

## Модуль-2

### Тип записи

Не должен быть пустым (не содержащим полей), поскольку его нельзя расширить. Оператор WITH используется для короткого доступа к полям такой переменной.

### Процедура

Параметр-переменная (с идентификатором VAR) не может быть выражением или константой. В качестве результата процедуры-функции не могут выступать: запись, массив, поддиапазон, перечисления, множество. Иначе надо использовать процедуру с параметром-переменной (с идентификатором VAR). Поскольку указатель имеет целочисленный тип, он может быть аргументом процедуры и результатом процедуры-функции. Идентификатор VAR говорит о том, что эта переменная может модифицироваться в процедуре и хранить полученный результат.

## Оберон, Оберон-2

### Тип записи

Может быть пустым (не содержащим полей), поскольку его можно расширить. Базовому типу можно присвоить значение расширенного типа, но не наоборот. Указателю базового типа можно присвоить значение указателя расширенного типа, но не наоборот.

### Процедура

Идентификатор VAR говорит о том, что эта переменная может модифицироваться в процедуре и хранить полученный результат. Если используемая в первой процедуре вторая процедура описана ниже первой процедуры, то будет ошибка компиляции. Для правильной компиляции программы необходимо перед первой процедурой записать заголовок второй процедуры, в котором после слова PROCEDURE поставить символ ^, то есть PROCEDURE^.

### Процедура, связанная с типом записи

Объявление такой процедуры содержит параметр-приёмник. Связь с типом записи выражается типом приёмника в заголовке объявления процедуры. Приёмник может быть или типом записи в виде параметра-переменной (с идентификатором VAR), или указателем типа записи в виде параметра-значения. Если процедура связана с базовым типом записи, то она неявно связана с любым расширением этого типа. Однако другая процедура, которая явно связана с расширением базового типа записи, с тем же самым именем, что и процедура, связанная с базовым типом записи, перекрывает её и рассматривается как её переопределение. Формальные параметры обеих процедур должны совпадать. Если к вызову переопределённой процедуры добавить ^, то вызывается процедура, связанная с базовым типом записи, — так называемый супервызов. Осуществить супервызов можно только из переопределённой процедуры. Это связано с инкапсуляцией данных базового типа записи в расширенном типе записи.

Преимуществами процедур, связанных с типом записи, являются следующие:

1. Во время создания переменной данного типа записи нет риска забыть присвоить процедуру, как в случае процедурного типа.
2. Память, потребляемая процедурными переменными, будет меньше.

### Процедурный тип

Используется в типах записи, число и тип параметров определяются при проектировании. Обязательно в списке параметров фигурирует параметр-переменная (с идентификатором VAR), соответствующая базовому типу записи. Для каждого расширенного типа записи процедурному типу назначается своя процедура. Поскольку в параметрах процедуры указана базовый тип записи, в теле процедуры обязательно должна быть проверка типа на соответствие этому расширенному типу записи. Преимуществом процедурного типа является отсутствие переопределения процедуры в смысле процедуры, связанной с типом записи и отсутствие необходимости менять тип параметра-приёмника и нет необходимости в специальных символах для супервызова. Недостатком является возможность забыть присвоить процедуру при создании переменной типа записи.

### Проверка типа

Она осуществляется с помощью одного из двух операторов - IS или WITH. Проверка может осуществляться либо сравнением разыменованного указателя с типом записи (базовым или расширенным), что

условно может называться проверкой по типу, либо сравнением указателя с указателем на тип записи (базовый или расширенный), что условно может называться проверкой по указателю. Проверка по типу должна осуществляться в отдельной процедуре, поскольку разыменованный указатель или переменная типа записи считаются статическим типом и сравнение с ними надо проводить с передачей параметра-переменной с идентификатором VAR, который преобразует входной статический тип в динамический (который может изменяться), или ещё с идентификатором IN для Компонентного Паскаля. Проверка по указателю может проводиться и без отдельной процедуры, поскольку указатель считается динамическим типом (ему может быть присвоено любое допустимое значение). Оператор IS по указателю (или разыменованному указателю) формирует динамический тип данных, находящихся по данному адресу. Оператор IS рассматривает указатель p или разыменованный указатель p^ как динамический объект, который на момент проверки имеет тип T, и обращается к нему как p(T), а к полям записи как p(T).имя поля. Оператор WITH проводит проверку аналогично оператору IS, отличие только в том, что он рассматривает указатель p или разыменованный указатель p^ как статический объект типа T, и обращается к нему просто как p, а к полям записи как p.имя поля, хотя допустимой является и запись p(T).

При проверке операторами IS и WITH сравнение надо начинать с расширенных типов записей, а в конце с базовым типом записи, поскольку, если начинать сравнение с базового типа записи, то расширенные типы записи тоже будут соответствовать базовому типу записи и сравнение будет ограничиваться только этой веткой сравнения, а другие ветки сравнения будут недоступны.

### Экспорт переменных и типов

Экспортируемые переменная и тип становятся глобальными в группе, куда входят модули, которые их импортируют, и модуль, который их экспортирует.

В импортирующем модуле для доступа к полям импортированной переменной типа записи в экспортирующем модуле достаточно экспортировать только поля типа записи. Если в импортирующем модуле необходимо создавать переменные импортированного типа записи, то в экспортирующем модуле необходимо экспортировать тип записи. При этом если не экспортировать поля экспортируемого типа записи, то в импортирующем модуле можно создавать переменную импортированного типа записи, но без доступа к полям типа записи.

Экспорт «только для чтения» используется только для переменных и полей типа записи.

### Особенности расширения типа записи

Тип записи, содержащий в качестве своего поля другой тип записи, не может считаться его расширением (хотя структура его полей идентична аналогичному расширенному типу записи), поскольку в языке выбрана номинальная (описанная в определении типа связь между типами записи), а не структурная типизация (по совпадению структуры записи).

## **Компонентный Паскаль**

### Тип записи

В качестве типа записи может быть использован указатель.

Для типа записи с атрибутом ABSTRACT нельзя создать переменную такого типа записи (чтобы подчеркнуть его абстрактный характер), но можно расширять и экспортировать.

Для типа записи с атрибутом EXTENSIBLE можно создать переменную такого типа, расширять и экспортировать.

Для типа записи с атрибутом LIMITED можно создать переменную такого типа только в том модуле, в котором он определён, а также расширять и экспортировать.

Для типа записи без атрибутов можно создать переменную такого типа, экспортировать, но нельзя расширять.

### Процедура

Идентификатор IN говорит о том, что эта переменная служит только для ввода данных и не может быть изменена в процедуре. Идентификатор IN допускается только для переменных типов записи и массив. Константы нельзя помечать идентификатором IN. Идентификатор OUT говорит о том, что эта переменная служит только для вывода результата и её значение в начале процедуры не определено. Всем переменным с этим идентификатором, содержащим указатели и процедурные типы, в начале процедуры присваивается значение NIL. Идентификатор VAR говорит о том, что эта переменная может модифицироваться в процедуре и хранить полученный результат.

### Процедура, связанная с типом записи

Процедура, возвращающая указатель на базовый тип записи, может быть переопределена таким образом, чтобы возвращать указатель на расширенный тип записи и быть связанной с расширенным типом записи.

Атрибут NEW используется для вновь создаваемой для данного типа записи процедуры (её имя должно быть уникальным для базового типа записи и его расширений). Используется для всех атрибутов типов записи.

Атрибут ABSTRACT указывает, что процедура является пустой (не содержит код). Такая процедура используется только для типа записи с атрибутом ABSTRACT и при создании переменной типа записи она должна быть определена.

Атрибут EXTENSIBLE указывает, что процедура может быть переопределена при расширении типа записи и используется для всех атрибутов типов записей.

Атрибут EMPTY указывает, что процедура является пустой (не содержит код). В отличие от процедуры с атрибутом ABSTRACT, при создании переменной типа записи такую процедуру не надо переопределять (или переопределять в случае необходимости). Она используется для типов записи с атрибутами ABSTRACT или EXTENSIBLE, и если она не переопределялась, то её вызов не даёт никакого эффекта. Такие процедуры не могут быть процедурами-функциями и не могут иметь OUT параметров.

### Описание свойств различных атрибутов типа записи

Атрибут	Создание переменной типа записи	Расширение	Экспорт
Без атрибута	Да	Нет	Да
ABSTRACT	Непосредственно нельзя, только через расширение	Да	Да
EXTENSIBLE	Да	Да	Да
LIMITED	Только в определяющем этот тип записи модуле	Да	Да

### Особенности расширения различных атрибутов типа записи

Атрибут расширенного типа записи	Без атрибута	ABSTRACT	EXTENSIBLE	LIMITED
Атрибут базового типа записи				
ABSTRACT	Да	Да	Да	Да
EXTENSIBLE	Да	Нет	Да	Да
LIMITED	Нет	Нет	Нет	Да