

**НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ**

**«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»**

**ДЕПАРТАМЕНТ ПРИКЛАДНОЙ МАТЕМАТИКИ**

---

**Технология обработки данных с  
применением синтаксического  
анализа методом простого  
предшествования Н. Вирта**

**Автор: аспирант 3-го года обучения Азаров В.А.**

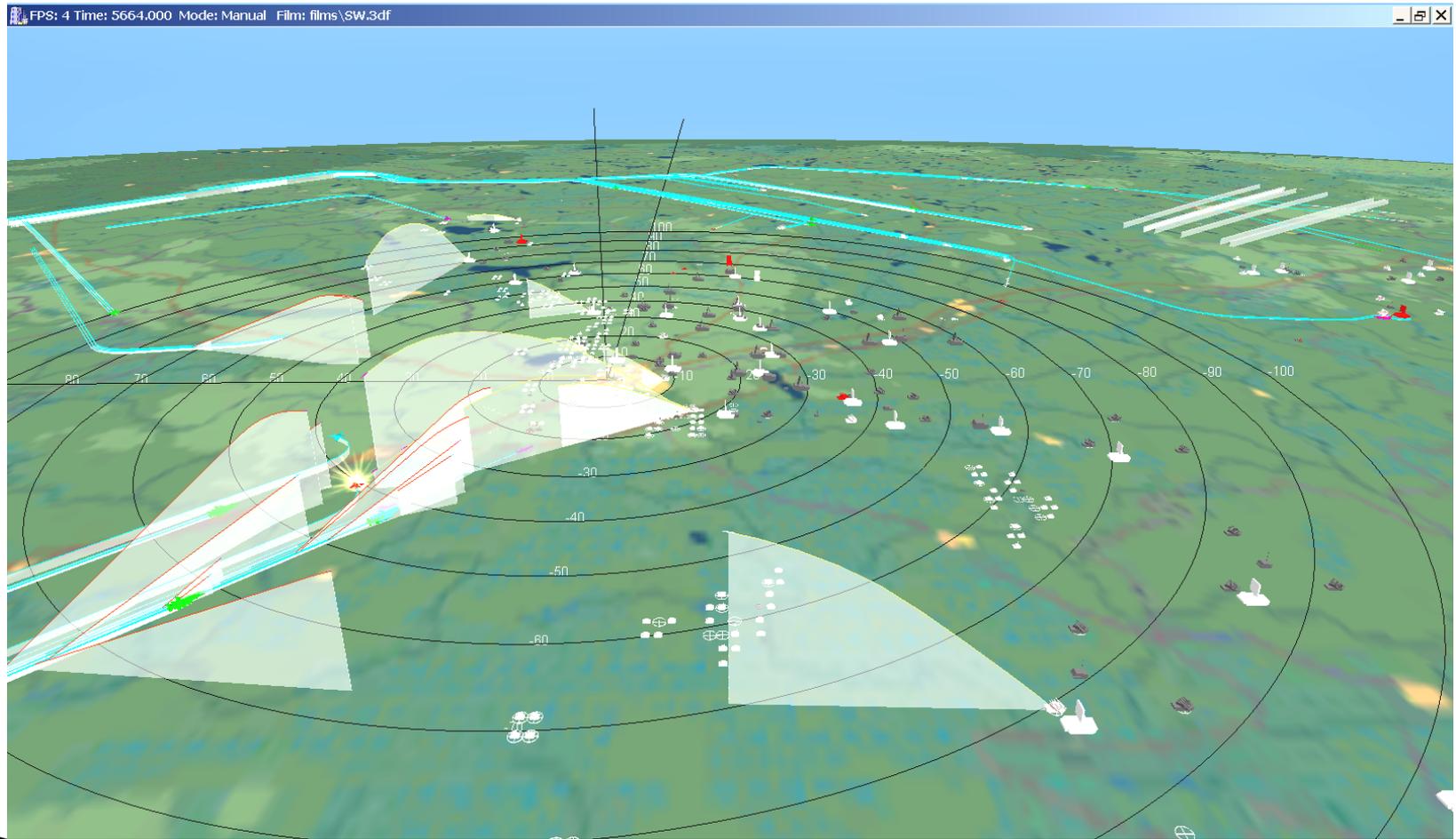
**Научный руководитель: к. т. н. Истратов А.Ю.**

**Москва 2015**

# Операционно-динамическое моделирование хода боевых действий

- ▶ Поиск и отработка групповой тактики боевого применения зенитной ракетной и авиационной техники возможны только на основе компьютерной имитации боевых действий
- ▶ Актуальны задачи оперативного сетецентрического управления действиями различными типами вооружений
- ▶ Требуется иметь средства комплексного компьютерного моделирования боевого пространства с имитацией действий взаимодействующих сторон

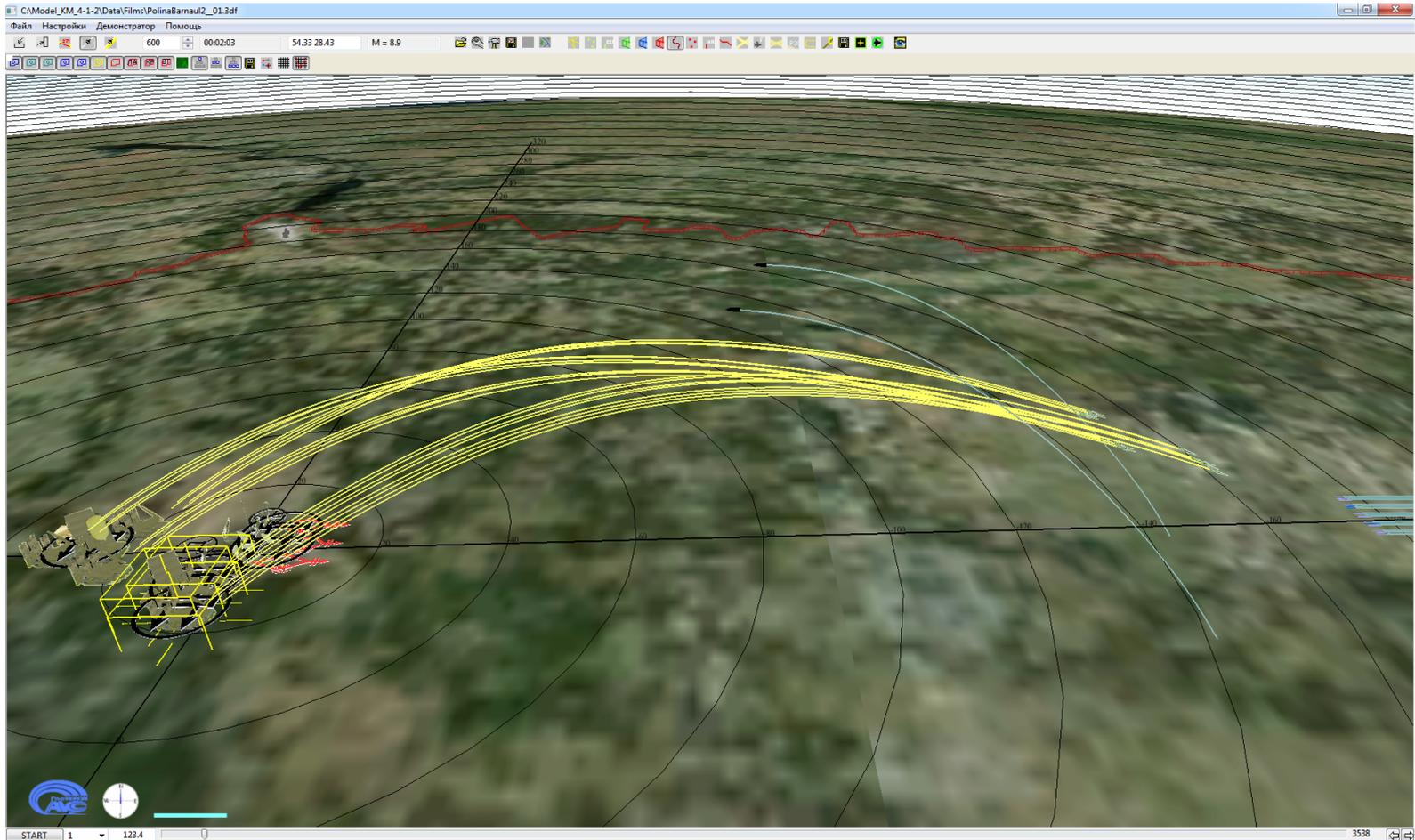
# Визуализация моделирования хода боевых действий



# Основные тенденции разработки математических моделей

- ▶ Повышенное внимание уделяется разработке универсальных моделирующих аппаратов, позволяющих воспроизводить работу различных технических средств на поле боя
- ▶ Существует практика объединения различных моделей для совместного решения общих задач
- ▶ Такие модели должны быть инвариантны к набору и составу входных данных, необходимых для проведения моделирования

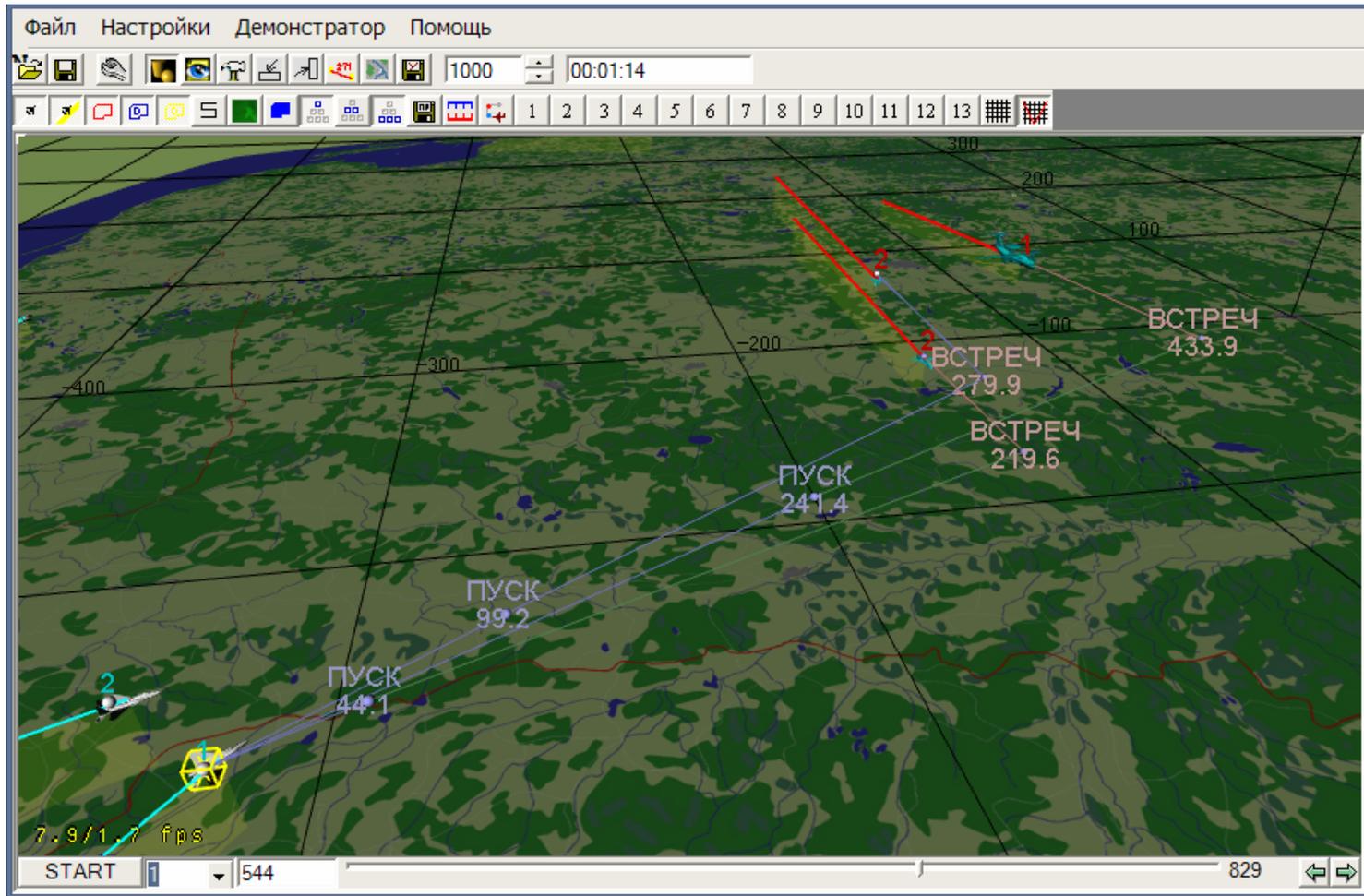
# Пример совместной работы различных моделей



# Планирование воздушных операций

- ▶ Авиация может применяться совместно со средствами ПВО
- ▶ Необходимо обеспечить безопасность авиационных комплексов в условиях ракетного и радиоэлектронного противодействия
- ▶ Требуется априорная информация о районе полета для составления различных вариантов полетных заданий

# Исследование вопросов безопасности проведения воздушных операций



# Характер входных данных для проведения исследований

Данные тактико-технических характеристик представляют состояние и технические особенности объектов моделирования

Данные сценариев описывают движение и поведение объектов

Конфигурационные параметры определяют различные режимы работы моделирующего аппарата

# Проблема выбора хранилища входных данных

- ▶ Современные СУБД предлагают зачастую избыточные функций для хранения и извлечения данных
- ▶ При переходе на каждую новую версию СУБД может потребоваться доработка моделирующего аппарата
- ▶ Требуется государственная сертификация при работе по оборонной тематике
- ▶ К входным данным не применяются операции записи

# Использование текстовой базы данных

- ▶ Позволяет представлять данные в удобном для пользователя формате
- ▶ Работа с данными возможна посредством текстового редактора
- ▶ Не требует финансовых затрат на приобретение лицензий

# Постановка задачи

Пусть есть набор входных данных, записанных в текстовый файл в заданном формате

Требуется разработать средство обработки данных, позволяющее задействовать эти данные в работе моделирующего аппарата

# Пример формата представления данных

- ▶ **Массив:**

    Sending\_Settings       = 2 200 0.2 1 3

- ▶ **Таблица:**

    [DNA=]  
          0.0               1.0E-2  
          4.0               1.0E-2  
    [/DNA]

- ▶ **Группа:**

    [Zone.1]  
          MinRange = 1000.0  
          [DNA=]  
              4.5            5.0E-3  
              12.0           5.0E-3  
              12.5           1.6E-4  
              360.0           1.6E-4  
          [/DNA]  
    [/Zone.1]

# Основные определения

**Символ** – это любой знак, который рассматривается как нечто неделимое с точки зрения смыслового контекста

**Алфавит** – это конечное непустое множество символов

**Цепочка (или слово)** – это конечная одномерная последовательность символов алфавита; слово, не содержащее ни одного символа называется пустым словом

Пусть  $A^*$  множество всех слов, включая пустое слово, состоящих из символов алфавита  $A$

**Формальный язык**  $L$  над алфавитом  $A$  – это подмножество множества  $A^*$ , для которого можно указать алгоритм перебора, порождения или распознавания принадлежности языку всех входящих в него слов

# Понятие грамматики языка

$$G = (T, N, P, S)$$

**T** – конечное множество терминальных (основных) СИМВОЛОВ

**N** – конечное множество нетерминальных (вспомогательных) символов

**P** – конечное множество правил вывода (продукций)

**S** – нетерминальный символ, называемый аксиомой ( $S \in N$ )

# Пример подстановки

Пусть имеется грамматика  $G_1=(T,N,P,S)$ , заданная следующим образом:

$T: a,b$

$N: S$

$P: S \rightarrow asb$

$S \rightarrow \varepsilon$

Пример порождения слова:

$S \rightarrow asb \rightarrow aaSbb \rightarrow aabb$

# Понятие контекстно-свободной и автоматной грамматики

Грамматика  $G$  называется **контекстно-свободной**, если все ее productions имеют вид:

$$A \rightarrow w, \text{ где } A \in N, w \in (T \cup N)^*$$

Грамматика  $G$  называется **автоматной**, если все ее productions имеют вид:

$$A \rightarrow aB, A \rightarrow a \text{ или } A \rightarrow \varepsilon,$$

$$\text{где } A, B \in N, a \in T$$

# Понятие конечного автомата

$$K = (A, S, S_F, P)$$

**A** – конечный, непустой алфавит автомата

**S** – конечное непустое множество состояний автомата

**S<sub>F</sub>** – подмножество **S**, множество финальных состояний

**P** – конечное непустое множество команд автомата

# Принцип работы конечного автомата Мили



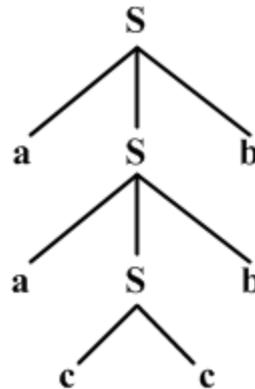
# Дерево вывода цепочки

Пусть имеется цепочка «аассбб» и грамматика вида

$G: S \rightarrow aSb$

$S \rightarrow cc$

Дерево вывод а в данной грамматике имеет вид:



# Понятие синтаксического анализа

Синтаксическим анализом слова  $x$  в грамматике  $G$  называется процедура построения дерева вывода слова  $x$  в грамматике  $G$

Согласно теореме Эрли для любой КС-грамматики может быть построен анализатор, осуществляющий обработку входной строки длины  $n$  за  $O(n^3)$  шагов

# Отношения простого предшествования

Между парой символов  $a, b$  в грамматике  $G$  определено отношение («**синтаксическое равенство**»), если в этой грамматике можно найти продукцию вида  $A \rightarrow uabv$ , где  $A \in N$ ,  $u, v \in (T \cup N)^*$

Между парой символов  $a, b$  в грамматике определено отношение  $a < b$  («**синтаксически меньше**»), если в этой грамматике можно найти продукцию вида  $A \rightarrow uaBv$ , где  $A, B \in N$ , где  $B = bw$ ,  $u, v, w \in (T \cup N)^*$

Между парой символов  $a, b$  в грамматике определено отношение  $a > b$  («**синтаксически больше**»), если в этой грамматике можно найти продукцию вида  $A \rightarrow uBbv$ , где  $A, B \in N$ , где  $B = wa$ ,  $u, v, w \in (T \cup N)^*$

Между парой символов  $a, b$  в грамматике определено отношение  $a \succ b$  («**синтаксически больше**»), если в этой грамматике можно найти продукцию вида  $A \rightarrow uBCv$ , где  $A, B, C \in N$ , где  $B = wa$ ,  $C = bz$ ,  $u, v, w, z \in (T \cup N)^*$

# Алгоритм Р.Флойда построения таблицы простого предшествования

- ▶ Для каждого нетерминала грамматики  $G$  составить множества левовыводимых  $L(A) = \{a : A \Rightarrow au, a \in T \cup N, u \in (T \cup N)^*\}$  и правовыводимых символов  $R(A) = \{a : A \Rightarrow ua, a \in T \cup N, u \in (T \cup N)^*\}$
- ▶ Рассмотреть все пары соседних символов из правых частей всех продукций грамматики, для каждой пары ввести отношение предшествования
- ▶ Объединить отношения в таблицу простого предшествования

# Алгоритм Н. Витрта синтаксического анализа

- ▶ Пусть дана грамматика  $G=(T,N,P,S)$ ,  $a,b \in T$
- ▶ Введем понятие пустого символа-ограничителя ( $\#$ ), который располагается слева и справа от входной цепочки
- ▶ Составим таблицу простого предшествования
- ▶ Положим, что левый ограничитель  $\# < c$ , а правый ограничитель  $\# > c$ , где  $c$  – любой символ
- ▶ Сконструируем команду переноса левого ограничителя в стек МП-автомата:

$$S_1\# \rightarrow \#S_2$$

- ▶ Для каждой пары символов, для которых определены отношения  $a=b$  или  $a < b$  сконструируем команду переноса в стек:

$$aS_2b \rightarrow abS_2$$

- ▶ Для каждой пары символов, для которых определено отношение  $a > b$  или сконструируем команду перехода в состояние свертки:

$$aS_2b \rightarrow aS_3b$$

- ▶ Для каждой продукции вида  $A \rightarrow u$  запишем команду свертки

$$uS_3 \rightarrow AS_2$$

- ▶ Сконструируем команду перехода в финальное состояние, если в стеке остается только аксиома:

$$\#SS_2 \rightarrow S_4$$

$$S_4 \rightarrow$$

# Представление грамматики языка в виде БНФ

- ▶  $\langle \text{программа} \rangle ::= \langle \text{команда} \rangle$
- ▶  $\langle \text{программа} \rangle ::= \langle \text{программа} \rangle \langle \text{команда} \rangle$
  
- ▶  $\langle \text{команда} \rangle ::= \text{имя} = \langle \text{множ\_знач} \rangle$
- ▶  $\langle \text{команда} \rangle ::= \text{груп\_скобка} \langle \text{программа} \rangle \text{прав\_скобка}$
- ▶  $\langle \text{команда} \rangle ::= \text{табл\_скобка} \langle \text{множ\_знач} \rangle \text{прав\_скобка}$
- ▶
- ▶  $\langle \text{множ\_знач} \rangle ::= \text{набор значений}$
  
- ▶ имя - последовательность букв и цифр, начинающаяся с буквы
- ▶ набор значений - число, массив чисел, строка, массив строк
- ▶ табл\_скобка - конструкция вида [имя=]
- ▶ груп\_скобка - конструкция вида [имя]
- ▶ прав\_скобка - конструкция вида [/имя]

# Полученная грамматика для описания формата представления данных

- ▶  $S \rightarrow A$
- ▶  $S \rightarrow SA$
  
- ▶  $A \rightarrow i=B$
- ▶  $A \rightarrow [S)$
- ▶  $A \rightarrow \{B)$
  
- ▶  $B \rightarrow v$
  
- ▶  $i$  - последовательность букв и цифр, начинающаяся с буквы
- ▶  $v$  - набор значений (число, массив чисел, строка, массив строк)
- ▶  $[$  - конструкция вида [имя]
- ▶  $\{$  - конструкция вида [имя=]
- ▶  $)$  - конструкция вида [/имя]

# Выводы

- ▶ Описана проблема хранения и использования входных данных при решении задач компьютерного моделирования
- ▶ Описан формат представления данных в текстовом файле, перечислены его особенности
- ▶ Предложена технология обработки данных с применением метода синтаксического анализа Н. Вирта