

Приложение F. Сопоставление со стандартом ISO Pascal

В этом приложении дается сопоставление UniPascal-я со стандартом ISO Pascal, описанным в книге K.Jensen N.Wirth, *Pascal User Manual and Report*, Springer-Verlag, 1974, 1985 (ISBN 3-540-96048-1).

F.1. Нереализованные возможности ISO Pascal-я

- В ISO Pascal-е идентификатор может быть неограниченным по длине, при том все символы имени значащие. В UniPascal-е идентификатор может содержать неограниченное число символов, но только первые 8 являются значащими.
- В ISO Pascal-е символ @ является альтернативным представлением символа ^ . В UniPascal-е символ ^ не имеет другого представления.
- В ISO Pascal-е разрешается использование различных комментарных скобок в одном комментарии, т.е. комментарий может начинаться с (* и заканчиваться другой скобкой } и наоборот. В UniPascal-е этого нельзя делать. Комментарии должны начинаться с скобки одного типа и заканчиваться скобками того же самого типа.
- В ISO Pascal-е каждое значение множества значений, принадлежащих типу поля вариантной части записи, должно быть указанным. В UniPascal-е не существует такое требование.
- В ISO Pascal-е каждой переменной файлового типа ассоциирована буферная переменная. В UniPascal-е файловым переменным не ассоциируются буферные переменные.
- В ISO Pascal-е поле-селектор вариантной части записи не может быть использованным в качестве фактического параметра-переменной. В UniPascal-е такое требование не существует.
- В ISO Pascal-е имеется возможность использовать процедуры в качестве параметров. В UniPascal-е этого невозможно.
- В ISO Pascal-е стандартные процедуры RESET и REWRITE не требуют имени физического файла. В UniPascal-е синтаксис стандартных процедур RESET и REWRITE расширен и требует имени внешнего файла.
- В ISO Pascal-е дефинированы стандартные процедуры GET и PUT, которые используются для чтения из файла и для записи в нем. В UniPascal-е эти процедуры не реализованы.
- В ISO Pascal-е дефинированы стандартные процедуры PACK и UNPACK, которые используются упаковать и распаковать пакетированные переменные. В UniPascal-е эти процедуры не реализованы.
- В ISO Pascal-е операция $I \bmod J$ всегда в качестве результата дает положительное число и выдается ошибка, если $J \leq 0$. В UniPascal-е операция $I \bmod J$ вычисляется как $I - (I \text{ div } J) * J$ и ее можно использовать при отрицательных значениях J .
- В ISO Pascal-е оператор GOTO может использовать дефинированную в внешнем блоке метку. В UniPascal-е эта ситуация запрещена и компилятор сообщает ошибкой об ее наличии.
- В ISO Pascal-е селекторное выражение (selector) в CASE операторе должно иметь значение, одинаковое с одной из использованных констант. В UniPascal-е этого не

требуется. Если значение выражения не совпадает ни с одной из них, выполняется оператор ELSE части CASE оператора. Если такой нет, ничего не делается.

- В ISO Pascal-е не разрешается изменить значение управляющей переменной FOR цикла. В UniPascal-е проверка не производится.
- В ISO Pascal-е стандартная процедура READ присваивает пробел переменной типа CHAR, если до чтения EOLN = TRUE. В UniPascal-е переменной типа CHAR в той же ситуации присваивается <cr> (ASCII код 13).
- В ISO Pascal-е при выполнении стандартной процедуры WRITE в текстовом файле переменной типа упакованной строки урезается, если указанная длина меньше чем необходимую. В UniPascal-е всегда записывается вся строка, независимо от указанного значения длины.
- В ISO Pascal-е допускается использование переменных логического типа в стандартной процедуре WRITE. В UniPascal-е этого невозможно сделать.
- В ISO Pascal-е дефинирована стандартная процедура PAGE, которой вызывается переход к новой странице. В UniPascal-е такой процедуры нет. Обычно, используется WRITE(F,CHR(12)).
- В ISO Pascal-е дефинированы совместимые массивы-параметры. В UniPascal-е параметров такого вида нет.
- В ISO Pascal-е дефинирована стандартная константа MAXINT. В UniPascal-е эта константа не дефинирована. Вместо нее можно использовать функцию MAX(INTEGER).

F.2. Расширения относительно ISO Pascal-я

- В UniPascal-е следующие идентификаторы добавлены к зарезервированным словам: **INTERFACE, IMPLEMENTATION, SEGMENT, UNIT, USES, XOR**.
- Символ подчеркивания '_' выполняет роль буквы.
- Целые константы можно записывать в шестнадцатеричной системе счисления. Тогда символ \$ предшествует константе.
- В качестве меток можно употреблять любые идентификаторы.
- Синтаксис символьной строки (символьной константы) расширен. Они могут содержать управляющие символы и могут заключаться в кавычках (") или апострофах (').
- Описание меток, констант, типов, переменных, процедур и функций может появляться в тексте программы в любом порядке и разделы могут появляться в тексте много раз.
- Везде, где в ISO Pascal-е требуется константа, в UniPascal-е можно использовать константное выражение.
- В UniPascal-е реализованы дополнительные целые типы - SHORTINT, SHORTCARD, LONGINT, CARDINAL, NATURAL, а так же и дополнительные типы BYTE, WORD, LONGWORD.
- В UniPascal-е реализован стандартный тип string, который отличается от упакованных массивов символов тем, что он включает динамическую длину строки.
- Совместимость типов расширена, чтобы совместимость между типами STRING, CHAR и PACKED ARRAY OF CHAR стала возможной.

- Тип выражения или переменной может быть изменен (Type Cast).
- Дефинирована логическая операция XOR - исключаящее "или".
- Операции AND, OR, XOR можно применять над типами BYTE, WORD и LONGWORD.
- Операции отношения - могут применяться над упакованными массивами символов и над строками.
- Операции отношения равенства (=) и различия (<>) могут применяться над операндами любого типа за исключением файлового типа.
- Синтаксис CASE оператора расширен и включает в нем ELSE часть.
- Расширяется классификация параметров введением нового вида - параметр-константа.
- Параметры-константы и параметры-переменные могут быть нетипизированными, при том сопоставляемые им фактические параметры могут иметь любой тип.
- Фактические параметры, сопоставляемые формальным параметрам типа BYTE, WORD и LONGWORD, могут быть переменными любого типа с размерами 1, 2 и 4 байта, соответственно.
- В UniPascal-е реализованы модули (UNIT), чем обеспечивается раздельно-модульная компиляция.
- В UniPascal-е дефинированы следующие стандартные процедуры и функции, которых нет в ISO Pascal-е:

| | | |
|-------------|----------|------------|
| ADDR | HALT | MOVE |
| APPEND | HI | MOVEWORDS |
| BLOCKREAD | HIWORD | MSGLIMIT |
| BLOCKWRITE | INC | OPEN |
| CLOSE | INCL | PARAMCOUNT |
| DEC | INSERT | PARAMSTR |
| DELETE | INT | POS |
| EXCL | IORESULT | RANDOM |
| EXIT | LENGTH | RANDOMIZE |
| FILEPOS | LO | RELEASE |
| FILESIZE | LONGSWAP | RETURN |
| FILLCHAR | LOWORD | SEEK |
| FILLWORD | MARK | SIZEOF |
| FRAC | MAX | SWAP |
| FREEMEMWORD | MAXAVAIL | TRUNCATE |
| GETDATE | MEMAVAIL | UPCASE |
| GETMEMWORDS | MIN | SIZEOF |
| GETTIME | | |

- В UniPascal-е возможно использование одного из четырех синтаксических описаний операторов. Четыре множества представляют собой, соответственно: язык Pascal; расширение языка Pascal; язык, подобный Modula-2; расширение этого Modula-2 подобного языка.

