, которая устраняет избыточность в файлах классов и выполняет ряд других оптимизаций. J.Class Optimizer может быть использован для обработки любых файлов классов. Размер файлов классов уменьшается до 50%. J.Class Optimizer написан на Оберон-2 (примерно 4000 строк) и откомпилирован для платформы Windows с помощью компилятора XDS.

Арифметика синтаксиса

Одновременно с освоением и реализацией Оберон мы знакомимся с языком Ява. В печати появлялись многочисленные статьи, в которых обсуждались свойства нового языка. В частности отмечалось, что Ява более простой язык чем Си++. Испытывая недостаток документации по Яве на русском языке, мы занялись переводом спецификации Явы. Спецификация оказалась объемным и сложным документом. Тезис о простоте Явы подвергся сомнению. Объемы спецификаций Оберона и Явы, вполне сопоставимых функционально языков, различались на порядок. В то же время было понятно, что утверждение о том, что Ява более чем в десять раз сложнее Оберон было бы несправедливо. Стиль этих двух спецификаций был слишком непохожим. Например, в Java Language Specification подробно рассматривается формат представления чисел, все ситуации, которые могут возникать при выполнении операций с ними. Естественным было попытаться выработать более объективные критерии сравнения сложности языков.

Мы решили измерить объем синтаксических описаний Оберона, Явы и других языков. Для этого синтаксис сравниваемых языков был записан в расширенной форме Бэкуса-Наура (РБНФ). Рассмотрены языки Алгол-60, Паскаль, Модула-2, Оберон, Оберон-2, Турбо Паскаль, Объектный Паскаль (Delphi), С (К&R), Си-88, Си-99, С++, Ява, Ада, Си#. С помощью программы анализатора РБНФ подсчитали количество грамматических правил (число нетерминалов), число лексем в синтаксических формулах, количество зарезервированных слов.



Мобильный инструментальный комплекс

Совершенство Оберона и его изысканный минимализм, желание расширить свой опыт в области разработки систем программирования, создать постоянную основу для курсовых и дипломных работ, а также стремление продвинуть Оберон в сферу программирования встроенных систем стимулировало работы по проектированию мобильного многоцелевого инструментального комплекса на базе языка Оберон. Подготовлены предпроектные материалы, на специально организованном семинаре «Разработка инструментальных средств» обсуждалась архитектура комплекса, выполнены ряд исследований.



Разработан модульный транслятор Оберона, написанный на своем собственном входном языке и порождающий код на ANSI Си. Решены вопросы интеграции транслятора с внешними библиотеками, в частности с OpenGL. С использованием этого транслятора начата разработка интегрированной среды. Сделаны ряд разработок, направленных на добавление в систему новых языков, оснащение ее такими инструментами, как отладчик, трассировщик, анализатор тестового покрытия.

Оберон-компиляторы

Мы используем несколько Оберон-систем. Выбор компилятора для выполнения учебных заданий предоставлен студентам. Первыми в 1997 году стали применяться Oberon for Windows (V4) и Oberon/F (educational version). Из этих двух систем студенты обычно выбирали Оберон/F из-за его более привычного пользовательского интерфейса. Позднее мы стали также использовать компилятор JOB и образовательную версию XDS-компилятора для Windows. XDS-компилятор пользуется теперь наибольшей популярностью. В настоящее время он имеет статус свободно распространяемой системы. Небольшое число самых любознательных выбирают для выполнения заданий Оберон V4. Очевидно их привлекает возможность познакомиться на практике с оригинальной концепцией Оберон системы. Компилятор JOB используют для выполнения небольших заданий и для разработки апплетов на Обероне.

Кроме компиляторов сторонней разработки создано несколько собственных экспериментальных реализаций Оберона. Это уже упоминавшийся компилятор JOB (Оберон-2), компилятор подмножества Оберона, написанный на языке Ява и использующий стековую виртуальную машину, самокомпилятор подмножества Оберона, использующий стековую виртуальную машину, конвертор подмножества Оберона в ANSI Си.

Продвижение Оберона

Оберон — совершенный язык, заслуживающий всеобщее внимание. Поэтому мы стараемся предпринимать шаги по его продвижению. В течение 1996-98 годов в

ведущих российских компьютерных еженедельниках были опубликованы несколько статей про Оберон.

В июле 1998 года открылся веб-сайт нашего университета. Одними из первых его материалов стали страница про язык Оберон и страница о компиляторе JOB.

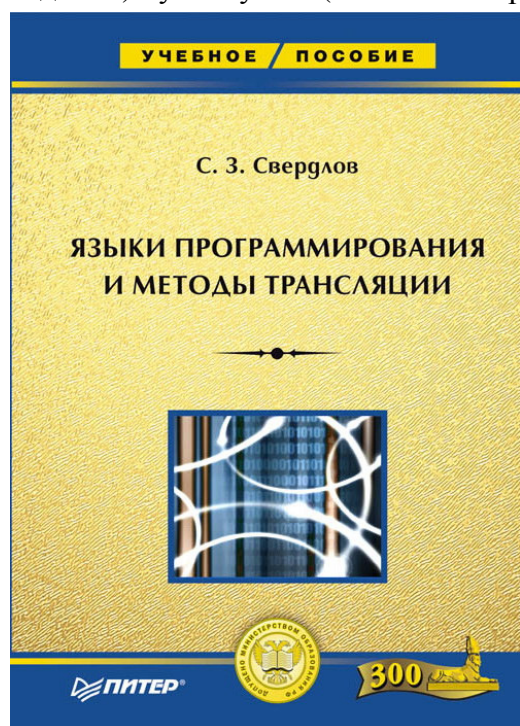
Страница про язык Оберон и в настоящее время является одним из основных ресурсов русского Интернета по языку Оберон.

Тогда же стенд нашего университета участвовал в первом Интернет-форуме «Языки программирования и Интернет», организованном российским информационным Интернет-агентством «Infoart».

Учебное пособие «Языки программирования и методы трансляции»

К новому 2005-2006 учебному году российское издательство «Питер» выпускает книгу «Языки программирования и методы трансляции» — учебное пособие по одноименному курсу для студентов, обучающихся по специальности прикладная математика и информатика.

Важную роль в книге играет Оберон. В приложении (впервые в печатном издании) публикуется (с любезного разрешения проф. Н. Вирта и проф. Х. Мёссенбёка)



русский перевод спецификации языка Оберон-2.

При изучении наук примеры полезнее правил — утверждал И. Ньютон. Это тезис, если речь идет о науках прикладных, в частности о программировании и о разработке компиляторов, актуален необычайно. Попытка применения конструктивного, основанного на конкретных примерах, подхода при построении курса «Языки программирования и методы трансляции» сделана в книге, написанной на основе курса, который автор в течение ряда лет читает студентам факультета прикладной математики и компьютерных технологий. Практика показала, что прошедшие этот курс студенты хорошо понимают методы построения компиляторов, в состоянии самостоятельно реализовать несложный транслятор. Предметность и конструктивный подход способствуют формированию высокого уровня мотивации, многие студенты берутся за задачи

повышенной сложности и не без успеха решают их.

В книге дается подробная характеристика Оберона и история создания языка. Он используется как основной инструмент при рассмотрении объектно-ориентированного программирования. Основное содержание главы о трансляции языков программирования — разработка компилятора для подмножества Оберона. Текст этого компилятора, написанного на Обероне, тоже приводится в книге.

Вместо заключения

Весной 2003 года студенческая команда нашего университета (в числе 8 команд из России и 16 из Европы) прошла в финал чемпионата мира ACM по программирова-

С.З.Свердлов, Доклад на II Всеросс. совещании по ИТ-образованию, 27-29 июня 2005, МГУ

нию, проходивший в Калифорнии. Команда носила гордое имя «Оберон». Уверен, что такой успех ребят из провинциального университета обусловлен и тем, что они изучали Оберон — язык, который способствует формированию ясного взгляда на многие вещи в программировании.