



Руслан Богатырев

## Язык Оберон. Краткий путеводитель

### 1. Представление языка

Автор	Никлаус Вирт (Niklaus Wirth, р. 1934)
Год рождения	1988
Место рождения	Швейцария, Цюрих. Высшая Политехническая школа ETH
Каноническое описание	Oberon Language Report. Revised Edition. 01.10.1990

В противовес промышленным языкам (включение максимума средств, принцип сундука) исповедует принцип чемоданчика (только самое необходимое). Проще и при этом мощнее канонического Паскаля.

Поддерживает такие парадигмы (подходы) программирования, как

- процедурное (procedural programming);
- модульное (modular programming);
- программирование абстрактных типов данных (ADT programming);
- расширяющее (проеекционное, extensible programming);
- объектно-ориентированное (object-oriented programming);
- компонентное (component-oriented programming);
- системное (systems programming);
- рефлексивное (reflective programming).

Весь ввод-вывод, параллельное программирование и обработка исключений вынесены на уровень внешних библиотек.

В основе языка лежат модули и расширение комбинированных типов (RECORD). Это позволяет выступать Оберону в качестве основы построения расширяемого каркаса программной системы (с подключением компонентов без перекомпиляции) и как ассемблеру ООП.

Крайне прост в изучении и освоении. Легко воспринимается теми, кто знаком с Паскалем.

Поддерживает преемственность базовых идей программирования. Ниде приведен псевдокод, отражающий в синтаксисе языка эту преемственность и демонстрирующий ключевой механизм расширения (наследования) типов.

Рис. Три кита структурного программирования, Паскаля и Оберона

```
IMPORT
  Dijkstra, Wirth, Hoare, (* авторы идей: Дейкстра, Вирт, Хоар *)
  Mesa, Modula2;          (* языки-источники идей: Mesa, Modula-2 *)

TYPE
  StructuredProgramming = (* 1966 *)
    RECORD
      Следование : Dijkstra.Sequence;
      Ветвление  : Dijkstra.Selection;
      Цикл        : Dijkstra.Iteration;
    END;

  Pascal = (* 1970 *)
    RECORD (StructuredProgramming)
      Синтаксис      : Wirth.Syntax;
      ОпределениеТипов : Wirth.TypeStructures;
      УказателиНаТипы : Hoare.PointerBinding;
    END;

  Oberon = (* 1988 *)
    RECORD (Pascal)
      Модули          : Mesa.Modules;
      ПроцедурныеТипы : Modula2.ProcTypes;
      РасширяемыеТипы : Wirth.TypeExtension;
    END;
```

## 2. Визитная карточка языка

Базовые типы	BOOLEAN, CHAR, SET, SHORTINT, INTEGER, LONGINT, REAL, LONGREAL
Операции	+, -, *, /, DIV, MOD, &, OR, ~, IN, IS, =, #, <, <=, >, >=
Конструкторы типов	ARRAY, RECORD, POINTER, PROCEDURE
Эквивалентность типов	Именная. Совместимость по включению для базовых числовых типов. Преобразование типов через процедуру VAL.
Операторы ветвления	IF (ELSIF), CASE
Операторы цикла	LOOP, WHILE, REPEAT
Операторы выхода	EXIT, RETURN
Комментарии	(* *)
Процедуры	Процедуры и процедуры-функции. Вложенность допустима.
Передача параметров	По ссылке (VAR) и по значению.
Стандартные процедуры	INC, DEC, INCL, EXCL, COPY, NEW, HALT
	ABS, ODD, CAP, ASH, LEN, MAX, MIN, SIZE
	ORD, CHR, SHORT, LONG, ENTIER
Модули	Явный экспорт и импорт, псевдонимы, квалификация импортируемых идентификаторов. Вложенность недопустима.
Системные типы	BYTE
Системные процедуры	ADR, BIT, CC, LSH, ROT, VAL, GET, PUT, MOVE, NEW

**Специфика ООП в Обероне.** Особенность языка в том, что он ввел механизм наследования через тип (RECORD), а не через модуль (CLASS). Это дало возможность более аккуратно управлять областью видимости: несколько RECORD-классов можно включать в капсулу одного модуля, что дает возможность получать им доступ к деталям реализации соседей без нарушения принципа сокрытия информации и без лишних накладных расходов. Методы в Обероне представляются полями процедурных типов (процедурные типы Оберона взяты в C# за основу представления делегатов).

Не нужны пространства имен — вместо этого есть восходящая к Парнасу четкая концепция модуля, оперирующего понятием экспорта-импорта сущностей языка (константы, типы, поля типов, переменные, процедуры).

Модули поддерживают полноценную раздельную компиляцию (интерфейсная часть извлекается автоматически по имеющейся маркировке экспорта), являются единицей загрузки и основой построения программных компонентов.

## 3. Отличия от Паскаля

### А. Синтаксис

1. Важен регистр букв в названии идентификаторов.
2. Зарезервированные идентификаторы языка всегда пишутся большими буквами.
3. Убраны скобки begin-end. Начало оператора определяется его именем, окончание — словом END. Для модулей и процедур требуется в конце повторять свое имя после слова END. Комментарии обозначаются связкой (\* \*).

### В. Типы данных

1. Убраны перечисления, диапазоны. Тип SET определяет набор (множество) целых в диапазоне от 0 до MAX(SET). Введен процедурный тип (PROCEDURE) с операциями вызова и присваивания.
2. Индексы массивов нумеруются с нуля. Введены многомерные открытые массивы (ARRAY OF).
3. Убраны варианты записи. Они заменяются расширяемыми комбинированными типами (см. E1).

### С. Процедурное программирование

1. Убрано деление на процедуры и функции: процедуры могут возвращать значения через свое имя.
2. Убран оператор FOR как источник ошибок.
3. Добавлен универсальный цикл LOOP-END с оператором выхода EXIT.

## D. Модульное программирование

1. Исключено понятие программы (Паскаль), как и деление на программные и библиотечные модули (Modula-2). Есть единое понятие модуля — единицы компиляции и исполнения (загрузки). Он может экспортировать сущности: константы, типы, поля типов, переменные, процедуры. Вложенные (локальные) модули запрещены. Экспортируемые процедуры без параметров называются командами и определяют точки вызова модуля (программы).
2. Для экспорта используется значок \* (звездочка) при определении сущности сразу после ее имени. Интерфейсный (описательный) модуль строится автоматически компилятором.
3. Импортируются только модули (IMPORT). Допустимо введение псевдонима модуля. Импортируемые идентификаторы в тексте обязательно квалифицируются именем соответствующего модуля.

## E. Объектно-ориентированное программирование

1. Комбинированный тип (RECORD) может расширяться путем добавления полей записи (в том числе и процедурных типов). Это определяет механизм проекции типов, аналогичный наследованию в традиционном ООП.
2. Для работы с расширяемыми типами введена операция принадлежности типу (IS), а также оператор WITH (привратник типа).
3. Язык предусматривает механизм автоматической сборки мусора.

## F. Системное программирование

1. Введен псевдомодуль SYSTEM, который является интерфейсом между языком и низкоуровневыми средствами. Импортирование SYSTEM сигнализирует о привязке к конкретной операционной платформе.
2. Введены процедуры адресной арифметики (ADR, BIT, LSH, ROT, CC), преобразования типа (VAL), работы с памятью (GET, PUT, MOVE) и резервирования области памяти (NEW).
3. Введен тип BYTE. Формальный параметр вида ARRAY OF BYTE, передаваемый по ссылке, совместим с любым типом.

## 4. Краткая характеристика основных языков Оберон-семейства

1. **Оберон** (Никлаус Вирт, 1988) — изящный компактный язык (меньше и проще Паскаля), идеален для преподавания основ информатики (computer science), концепций структурного, модульного и объектно-ориентированного программирования (ООП). Хорошо подходит для реализации малых и средних проектов. Имеет компиляторы внутри Oberon System и Juice. Блестящий кандидат на роль эсперанто программирования.
2. **Oberon-2** (Ханспетер Мессенбок, Никлаус Вирт, 1991) — развитие Оберона в сторону привычного ООП, в связке с Modula-2 (как языка системного программирования) создает хорошую основу для реализации крупных проектов из макромира и микромира (встроенных систем, систем реального времени). Двухязыковая связка отлично реализована в системе XDS (Excelsior). Позволяет осуществлять кросс-разработку в Win32/Linux за счет трансляторов промышленного качества в Си и C++. Если нужно обобщение алгоритмов (задействование ООП), написание автономных программ, устойчивая работа с ОС на уровне системных вызовов, использование внешних библиотек на других языках, перенос на другие платформы через Си/C++, эффективная реализация (оптимальный объектный код), то подходит Oberon-2 в исполнении XDS.
3. **Компонентный Паскаль/Component Pascal** (Клеменс Шиперски, Куно Пфистер, 1997) — развитие Оберона и Oberon-2 в сторону компонентно-ориентированного программирования (КОП). Хорошо проявляет себя для программирования в большом. Если требуется строить расширяемую систему с использованием КОП, подходов программной инженерии, иметь прямой выход на современные наработки для Win32, .NET и Java Platform, то нужен Component Pascal в реализациях BlackBox и GPCP. В реализации BlackBox обладает уникальной особенностью динамического расширения систем ("на лету") за счет поддержки Оберон-компонентов, легко настраивается на решение задач любого уровня сложности (от преподавания информатики в школах до сложных исследовательских систем), имеет средство формирования COM-компонентов со сборкой мусора (Direct-To-COM Compiler), получившее на CeBIT приз за технологическое совершенство.
4. **Active Oberon** (Юрг Гуткнехт, Патрик Реали, 2000) — воплощение в Обероне идеи активных объектов (мультипроцессные системы для многопроцессорных конфигураций). Исследовательский проект, выполняемый группой проф. Гуткнехта (ETH, Цюрих). Базовый язык для реализации ОС Bluebottle — дальнейшего развития системы Oberon. Поддерживает работу в Win32, на голой машине (PC), сосуществует с Java и .NET. Показал отличную эффективность в специфических областях (мультимедиа, повсеместный компьютеринг).
5. **Zonnon** (Юрг Гуткнехт, Евгений Зуев, 2003) — ревизия Modula-2 и Оберона сквозь призму идей языка Mesa (Xerox PARC) и платформу Microsoft .NET. Исследовательский проект группы проф. Гуткнехта. Ориентирован в большей степени на особенности реализации новых языков и компиляторов для платформы

.NET с последующей интеграцией в Microsoft Visual Studio .NET (2005). Представляет интерес в преподавании на старших курсах вузов, несколько сложен для начинающих программистов, но достаточно красиво решает проблемы "мирного сосуществования" модульного программирования, ООП и КОП. Представитель композиционного программирования.

## 5. Основные системы программирования

Платформы	Oberon-2	Component Pascal
S=W32 T=W32	XDS-Win	BlackBox
S=W32, T=JAV	JOB	GPCP-JVM
S=W32, T=NET	—	GPCP-NET
S=W32, T= ANY	XDS-C-Win	—
S=JAV, T=JAV	—	GPCP-Eclipse
S=LNx, T=LNx	XDS-Linux, OO2C	—
S=LNx, T=ANY	XDS-C-Linux	—

### Условные обозначения:

S — инструментальная платформа  
T — целевая платформа

W32 — Win32

NET — .NET

JVM — Java (байт-код)

LNx — Linux

ANY — любая операц. платформа (Си)

BlackBox — BlackBox Component Builder [Oberon microsystems]  
XDS-Win — Native XDS-x86 for Windows [Excelsior]  
XDS-Linux — Native XDS-x86 for Linux [Excelsior]  
XDS-C-Win — XDS-C for Windows [Excelsior]  
XDS-C-Linux — XDS-C for Linux [Excelsior]  
GPCP-NET — Gardens Point Component Pascal on .NET [QUT]  
GPCP-JVM — Gardens Point Component Pascal on JVM [QUT]  
GPCP-Eclipse — Gardens Point Component Pascal for Eclipse [QUT]  
JOB — JOB Compiler (Oberon-2 → JVM) [С.Свердлов]  
OO2C — Oberon-2 to ANSI-C Translator [OpenSource]

## 6. Эталонные описания языков Оберон-семейства

### 1. Оберон

N. Wirth

"The Programming Language Oberon" (01.10.1990)

### 2. Oberon-2

H. Moessenboeck, N. Wirth

"The Programming Language Oberon-2" (March 1995)

### 3. Component Pascal

Oberon microsystems

"Component Pascal Language Report" (March 2001)

### 4. Active Oberon

P. Reali

"Active Oberon Language Report" (27.10.2004)

### 5. Zonnon

J. Gutknecht, E. Zueff

"Zonnon Language Report" (October 2004)

## 7. Источники информации

ETH Oberon

Проект "Информатика-21"

Королевство Delphi

<http://www.oberon.ethz.ch>

<http://www.inr.ac.ru/~info21/>

<http://www.delphikingdom.ru/asp/talk.asp>

Oberon microsystems

Excelsior

QUT

JOB

OO2C

Active Oberon

Zonnon

<http://www.oberon.ch>

<http://www.excelsior-usa.com/>

<http://www.plas.fit.qut.edu.au/gpcp>

<http://www.uni-vologda.ac.ru/~c3c/>

<http://ooc.sourceforge.net/>

<http://www.bluebottle.ethz.ch/>

<http://www.zonnon.ethz.ch/>