

Урок 89. Размерные и ссылочные типы

Большинство типов в PascalABC.NET подразделяются на две большие группы: *размерные* и *ссылочные*. Кроме того, имеется несколько типов, унаследованных от Delphi Object Pascal, которые трудно отнести к размерному или ссылочному типу.

Размерные и ссылочные типы отличаются следующими характеристиками:

- выделение памяти
- управление памятью
- присваивание
- сравнение
- передача в подпрограммы

1. Размерные и ссылочные типы в PascalABC.NET

<i>Размерные</i>	<i>Унаследованные от Delphi Object Pascal</i>	<i>Ссылочные</i>
Все простые типы и записи	Статические массивы, множества, размерные строки и файлы	Строки, динамические массивы, кортежи, классы, последовательности, процедурный тип
Наследуются от .NET-типа System.ValueType		Наследуются от .NET-типа System.Object и не наследуются от System.ValueType
Более эффективны при вычислениях: они занимают меньше памяти и операции, выполняемые над небольшими размерными типами, максимально эффективны.	По представлению в памяти относятся к ссылочному типу, но по поведению (присваивание, сравнение, передача в подпрограммы) - к размерному.	Обладают большей гибкостью: память под них выделяется динамически в процессе работы программы и освобождается автоматически когда объект ссылочного типа перестаёт использоваться.
Память выделяется при запуске программы (для глобальных переменных) или в момент вызова подпрограммы (для локальных переменных).		Для инициализации ссылочных переменных используется вызов конструктора соответствующего класса
Переменная размерного типа хранит значение этого типа.		Переменная ссылочного типа хранит ссылку на объект некоторого класса в динамической памяти. Если она не инициализирована, то хранит специальное значение <code>nil</code> (нулевая ссылка). Однако, строки инициализируются пустой строкой.
Память освобождается при завершении программы (для глобальных переменных) или при выходе из подпрограммы (для локальных переменных)	С точки зрения распределения памяти относятся к ссылочным и память, занимаемая значениями этого типа, также управляется сборщиком мусора.	Управление памятью для ссылочных типов осуществляется автоматически сборщиком мусора.

2. Выделение памяти для ссылочных типов. Операция new

Операция new имеет вид:

```
new ИмяКласса(ПараметрыКонструктора)
```

Она вызывает конструктор класса *ИмяКласса* и возвращает созданный объект. Например:

```
type Person = auto class
    name: string;
    age: integer;
end;
```

```
var p: Person; //p хранит значение nil, память под объект не выделена
p := new Person('Иванов',20); //конструктор выделяет память под объект
                               //и записывает ссылку на него в переменную p
```

3. Освобождение памяти. Сборщик мусора

Все ссылочные типы в .NET находятся под управлением так называемого *сборщика мусора*. После того как объект становится не нужным, ему следует присвоить `nil`.

При нехватке динамической памяти выполнение программы приостанавливается, и запускается специальная процедура, называемая сборкой мусора. Она определяет все так называемые достижимые объекты. Если на данный объект более никто не указывает, то он считается недостижимым и будет собран сборщиком мусора. Время вызова сборщика мусора считается неопределенным.

```
p := nil; //память, отведенная под p, после присваивания ей nil станет
          //недостижимой и будет собрана в непредсказуемый момент.
```