

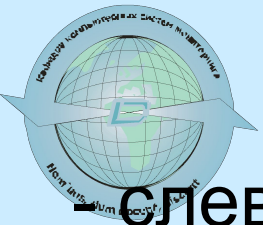
Министерство образования и науки Украины
Донецкий национальный технический университет
Кафедра компьютерных систем мониторинга



ГЕОИНФОРМАЦИОННЫЕ СИСТЕМЫ

Харитонов А. Ю.

Лекция 13. Синтаксис Avenue

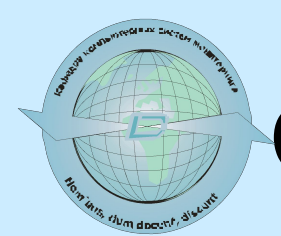


Порядок выполнения запросов-

слева направо. Заключенное в скобки выражение оценивается как единый элемент. Арифметические операторы также подчиняются правилу выполнения слева направо

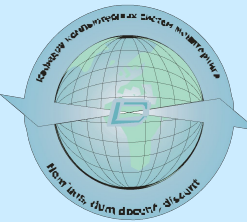
$$x = 3 + 5 * 6$$

$$x = 3 + (5 * 6)$$



Соглашения об именах запросов

Большинство имен запросов Avenue имеет формат `<action> <subject>` (действие, предмет или субъект), например `GetProject`, `FindScript`, `AddDoc`, и `SetValue`, где `Get`, `Find`, `Add` и `Set` являются действиями, а `Project`, `Script`, `Doc` и `Value` - субъектами. Действие идентифицирует роль и поведение запроса. Субъект разъясняет и определяет запрос. Общие ключевые слова действия именуют запросы через иерархию классов.



Соглашения об именах запросов

Функция **Add** добавляет объект в цель (target) запроса.

```
myList.Add( "Illinois" )
```

Функция **As** возвращает версию объекта, преобразованного в ИНОЙ ТИП.

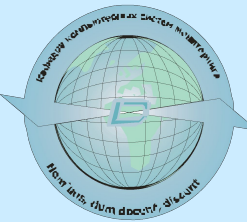
```
x = "56789".AsNumber
```

Функция **Can** возвращает утверждение true (истина) или false (ложь), если функция может быть выполнена.

```
File.CanDelete( "$MYDATA/states.dbf".AsFileName )
```

Функция **Find** возвращает ссылку для поименованного объекта.

```
myView = av.GetProject.FindDoc( "View1" )
```



Соглашения об именах запросов

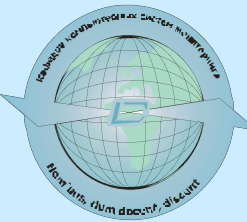
Функция **Get** возвращает ссылку на другой связанный объект. В отличие от **Get** функция **Return** выдает ссылку на новый объект.

```
themeList = myView.GetThemes
```

```
theProject = av.GetProject 'gets a reference to the active project.
```

Функция **Has** возвращает true, если объект включен или имеет указанное условие или состояние. Запрос **HasError** наиболее общий. Ниже запрос HasError определяет корректность созданного нового объекта текста программы (скрипта).

```
myScript = Script.Make( "av.minimize" )  
if (myScript.HasError) then  
    MsgBox.Error( "Script failed to make", "" )  
end
```



Соглашения об именах запросов

Функция **Is** возвращает true или false, указывая значение Логического (Boolean) параметра или состояния.

```
theFile = "$MYDATA/test".AsFileName  
if (theFile.IsDir) then
```

```
...
```

```
end
```

Функция **Make** создает новый экземпляр целевого класса.

```
aView = View.Make
```

Функция **Return** создает новый объект и возвращает ссылку на него.

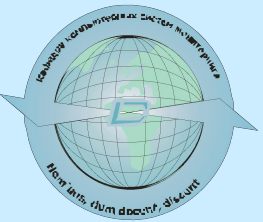
```
myView.ReturnExtent
```

```
aLine.ReturnCenter
```

Функция **Set** присваивает новое значение для объекта или свойства целевого объекта.

```
aDoc.SetGUI(aGUIName) 'Документу назначается GUI с именем  
aGUIName
```

```
aButton.SetIcon(anIcon) 'задается значок для кнопки
```



Оператор присваивания Assignment

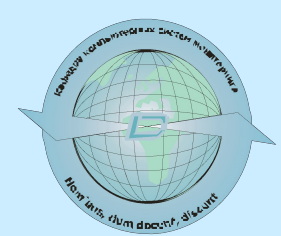
Оператор присваивания вызывает переменную для ссылки на связанный объект. Переменная создается при ее написании слева от оператора присваивания. Справа от оператора задается объект, на который производится ссылка.

Переменная (variable) = Объект (Object)

Имя переменной начинается с буквы или знака подчеркивания, содержит буквы и цифры и не зависит от регистра. Если имя переменной начинается со знака подчеркивания, то переменная является глобальной. Действие глобальной переменной распространяется на все скрипты проекта, в то время как действие локальной переменной ограничено пределами скрипта, которому она присвоена.

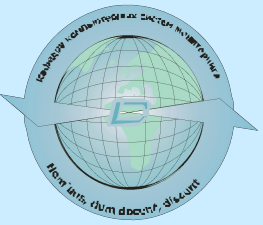
Имя переменной должно быть уникальным.

theView = av.ActiveDoc



Оператор If...Then...Else

```
if ( выражение1 ) then  
    блок запросов1  
elseif ( выражение2 ) then  
    блок запросов2  
[else]  
    блок запросов3  
end
```

Оператор For Each

Оператор For Each применяет набор запросов к элементам множества.

for each переменная in множество [by шаг]

блок запросов

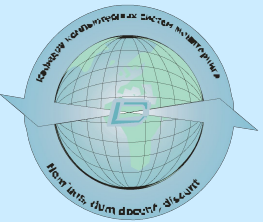
end

for each i in 1..100

MsgBox.Info(i.AsString, "")

end

Эта переменная является итерационной переменной. Она получает значение каждого элемента в множестве. При каждом выполнении цикла переменная получает следующее значение. В приведенном примере *i* является итерационной переменной и она последовательно получает значения 1, 2, 3, 4, ..., 99, 100, изменяя значения при каждом цикле процесса. Каждый цикл *for each* выполняется 100 раз.



Оператор While

Оператор **While** повторяет выполнение блока запросов, пока заданное выражение остается истинным (true).

Убедитесь, что цикл while обеспечивает условия завершения, иначе Вы получите бесконечный цикл.

while (выражение)

блок запросов

end

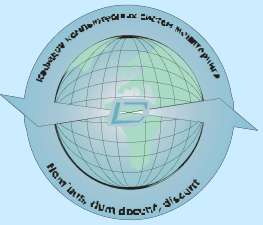
Блок запросов может выполняться один раз, много раз, или не выполняться совсем.

while (msgbox.yesno("Run the script again?","ArcView",TRUE))

av.Run("TestScript",nil)

End

Оператор **Break** прекращает выполнение ближайшего оператора *For Each* или *While*, в который он вложен.



Оператор Continue

Оператор Continue управляет переходом к следующей итерации цикла, в котором он встречается, с пропуском всех остальных операторов в этом цикле.

```
theView = av.GetActiveDoc  
for each t in theView.GetThemes  
  for each f in t.GetFtab.GetFields  
    if (f.IsTypeNumber) then  
      continue  
    end  
    f.SetVisible(true)  
  end  
t.GetFtab.Refresh  
End
```

В данном примере скрипт проверяет каждое поле каждой темы в Виде, делая все нечисловые поля скрытыми (невидимыми). Скрипт использует оператор Continue, чтобы перейти к следующему полю в списке полей, если текущее поле является числовым, тем самым пропуская запрос SetVisible.