

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



ГЕОИНФОРМАЦИОННЫЕ СИСТЕМЫ

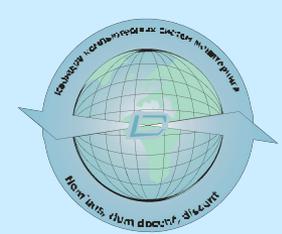
Харитонов А. Ю.

Лекция 12. Основы Avenue

Avenue -



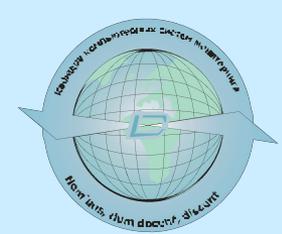
объектно-ориентированный язык программирования. Особенность Avenue, (как и всех объектно-ориентированных языков) - идентификация **объектов** а, затем, посылка к ним **запросов**. В отличие от процедурного программирования, в котором упор сделан на функции (процедуре), в Avenue вместо непосредственного вызова функций с аргументами посылается **запрос к объекту**. Когда объект получает этот запрос, он выполняет какое-то **действие**. Объекты ArcView являются элементами **иерархии классов**, которые организованы по функциональным категориям, связанным со всеми аспектами приложения.



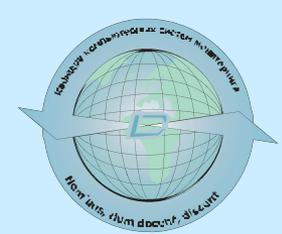
Объект -

элемент, который представляет что-либо, с чем Вы работаете в ArcView. Объекты:

- элементы интерфейса (элементы управления и окна документов),
- компоненты проекта (виды),
- базовые элементы Avenue, (числа, даты, логические выражения, строки, или графические примитивы типа точек, линий и т.д.).



Для каждого класса в **Иерархии классов** ArcView задаются характеристики, общие для всех объектов, которые являются экземплярами этого класса. Например, класс *Doc* имеет 6 подклассов: *Chart*, *Layout*, *Project*, *Script*, *Table* и *View*. Каждый из этих подклассов наследует все характеристики класса *Doc* и дополнительные атрибуты и запросы, присущие данному подклассу.



Запросы

- позволяют создавать, управлять или получать информацию об **объектах**. Для каждого класса Avenue имеются запросы, действующие с данным классом или элементами этого класса. Запросы инициируют **методы**, присущие **классу**.

Запрос к объекту в Avenue возвращает **другой** объект. В некоторых случаях эти объекты представляют собой информацию о других объектах. Например, оператор:

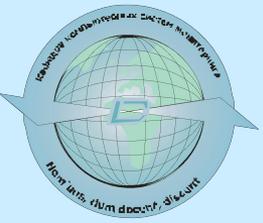
```
theName = theView.GetName
```

возвращает объект строки, являющийся именем объекта "theView". Переменная theName получает этот объект.

Аналогично, выражение:

```
myNumber = 2 + 5
```

возвращает числовой объект, который является результатом запроса на сложение двух числовых объектов. ArcView приписывает этот числовой объект переменной myNumber. Некоторые запросы приводят к возврату объекта NIL.



Запросы

сгруппированы в две категории:

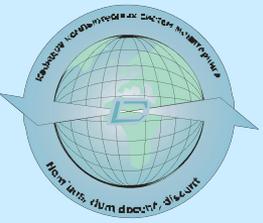
- созданные непосредственно для класса;
- созданные для объектов или экземпляров этого класса.

Запросы к классу не нацелены на отдельный объект, но обеспечивают механизм для более общих функциональных возможностей. В запросах к классу **фактическое имя класса** (например, View, LineFile, List) является **целью ("target")** запроса. Запросы к классу обычно создают экземпляры класса или **возвращают информацию** о воздействии на класс. *Пример:*

```
FileName.Make( "myfile.txt" )
```

возвращает новый экземпляр FileName.

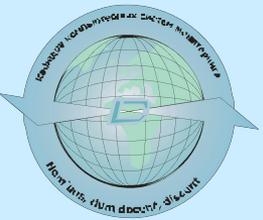
Запросы к экземпляру (instance requests)



- позволяют выполнять действие с объектом. Например, можно максимизировать окно показа Виду с именем *View1*, можно сделать тему под названием "Города" в Виде "Россия" активной, можно изменить цвет графики при выводе на экран. Запросы к экземпляру обычно выполняют некоторое действие или изменяют состояние объекта.

Различные классы поддерживают различные запросы. Например, можно послать запрос к Виду на его зуммирование, но числа не понимают запрос зуммирования.

В некоторых случаях, можно послать одинаковый запрос (такой как, *Save* (сохранить), *Open* (открыть), *+*, или *Make* (сделать)) к объектам, которые принадлежат разным классам. Эти запросы приводятся в действие по-разному, но приводят к одинаковому общему результату. Это так называемый **полиморфизм**. Например, запрос *Make* сделанный для класса *List* выполнит иную функцию, чем запрос *Make* сделанный для *TextFile*, но оба запроса *Make* сгенерируют новый экземпляр целевого класса.



Синтаксис запросов

Постфиксная форма (Postfix), где запрос используется после объекта.

- Синтаксис : *Object.Request* (объект.запрос)
- Пример: *av.GetProject*

Постфиксные запросы также могут быть объединены в виде цепочки:

- Синтаксис : *Object.Request.Request...Request*

Каждая пара “запрос-объект” (*object.request*) возвращает другой объект, а несколько пар “запрос-объект” выполняются по порядку, слева направо.

- Пример: запрос *theView.FindTheme("Roads")* возвращает объект темы. Запрос *SetVisible* делает объект темы невидимым.
- *theView.FindTheme("Roads").SetVisible(false)*

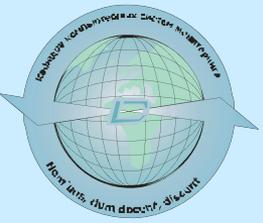
Инфиксная форма (Infix) для арифметических операторов.

- Синтаксис: *Object Request Object*
- Пример: *2 + 3*

Выражения выполняются слева направо. Avenue не поддерживает обычного приоритета при порядке выполнения операторов.

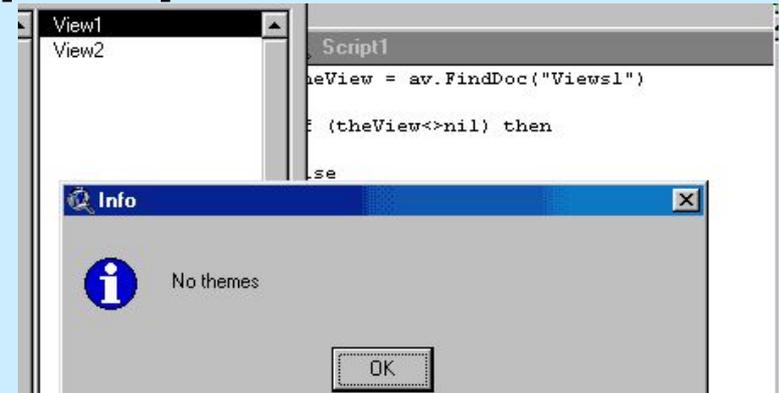
Префиксная форма (Prefix) - запрос стоит перед объектом.

- Синтаксис: *Request Object*
- Пример: *not True*



Первая программа

```
theView = av.FindDoc("View1")  
if (theView<>nil) then  
else  
    MsgBox.Info( "No themes", "")  
end
```



Av - зарезервированное слово, которое представляет в **ArcView** объект прикладной программы. Запуская **ArcView**, создается объект прикладной программы, который остается, пока работает **ArcView**. При записи сценария, **av** - отправная точка. Например, **av.GetProject** возвращает текущий проект или **av.GetActiveDoc** возвращает активный вид. Можно минимизировать или максимизировать окно прикладной программы, обращаясь к **av.Minimize** или **av.Maximize**. У этой этого объекта есть метод **FindDoc**. Этот метод ищет вид. Возвратит нуль, если не нашел.