

**АНАЛИТИК-2007**

---

**Abstract:** In the paper the system of computer algebra ANALITIC-2007 and its source language which is the newest version of languages of family ANALITIC developed in the Institute of Problems of Mathematical Machines and Systems NAS of Ukraine is described. Works on languages of family ANALITIC have been started under the leadership of academician Glushkov V.M. in 60th years. In total 8 versions of the language have been realized. The basic differences of on languages of this family are the advanced means of dialogue and a high level of the artificial intellect. From the previous versions of languages of family ANALITIC-2007 differs more deep study and classification of means of computer algebra, the further development of control facilities by calculations, the device of the decision of problems by interactive methods and means of recognition of structural and functional properties of objects of computer algebra. Language differs laconicism and generality of objects and functions. ANALITIC-2007 it is intended for the decision of scientific and applied problems, and also for intensification of training disciplines with the raised level of mathematical modelling.

**Key words:** numerical-analytical transformations, recognition of properties of objects, management of symbolical transformations, an automatic mode, a dialogue mode.

**Анотація:** У роботі описані система комп'ютерної алгебри АНАЛІТИК-2007 і її вхідна мова, яка є новою версією мов сімейства АНАЛІТИК, що розробляються в Інституті проблем математичних машин і систем НАН України. Робота над мовами сімейства АНАЛІТИК розпочалася під керівництвом академіка Глушкова В.М. в 60-х роках. Усього було реалізовано 8 версій мови. Основними відмінностями мов цього сімейства є розвинені засоби діалогу та високий рівень штучного інтелекту. Від попередніх версій мов сімейства АНАЛІТИК-2007 відрізняється більш глибоким дослідженням і класифікацією засобів комп'ютерної алгебри, подальшим розвитком засобів керування обчисленнями, апарату рішення задач інтерактивними методами й засобів розпізнавання структурних і функціональних властивостей об'єктів комп'ютерної алгебри. Мова відрізняється лаконічністю й спільністю об'єктів і функцій. АНАЛІТИК-2007 призначений для рішення наукових і прикладних задач, а також для інтенсифікації навчання дисциплінам з підвищеним рівнем математичного моделювання.

**Ключові слова:** чисельно-аналітичні перетворення, розпізнавання властивостей об'єктів, керування символічними перетвореннями, автоматичний режим, діалоговий режим.

**Аннотация:** В работе описаны система компьютерной алгебры АНАЛИТИК-2007 и ее входной язык, который является новейшей версией языков семейства АНАЛИТИК, разрабатываемых в Институте проблем математических машин и систем НАН Украины. Работы над языками семейства АНАЛИТИК были начаты под руководством академика Глушкова В.М. в 60-х годах. Всего было реализовано 8 версий языка. Основными отличиями языков этого семейства являются развитие средства диалога и высокий уровень искусственного интеллекта. От предыдущих версий языков семейства АНАЛИТИК-2007 отличается более глубоким исследованием и классификацией средств компьютерной алгебры, дальнейшим развитием средств управления вычислениями, аппарата решения задач интерактивными методами и средств распознавания структурных и функциональных свойств объектов компьютерной алгебры. Язык отличается лаконичностью и общностью объектов и функций. АНАЛИТИК-2007 предназначен для решения научных и прикладных задач, а также для интенсификации обучения дисциплинам с повышенным уровнем математического моделирования.

**Ключевые слова:** численно-аналитические преобразования, распознавание свойств объектов, управление символными преобразованиями, автоматический режим, диалоговый режим.

## 1. Введение

Данная статья развивает направление работ по созданию языков и систем компьютерной алгебры семейства АНАЛИТИК, начало которому было положено в 60-х годах прошлого столетия академиком В.М. Глушковым. Первые разработки этого направления, аппаратно реализованные на ЭВМ серии МИР, были ориентированы на решение инженерных и исследовательских задач. Совокупность требований к этим ЭВМ (за исключением габаритов) совпадала с основными свойствами современных ПЭВМ, что дает основание многим ученым в области компьютерной техники считать их прообразами ПЭВМ [1–8].

Интерпретация языка высокого уровня резко повысила качество аппарата интерактивного режима решения задач и позволила создать эффективный аппарат автоматизации формульных выкладок с развитым уровнем интеллекта.

На языках семейства АНАЛИТИК численно-аналитическими методами решается большое количество разнообразных по тематике научных и прикладных задач.

Основным объектом систем компьютерной алгебры в настоящее время является выражение, подобное выражению математического анализа. Однако более глубокое исследование задач и алгоритмов привело к тому, что сейчас объектом языка АНАЛИТИК является некоторое обобщение языка математического анализа символами операций таких, как векторная запятая «,», текстовая кавычка «'» и подстановка значения «[» и «]» [12].

Операндами функций и операторов систем компьютерной алгебры обычно являются именуемые выражения, которые содержат переменные. Эти переменные в свою очередь могут именовать подобные выражения. Именуемые выражения других функций и операторов содержат такие же переменные. Таким образом, множество выражений, которые подаются на вход пользовательской программы или появляются на каком-либо этапе выполнения (промежуточные данные), образуют целостные иерархические структуры. Отличительной особенностью АНАЛИТИКа-2007 является то, что подобная целостная структура стала основным объектом языка. Каждый операнд рассматривается как именуемое выражение такой структуры.

Другой особенностью АНАЛИТИКа-2007 является высокий уровень его искусственного интеллекта. Это выражается в том, что, наряду с развитым аппаратом интерактивного решения задач, он радикально ориентирован на решение задач в автоматическом режиме. Программы рассматриваются как системы блоков преобразователей и распознавателей. Причем распознаватели стоят в узлах ветвления программ и определяют направление дальнейших преобразований. Распознаватели программы строятся из специальных процедур (базовых распознавателей), образующих систему распознавания АНАЛИТИКа-2007.

Если задача решается в интерактивном режиме, роль распознавателя играет, в основном, человек-оператор, и хотя работа программных распознавателей часто требует больше машинного времени, чем работа преобразователей, общее время решения задачи в автоматическом режиме несопоставимо меньше времени решения в интерактивном режиме, к тому же не требуя затрат времени высококвалифицированных операторов, в совершенстве знающих задачу [11].

Нетрадиционная классификация процедур языка с разделением функций и операторов на классы базовых распознавателей и базовых преобразователей определяется высоким уровнем искусственного интеллекта и его значением для автоматического выполнения программ [9, 13–16].

### **Общие сведения и назначения**

A-2007 устанавливается на ПЭВМ и работает как приложение ОС Windows-95 и выше.

A-2007 состоит из ядра и пакетов расширения. Компактное ядро образует множество функций, обеспечивающих семантическую общность языка и универсальность функциональных свойств системы и поддерживающих работу A-2007 в среде Windows, осуществляющих трансляцию и ретрансляцию программы и данных, выполнение программы, преобразование

объектов языка системы (в т.ч. и программы, которая также является объектом языка). Система автоматически определяет доступный для нее объем памяти компьютера и по умолчанию использует максимальную часть доступного ОЗУ. Пользователю предоставлена возможность заказывать объем памяти, необходимый для нормальной работы его программ. В случае превышения заказанной пользователем памяти объема ОЗУ используется предоставляемая Windows виртуальная память. Максимальный объем памяти, который может быть использован системой, ограничен только размерами дискового пространства, отведенного для виртуальной памяти в ОС Windows.

Проблемная ориентация достигается за счет пакетов расширения. Эти пакеты объединяют программы или процедуры, написанные на входном языке системы А-2007, предназначенные для решения определенного класса задач. При желании пакет включается в систему, а имена программ и процедур, входящих в пакет, автоматически становятся символами языка. Такое состояние системы сохраняется до соответствующей санкции на исключение пакета [8].

### **Алгоритмический язык АНАЛИТИК-2007**

При описании конструкций языка А-2007 в данной статье соблюдаются следующие соглашения:

- определяемая и определяющая конструкции разделяются символом  $\Leftrightarrow$ , который следует понимать как «ЕСТЬ ПО ОПРЕДЕЛЕНИЮ»;
- символ  $\rightarrow$  следует понимать как «ВОЗВРАЩАЕТСЯ РЕЗУЛЬТАТ ПРИМЕНЕНИЯ ПРОЦЕДУРЫ К ОБЪЕКТУ»;
- необязательные элементы или группы элементов формата заключаются в фигурные скобки;
- альтернативные элементы формата разделяются союзом «или»;
- список элементов формата, повторяющихся несколько раз, обозначается одним элементом, за которым следует многоточие.

В А-2007 для нижеописанных операций определен следующий приоритет операций: !, [, ], ', ?, #, <\*>, \*\*, \*, / (деление), +, -, <, <=, =, ^=, >, >=, ^, &, \, , (агрегация). Операции перечислены в порядке убывания приоритета. Операции одного уровня (умножение и деление, сложение и вычитание, отношения) выполняются слева направо. Для изменения порядка выполнения операций используются круглые скобки.

Для элементов формата используются мнемонические обозначения объектов системы: И – имя, Б – буква, Ц – цифра, Т – текст, ЦЧ – целое число, РЧ – рациональное число, Ч – число, П – переменная, В – выражение, О – оператор. В случае, если формат содержит несколько однотипных элементов, они снабжаются индексами.

### **2. Режимы работы АНАЛИТИК-2007**

Фактически система А-2007 имеет два режима работы: интерактивный и автоматический. Язык обладает развитым набором средств, направленным на распознавание структурных и функциональных свойств объектов. Эти средства позволили ориентировать язык на решение задач в автоматическом режиме. Под средствами в языке А-2007 понимаются функции и операторы

языка.

Богатый набор средств, который реализует все теоретические возможности интерпретации программ на языке программирования высокого уровня, позволяет эффективно решать задачи и проводить отладку программ в интерактивном режиме.

Автоматический режим работы программы осуществляется в тех случаях, когда весь сеанс работы и получение результата происходит без обращения к нижеописанным функциям «Прервать» и «Продолжить».

В начале сеанса работы после запуска А-2007 на экране появляется главное меню. Всего в главном меню 9 пунктов (рис. 1): Программы, Данные, Выполнить, Прервать, Продолжить, Запросы, Директивы, Сохранение, Редактор. Каждому из пунктов соответствует некоторое подменю.

Программы	Данные	Выполнить	Прервать	Продолжить	Запросы	Директивы	Сохранение	Редактор
-----------	--------	-----------	----------	------------	---------	-----------	------------	----------

Рис. 1. Главное меню

При обращении к соответствующим пунктам меню или подменю выполняются перечисленные ниже функции.

### 2.1. Программы

Подменю «Программы» содержит имена всех программ библиотеки программ. Над выбранной программой могут выполняться действия, предусмотренные другими функциями (такими, как «Выполнить», «Переименовать» и др.). Программа может содержать данные либо данные могут вводиться до или после выбора программы с помощью подменю «Данные». Подменю содержит элемент с именем «Новая». Он используется в процессе создания и включения в библиотеку новых программ. Выполнение программы начинается после выбора пункта главного меню «Выполнить».

### 2.2. Данные

В подменю каждый комплект данных обозначен одним именем (если данные образуют некоторое множество выражений, их можно оформлять в виде именованного вектора).

Структура данных может быть произвольной в зависимости от входного блока вызываемой программы (однако рекомендуется вводить данные в виде программы, которая имеет вид  $I : V$ ; где  $I$  – имя,  $V$  – выражение).

С помощью курсора можно выбрать имя данных, которые после выбора подаются на вход выбранной программы либо сохраняются с помощью функции «Сохранить» или «Переименовать».

### 2.3. Выполнить

Иницируется на выполнение последняя программа, к которой обращались во время работы. Предварительно прерывается выполнение программы, которая выполнялась в момент нажатия клавиши.

### 2.4. Прервать

Прерывается выполнение программы после выполнения очередного оператора. Программа

выводится в окно «Результаты», после чего с помощью функции «Продолжить» ее можно продолжать без повторного счета. После прерывания можно проанализировать информацию о состоянии программы и внести необходимые изменения в любую именованную часть программы или данных.

## **2.5. Продолжить**

Функция обеспечивает выполнение программы после прерывания, анализа состояния программы или данных и выполненных изменений.

## **2.6. Запросы**

Интерактивный режим выполнения программы в основном состоит из чередования “запросов” пользователя о состоянии программы или данных в текущий момент выполнения и “директив” пользователя об изменении промежуточных данных или программы или порядка ее выполнения. И запросы, и директивы являются программами на языке A-2007. Их сложность не ограничена. Те типичные запросы и директивы, которые предполагается употреблять многократно, сохраняются в специальных библиотеках и могут вызываться по мере надобности.

Функция прерывает выполнение программы. Подменю «Запросы» содержит имена соответствующих программ. Выбранная программа выводится в окно и выполняется с помощью функции «Выполнить».

Подменю содержит элемент с именем «Новый». Он используется в процессе создания новых запросов, которые могут выполняться непосредственно при создании без сохранения в библиотеке.

## **2.7. Директивы**

Функция прерывает выполнение программы. Подменю «Директивы» содержит имена соответствующих программ. Выбранная программа выводится в окно и выполняется с помощью функции «Выполнить».

Подменю содержит элемент с именем «Новая». Он используется в процессе создания новых директив, которые могут выполняться непосредственно при создании без сохранения в библиотеке.

## **2.8. Сохранение**

Подменю «Сохранение» содержит функции:

- «Сохранить»;
- «Переименовать»;
- «Удалить».

Функция «Сохранить» выполняет сохранение объекта: программы, данных, запроса или директивы. По умолчанию сохраняется прежнее имя объекта.

Функция «Переименовать» позволяет сохранить последний выбранный объект под новым именем в указанной библиотеке. После нажатия на «Переименовать» в нижней части дисплея

появляется поле для ввода нового имени объекта. После нажатия «Сохранить» объект запоминается с новым именем в соответствующей библиотеке.

С помощью функции «Удалить» можно удалить объект из библиотеки.

## 2.9. Редактор

Функции «Редактора» относятся к последнему окну, к которому обращались во время работы.

Пункт меню «Редактор» содержит функции:

- «Найти»;
- «Заменить»;
- «Шрифт (вид, размер, цвет)»;
- «Таблица символов»;
- «Распределение памяти».

Функция «Найти» отмечает указанный объект в помещенном в окне тексте.

Функция «Заменить» выполняет замену в помещенном в окне тексте.

Функция «Шрифт» меняет вид, размер, цвет шрифта в выделенном тексте.

Функция «Таблица символов» позволяет использовать для написания программ основные математические операции ( $\Sigma$ ,  $\Pi$ ,  $\partial$ ,  $d$ ,  $\neq$ ,  $\geq$ ,  $\leq$ ,  $\sqrt{\quad}$ ) и набор констант ( $\pi$ ,  $i$ ,  $e$ ).

Функция «Распределение памяти» дает возможность заказывать объем памяти, необходимый для нормальной работы программ.

## 3. Алфавит

Алфавит языка образует множество стандартных знаков, которое может быть расширено по желанию пользователя. Алфавит состоит из следующих групп знаков:

- прописные и заглавные буквы (русские, украинские, латинские, греческие и т.д.) и пробел;
- арабские цифры;
- знаки операций;
- специальные знаки.

При создании программ могут быть использованы практически любые символы, предоставляемые TTF-шрифтами, установленными в ОС Windows. В базовый набор символов включены основные математические операции ( $\Sigma$ ,  $\Pi$ ,  $\partial$ ,  $d$ ,  $\neq$ ,  $\geq$ ,  $\leq$ ,  $\sqrt{\quad}$ ), набор констант ( $\pi$ ,  $i$ ,  $e$ ).

## 4. Объекты языка А-2007

Символы и знаки языка записываются с помощью знаков алфавита.

Основным объектом языка, с помощью которого описываются данные задачи, является выражение (любая синтаксически правильная последовательность символов и знаков языка). Выражения строятся из имен, букв, цифр, текстов, чисел, переменных, операций, символов функций, координат, аналитических выражений, предикатов, векторов, форм. Синтаксические правила записи выражений в соответствии с их семантикой описаны ниже в соответствующих разделах.

Объектами языка также являются программа и части программы, так как особенности

реализации позволили определить на этих объектах ряд операторов и функций языка А-2007.

Кроме того, А-2007 поддерживает минимальный, но обладающий свойствами полноты, аппарат для работы с такими объектами как тексты, комплексные числа и программы.

Объекты системы А-2007 имеют имена, с помощью которых осуществляется обращение к объекту. Такими именами обычно являются именуемые выражения. Имена базовых средств языка носят мнемонический характер, в программе из соображения удобства используются их сокращения, содержащие такое количество первых знаков имени, которое исключает неоднозначность.

Входные данные задач КА обычно имеют вид выражений с символьными переменными, написанных на общепринятом в науке и технике языке. Они образуют систему выражений, связанных значениями переменных. Входящие в состав выражения переменные могут именовать подобные объекты, образуя, таким образом, **многоуровневую иерархическую структуру – МИС** (в частности, если все переменные не имеют значений, то выражение является одноуровневой МИС). Именуемое выражение, стоящее в начале такой структуры, именуется всю МИС. Так как операции и функции языка определены на таких объектах, то МИС является основным объектом языка А-2007.

#### 4.1. Числа

Числа определяются следующими форматами:

- целые без знака      ЦБ  $\Leftrightarrow$  Ц<sub>1</sub>{Ц<sub>2</sub>}... ;
- целые                    ЦЧ  $\Leftrightarrow$  {+ или – }ЦБ ;
- рациональные        РЧ  $\Leftrightarrow$  {+ или –}ЦБ<sub>1</sub>{/ЦБ<sub>2</sub>} ;
- десятичные            ДЧ  $\Leftrightarrow$  {+ или –}{ЦБ<sub>1</sub>}.ЦБ<sub>2</sub> или {+ или –}{ЦБ<sub>3</sub>}{.ЦБ<sub>4</sub>}ЕЦЧ.

В формате десятичного числа ЦБ<sub>1</sub>, ЦБ<sub>2</sub>, ЦБ<sub>3</sub>, ЦБ<sub>4</sub> задают мантиссу, а ЦЧ – порядок. Разрядность (число цифр) не ограничена.

#### 4.2. Константы

В языке используются следующие константы, представленные в десятичном формате: **PI** – число  $\pi$ , **EE** – число  $e$ . А также **II** – мнимая единица  $\sqrt{-1}$ .

#### 4.3. Переменные

Переменные описываются в программе в виде идентификаторов и могут именовать любой объект системы.

Пример.

**z := x\*\*2 + y\*\*2;**

В этом фрагменте *z* именуется выражение, *x* – значения не имеет (именует сам себя).

#### 4.4. Аналитические выражения

К классу аналитических выражений относятся выражения, которые образуются из чисел, переменных, символов операций, указателей функций, координат и скобок. При этом понятие

выражение в общепринятом в науке и технике языке расширено за счет включения в него символов операций «,» (векторная запятая), «'» (текстовая кавычка), «[ ]» (подстановка значений) и функций языка А-2007 таких, как ПРИМЕНИТЬ, ФОРМИРОВАТЬ, ПОДСТАВИТЬ и т.п. Синтаксис языка А-2007 не требует обычной для языков компьютерной алгебры лексикографической и степенной упорядоченности выражений.

Например, допускаются конструкции вида

$$A := x + x^{**}2 + \sin(x^{**}2 + y) + x + y + \sin(y + x).$$

В правой части не приведены подобные, не упорядочены степени и отсутствует лексикографическое упорядочение.

#### 4.5. Предикаты

Предикатами в языке АНАЛИТИК могут быть любые выражения, если их значениями являются 1 «истина» или 0 «ложь». Если они содержат знаки логических операций «и» или «или», то они образуют конъюнкцию или дизъюнкцию.

Пример.

$$x := 0 \setminus \text{tg}(0); \rightarrow 0$$

#### 4.6. Тексты

Текстами являются любые последовательности символов между ограничителями текста «'».

#### 4.7. Векторы

Вектор определяется форматом  $(B_1, B_2, \dots, B_n)$ . Компонентами  $B_i$  могут быть произвольные выражения языка. Используя векторы АНАЛИТИКА-2007, можно моделировать различные агрегатные объекты: векторы, матрицы, списки, перечислимые множества и т.п.

#### 4.8. Функции

Функция (пользователя) имеет вид

$$И(\Pi_1 \{ \Pi_2 \} \dots) : B_1$$

или

$$И(\Pi_1 \{ \Pi_2 \} \dots) :: B_1,$$

или

$$И(\Pi_1 \{ \Pi_2 \} \dots) := B_1,$$

где  $И$  – имя функции,  $\Pi_i$  – аргументы функции (свободные переменные),  $B_1$  – тело функции (выражение, которое может содержать свободные переменные, числа и символьные параметры).

#### 4.9. Программа

Так как ряд операторов и функций языка А-2007 определены над программой или ее частями, то программа и ее части являются объектами языка. Программа состоит из операторов, разделенных



точкой с запятой, последовательности операторов, заключенной в круглые скобки (составных операторов), описаний процедур-функций и подпрограмм.

## 5. Базовые средства

Базовыми средствами языка А-2007 будем называть операции, математические функции, функции языка А-2007, операторы, функции для свернутого представления суммы, произведения и агрегации, операции символьного дифференцирования и интегрирования, средства для модификации работы программы, средства, связывающие программу с системой, разделители.

Операнды описанных ниже функций языка А-2007 обычно содержат переменные, и если по контексту программы нет других указаний, то соответствующие функции оперируют с объектами, которые являются результатом подстановки значений этих переменных на один шаг. Если информации в указателе функции недостаточно для ее выполнения, например, какой-нибудь вообще числовой индекс не является числом, то результатом выполнения функции считается указатель функции. Синтаксически правильно записанные операторы, если они не могут быть выполнены из-за некорректно заданных операндов, воспринимаются как пустые операторы.

Базовые средства СКА АНАЛИТИК делятся на шесть типов. Такое деление условно, так как некоторые функции и операторы можно одновременно отнести к нескольким из перечисленных ниже типам:

- средства именованя;
- средства управления;
- средства преобразования;
- средства распознавания свойств объектов;
- средства связи с системой;
- разделители.

### 5.1. Средства именованя

Объекты системы А-2007 и их структурные части имеют имена, которыми являются именующие выражения ( $B$ ): идентификаторы, координаты, метки, номера. Объект становится значением  $ЗН(B)$  имени  $B$  в результате преобразований с учетом заданных модификаций (режимов, см. пп. 5.2) соответствующих средств преобразований.

Некоторые функции и операторы, которые в основном относятся к другим из вышеперечисленных типам и в результате работы которых образуются объекты с новыми именами, также можно считать средствами именованя. Они описаны в соответствующих пунктах: функция ПРИНАДЛЕЖНОСТЬ (пп. 5.4.4), операторы присваивания (пп. 5.3.3.5), оператор оптимизации вычислений (пп. 5.2.1.11), оператор ПАМЯТЬ (пп. 5.5.1), оператор ВВОД (пп. 5.5.7), процедура-функция (пп. 5.2.2.15), подпрограмма (пп. 5.2.2.16).

#### 5.1.1. Идентификаторы

Идентификатор – это произвольная последовательность букв и цифр, начинающаяся с буквы. Идентификатор может именовать любой объект.

## 5.1.2. Именуемые выражения

### 5.1.2.1. Выражение

Выражение может именовать иерархическую структуру, т.е. множество выражений, упорядоченных относительно операции подстановки, и использоваться как ссылка для доступа к именуемому выражению.

Пример.

z: x+y;

x: cos(tau)-2;

y: sin(tau)+1;

tau: a\*t + b;

Выражение  $x + y$  именуется МИС, образованную приведенными выражениями.

### 5.1.2.2. Координаты

Вычисляемые объекты СКА графически представляются в виде древовидного графа, вершинам которого соответствуют структурные части объекта (МИС). Так как на любом древовидном графе траектория от “корня” до любой вершины определяется однозначно, то древовидный граф может рассматриваться как система координат. Любую вершину можно рассматривать как корень соответствующего подвыражения, а траектория от корня древовидного графа определяет положение подвыражения в указанной системе координат, т.е. задает координаты этого подвыражения. Таким образом, любое подвыражение может именоваться именем объекта и своими координатами. Так как в А-2007 сигнатура операций включает операцию подстановка “[”, то МИС рассматривается как некоторое обобщение понятия “выражения” и координаты заданы относительно корня такого выражения.

В А-2007 координаты записываются в виде

$$И[\Gamma_1][\Gamma_2] \dots [\Gamma_n],$$

где  $И$  – выражение, именуемое объектом,  $\Gamma_i$  – указатели, которые обеспечивают доступ к структурным частям объекта.

Указатели  $\Gamma_i$  могут иметь вид

$$\Gamma_i \Leftrightarrow B_{1i}, B_{2i}, \dots, B_{mi},$$

при этом  $B_{ji}$  может иметь вид:

- ЦБ – номер операнда соответствующей главной операции;
- ЦБ<sub>1</sub> : ЦБ<sub>2</sub> – обеспечивается доступ к последовательности операндов соответствующей главной операции с номерами от ЦБ<sub>1</sub> до ЦБ<sub>2</sub>;
- ЦБ<sub>1</sub>; ЦБ<sub>2</sub>; ... – при этом обеспечивается доступ к последовательности операндов соответствующей главной операции с перечисленными номерами.

Если указатель группы вида \*, то он выделяет все операнды главной операции выражения.

Пример.

z: x+ sin(x)+ y<sup>p</sup>+ t<sup>3</sup>;

$p:\cos(x)+y;$

$u:=\text{KOOP}(z,\cos(x)); \rightarrow z[3,2][1].$

По форме координаты являются обобщением распространенной конструкции «индексная переменная». Для векторов и матриц он позволяет выбирать не только отдельные элементы или ряды, но и произвольные их комбинации.

Если в процессе решения задачи какой-либо объект именуется  $B[i]$  и главная операция выражения  $B$  имеет  $n$  операндов, где  $n < i$ , то выражение  $B$  соответствующим образом достраивается.

Пример.

$z:=(a,b);$

$z[4]:=5; \rightarrow z = (a,b, ,5).$

Именованые структурных частей МИС в АНАЛИТИКе с помощью координат позволяет выполнять преобразования над ними без предварительной подстановки.

Операндам главной  $n$ -местной операции выражения  $B$  (пп. 5.4.4) ставятся в соответствие имена вида  $B[i]$ , где  $i \in (1,2,\dots,n)$  – номер операнда или координата, где каждый операнд также рассматривается как многоместная операция. Его операнды именуется аналогичным образом.

Пример.

В выражении

$z : \sin(x) + y**u + v * n**2 + a**b**c;$

структурные части именуется:

$z[1] \rightarrow \sin(x)$

$z[2, 2] \rightarrow u$

$z[3, 2, 2] \rightarrow 2$

$z[4, 1] \rightarrow a.$

Таким образом, последовательность индексов  $\Gamma \Leftrightarrow i, j, \dots, m$  изоморфна последовательности главных операций, выполняемых при вычислении значения выражения  $B$  после присвоения значения объекту с именем

$$B[i, j, \dots, k, m].$$

Здесь  $m \in (1, 2, \dots, n_{ij\dots k})$ ;  $n_{ij\dots k} \geq 1$  – арность отношения, операндом которого является объект с указанным именем.

Для именованые структурных частей выражения  $B$  как иерархической структуры используется выражение вида

$$B[i_1, j_1, \dots, k_1, m_1][i_2, j_2, \dots, k_2, m_2] \dots [i_p, j_p, \dots, k_p, m_p].$$

Здесь последовательность кортежей изоморфна последовательности объектов, входящих в иерархию  $B$ , и которые в своей иерархической структуре содержат объект с указанным именем, а  $p$  – количество иерархических уровней в объекте  $B$ .

Пример.

$as: x**2+y**2;$

```
x: cos(tau)-2;  
y: sin(tau)+1;  
tau: = b + a*t ;  
ВЫВ(:as[1,1][1,1][2,1])      →      a
```

### 5.1.3. Имена операторов

#### 5.1.3.1. Номера

Каждому оператору программы, подпрограммы, процедуры или составного оператора присваивается имя в виде порядкового номера относительно начала соответствующей последовательности операторов. Поскольку на перечисленных объектах определены некоторые функции и операторы языка, такое именование обеспечивает доступ к соответствующим операторам.

#### 5.1.3.2. Метки

Любой оператор может быть именован меткой. Метка – это идентификатор. Для доступа к оператору могут использоваться и относительные номера, которые отсчитываются от оператора, именованного меткой.

```
Пример.  
HA(Lab+2);  
q:=5;  
Lab. s:=6;  
a:=3;  
ВЫВ(:s);
```

Оператор **HA(Lab+3)** передает управление второму после оператора с меткой **Lab** оператору, т.е. **ВЫВ(:s)**.

#### 5.1.4. Имя программы

Программа по умолчанию получает имя в виде зарезервированной переменной ВП (выполняемая программа). К ее частям можно обращаться с помощью координат, так как каждый оператор именуется своим номером относительно начала программы.

Процедуры и подпрограммы именуются идентификаторами (пп. 5.2.2.15 и 5.2.2.16).

#### 5.1.5. Именованное классов

В программах на языке А-2007 объявление типов объектов не является обязательным, однако в языке определяются следующие классы<sup>1</sup> объектов, являющиеся аналогами типов:

**ЦИФРА**, **БУКВА**, **ОПЕРАЦИЯ**, **ТЕКСТОВОЕ ВЫРАЖЕНИЕ**, **ЧИСЛО**, **ДЕСЯТИЧНОЕ ЧИСЛО**, **ЦЕЛОЕ ЧИСЛО**, **РАЦИОНАЛЬНОЕ ЧИСЛО**, **ПЕРЕМЕННАЯ**, **ФУНКЦИЯ**, **МАТЕМАТИЧЕСКАЯ ФУНКЦИЯ**, **ПОЛЬЗОВАТЕЛЬСКАЯ ФУНКЦИЯ**, **БАЗОВАЯ ФУНКЦИЯ**, **АНАЛИТИЧЕСКОЕ ВЫРАЖЕНИЕ**, **ЛОГИЧЕСКОЕ ВЫРАЖЕНИЕ**, **КООРДИНАТЫ**, **ВЕКТОРНОЕ ВЫРАЖЕНИЕ**.

<sup>1</sup> Жирным шрифтом здесь выделены обозначения соответствующих классов.

Перечисленные классы входят в состав наиболее общего класса, **ВЫРАЖЕНИЕ ЯЗЫКА А-2007**.

В программах используются следующие сокращения:

ЦИФ, БУК, ОПЕР, ТЕКСТ, ЧИС, ДЕС, ЦЕЛ, РАЦ, ПЕРЕМ, ФУНК, МФ, ПФ, БФ, АНАЛ, ЛОГ, КООР, ВЕКТ, ВЯЗ.

Пример.

s:= (x,2,y);

z:=s?ВЕКТ; → 1

z:=s?ТЕКСТ; → 0.

#### **5.1.6. Именованние специальных (системных) переменных**

Специальные (системные) переменные обозначаются зарезервированными идентификаторами и именуют соответственно следующие объекты: РАЗР – разрядность нормализованных мантисс десятичных чисел; АВОСТ – стандартную метку – имя программы обработки аварийной ситуации; АВОСТ1 – классификационный номер аварийной ситуации; АВОСТ2 – номер оператора программы, при выполнении которого возникла аварийная ситуация; ВП – выполняемая программа.

#### **5.2. Модификации (режимы) работы программы и ее базовых средств**

Особенностью аппарата управления в АНАЛИТИКе является то, что модификации выполнения программы, функций и операций определяются не типом данных, а специальными управляющими (режимными) функциями, операндами функций и операторами (глобально-режимными операторами, в пределах программы или нескольких операторов, до очередного изменения режима или локально – режимными функциями, в пределах объекта или его части). Приоритетом обладает локальное задание режимов. Режимные операторы и режимные функции обычно записываются с помощью подобных синтаксических структур и различаются только в контексте программ.

Логическая структура программ в АНАЛИТИКе формируется как с помощью ряда обычных для других языков операторов, так и с помощью ряда перечисленных ниже специфических для АНАЛИТИКа операторов, в основном связанных с возможностью динамического изменения программы в ходе ее выполнения.

Для статистически устойчивых групп, одновременно применяемых режимных функций и операций, в языке А-2007 существуют описанные ниже специальные символы, определяющие наборы модификаций операторов или функций для всей программы или ее частей.

Некоторые функции и операторы, которые в основном относятся к другим, из вышеперечисленных, группам и в результате работы которых осуществляется управление вычислениями или выполнением программы в А-2007, также являются средствами управления. Они описаны в соответствующих пунктах:

ПОДСТАВИТЬ (пп. 5.3.3.2), операторы присваивания (пп. 5.3.3.5), КЛАСС (пп. 5.4.3), ПРОГРАММА (пп. 5.2.2.7), оператор ПРИ (пп. 5.3.3.4), оператор ВВОД (пп. 5.5.7), оператор ВЫВОД (пп. 5.5.8).

### 5.2.1. Функции и операторы, управляющие преобразованиями данных

#### 5.2.1.1. Оператор РАЗРЯДНОСТЬ

При вычислениях десятичных чисел разрядность (число цифр) определяется оператором РАЗР, который имеет вид

$$РАЗР := B,$$

где  $ЗН(B)$  задает разрядность.

По умолчанию до выполнения первого оператора, задающего разрядность, она полагается равной 6 цифрам.

#### 5.2.1.2. Функция РАЦИОНАЛЬНЫЙ режим вычислений

Имеет вид

$$РАЦ(B\{B_1\}),$$

где  $B$  – вычисляемое выражение,  $ЗН(B_1)$  задает разрядность.

Если все операнды выражения  $B$  целые числа, вычисления производятся в алгебре рациональных чисел, результат также получается в виде целого числа неограниченной длины.

Если  $B$  содержит десятичные операнды, вычисления ведутся в алгебре десятичных чисел с разрядностью  $ЗН(B_1)$ . Действие функции локально.

#### 5.2.1.3. Функция ДЕСЯТИЧНЫЙ режим вычислений

Функция имеет вид

$$ДЕС(B_1, B),$$

где  $B$  – вычисляемое выражение,  $ЗН(B_1)$  задает разрядность.

Обеспечивает вычисление в алгебре десятичных чисел с разрядностью  $B_1$ . Действие функции локально.

#### 5.2.1.4. Оператор РАЦИОНАЛЬНЫЙ режим вычислений

Оператор РАЦ задает алгебру рациональных чисел для выражений с рациональными операндами. Для десятичных чисел сохраняется заданная в программе разрядность. Действие оператора распространяется на всю программу или ее часть.

Оператор операцию деления целых чисел заменяет на сокращение дроби и запрещает вычисление трансцендентных функций, аргументами которых являются рациональные числа.

Пример.

РАЦ;

$$x := 4/6; \quad \rightarrow \quad 2/3$$

$$b := ДЕС(10, x-РАЦ(\sin(x))/(x+1)); \quad \rightarrow \quad .666666666E0 - .599999999E0*\sin(2/3).$$

### 5.2.1.5. Оператор ДЕСЯТИЧНЫЙ режим вычислений

Оператор ДЕС задает алгебры десятичных чисел. Все рациональные числа превращаются в десятичные с разрядностью, которую задает оператор РАЗР.

### 5.2.1.6. Функции НЕКОММУТАТИВНЫЙ и КОММУТАТИВНЫЙ режимы вычислений

Функции имеют вид

$$НЕКОМ(B)$$

и

$$КОМ(B),$$

где  $B$  – вычисляемое выражение.

Функции, соответственно, отменяют и восстанавливают свойство коммутативности операций. Для коммутативных многоместных операций  $+$ ,  $*$ ,  $\&$ ,  $\setminus$  режим коммутативности может отменяться локально, если они входят в состав МИС.

Пример.

КФ(3);

$$b:=2*y+x+y+3*x; \quad \rightarrow \quad 3*y + 4*x$$

$$b:= НЕКОМ(x+y-5*x); \quad \rightarrow \quad x+y+-5*x.$$

### 5.2.1.7. Операторы НЕКОММУТАТИВНЫЙ и КОММУТАТИВНЫЙ режимы вычислений

Операторы имеют вид

$$НЕКОМ$$

и

$$КОМ ,$$

отменяют и задают свойство коммутативности операций глобально – в масштабе программы или ее части, действуют до очередного изменения режима.

### 5.2.1.8. Функция приведения к канонической форме

Нижеприведенные статистически устойчивые наборы преобразований в А-2007 называются каноническими формами и именуются номерами 0, 1, ..., 7. Функции и операторы А-2007 могут иметь операнды вида

$$(B | B_1),$$

где  $B$  – вычисляемое выражение,  $ЗН(B_1)$  – имя канонической формы. В соответствии с общими правилами А-2007, если нет других указаний, переменные выражения  $B$  подставляются на один шаг. Выражение  $B$  вычисляется циклически, пока не реализуются все модификации, заданные набором  $ЗН(B_1)$ . Подобные наборы могут разрабатываться пользователем и также становиться значением  $B_1$ . В соответствии с общим правилом действие функций локально и распространено только на выражение  $B$ .

Наборы преобразований приведения к каноническим формам пересекаются, т.е. одни и те

же элементарные преобразования могут входить в различные наборы. В частности, наборы от 0 до 5 связаны отношением вложенности, т.е. все процедуры  $i$ -го набора входят в состав  $n$ -го набора, если  $i < n$ .

Составы наборов преобразований, применяемых к объекту, следующие.

К **нулевой канонической форме** относятся преобразования, связанные с изменением структуры объекта. Она обеспечивает подстановку значений переменных и ликвидацию «лишних» скобок (не изменяющих порядок вычислений для многоместных операций).

Пример.

$z: a+b+(c+u);$

$R:=(z | 0); \rightarrow a+b+c+u.$

**Первая каноническая форма** выполняет упрощения выражений с учетом свойств нуля и единицы в математическом анализе и описывается соотношениями:  $B+0=B, 0+B=B, B-0=B, 0-B=(-1)*B, B*1=B, 1*B=B, B*0=0, 0*B=0, B**1=B, B**0=1, 1**B=1, 0**B=0, 0**0=ABOCT, B/1=B, 0/B=0, B/0=ABOCT, B\&0=0, 0\&B=0, B\&1=B, 1\&B=B, ^0=1, ^1=0, B\1=1, 1\B=1, B\0=B, 0\B=B, \sin(0)=0, \cos(0)=1, \tg(0)=0, \ctg(\pi/2)=0, \ln(1)=0, \lg(1)=0, \exp(0)=1.$

Во **второй канонической форме** выполняются все операции над числами и вычисляются все стандартные функции с числовыми аргументами. Преобразования второй канонической формы выполняются в АНАЛИТИКЕ по умолчанию.

Кроме того:

– выполняются преобразования:  $\text{dif}(\pi**0, B)=B, \text{dif}(\pi, \text{Ч})=0, \wedge B=B, B_1-B_2=B_1+(-1)*B_2, B_1/B_2=B_1*B_2**(-1);$

– выполняются преобразования, связанные со свойствами мнимой единицы:  $\Pi*\Pi=-1, \Pi**\text{ЦБ}=(-1)**\text{int}(2*\text{frc}(\text{ЦБ}/4))*\Pi**(2*\text{frc}(\text{ЦБ}/2)), \sqrt{\text{Ч}}=\Pi*\sqrt{\text{abs}(\text{Ч})}, \ln(\text{Ч})=\ln(\text{abs}(\text{Ч}))+\Pi*\Pi, \lg(\text{Ч})=\lg(\text{abs}(\text{Ч}))+\Pi*\Pi,$  где  $\text{Ч}$  – отрицательное число;

– вычисляются операции сложения и скалярного произведения для векторов (для векторов с нечисловыми компонентами арифметические операции не выполняются);

– раскрываются свернутые представления сумм, произведений и агрегатов с числовыми аргументами;

– выполняется функция, вычисляющая определенные интегралы по формуле Симпсона.

Пример.

РАЦ;

$a : 0.5;$

$b : 3/7;$

$c : t;$

$t:5;$

$z:=(\sin(a)+\cos(b)+\tg(c) | 2); \rightarrow .479425E0+\cos(3/7)+\tg(t).$

$\cos(3/7)$  не вычисляется, так как в программе задан рациональный режим.

**Третья каноническая форма** включает выполнение всех преобразований 2-ой канонической формы, выполняет преобразования над символьными объектами: производит приведение подобных членов и объединение степеней с общим основанием.



Кроме того:

с учетом коммутативности алгебраических операций +, \* и логических операций &, \ выполняются упрощающие преобразования вида

$B \& B = B$ ,  $B \setminus B = B$ ,  $\chi_1 * B + \chi_2 * B = \chi_3 * B$ , где  $\chi_3 = \chi_1 + \chi_2$ ,  $B ** B_1 * B ** B_2 = B ** (B_1 + B_2)$ ,  $B ** B_1 ** B_2 = B ** (B_1 * B_2)$ ,  
 $\ln(B_1 ** B_2) = B_2 * \ln(B_1)$ ,  $\lg(B_1 ** B_2) = B_2 * \lg(B_1)$ ,  $\exp(B) = E E ** B$ ,  $\text{sqrt}(B) = B ** (1/2)$ ,  
 $\text{dif}(\Pi ** B_1, \Pi ** B_2, B) = \text{dif}(\Pi ** (B_1 + B_2), B)$ .

Пример.

a :  $5 * x ** 7$ ;

b :  $x ** 4 * x ** 3$ ;

c : t;

t :  $2 * x ** 7$ ;

z := ( a+b+c | 3);  $\rightarrow 6 * x ** 7 + t$

KФ(3);

z1 := a+b+c ;  $\rightarrow 8 * x ** 7$ .

**Четвертая каноническая форма** включает выполнение всех преобразований 3-ей канонической формы, реализует дистрибутивные законы алгебраических и логических операций, выполняет умножение векторов на скаляр, раскрывает бином Ньютона.

Пример.

b := (2\*(x+y)+3\*x+7|4);  $\rightarrow 7+5*x+2*y$ .

**Пятая каноническая форма** включает выполнение всех преобразований 4-ой канонической формы, производится раскрытие сумм и произведений, формируются по заданному закону векторы.

Пример.

z := (agr (i, 0, 5, 1, x\*\*i) | 5);  $\rightarrow (1, x, x ** 2, x ** 3, x ** 4, x ** 5)$ .

**Шестая каноническая форма** включает выполнение всех преобразований 2-ой канонической формы и реализует аналитическое дифференцирование подвыражений, стоящих под знаками производная (dif) и частная производная (DIF). В случае, если выражение не продифференцировано полностью, то после выполнения преобразований в выражении сохраняются обозначения производных.

Пример.

a:  $x ** 2$ ;

z:  $x ** 3 + \sin(a) + a$ ;

d1 := (dif(x,z)|6);  $\rightarrow 3 * x ** 2 + \cos(a) * \text{dif}(x, a) + \text{dif}(x, a)$

d2 := (DIF(x, z)|6);  $\rightarrow 3 * x ** 2$ .

**Седьмая каноническая форма** включает выполнение всех преобразований 2-ой канонической формы, и реализуется процедура аналитического интегрирования выражений, которая точными аналитическими методами определяет первообразную, и при наличии пределов интегрирования вычисляет определенный интеграл по формуле Лейбница. Эти средства обеспечивают нахождение первообразной для элементарных табличных интегралов, а также для интегралов, подынтегральной функцией которых являются квазиполиномы со слагаемыми вида

$$x^{**n} \exp(a^*x + b)^* F(c^*x + d),$$

где  $n$  – целое число,  $a$ ,  $b$ ,  $c$  и  $d$  – не зависят от  $x$ ,  $F$  – целая тригонометрическая, гиперболическая или логарифмическая функция.

Пример.

$$b := (\ln(x, x^{**e^{**x}}) | 7); \quad \rightarrow \quad e^{**x^*(x-1)}.$$

Важнейшим обобщением аппарата канонических форм является проблемная каноническая форма. Эта каноническая форма конструируется пользователем при написании конкретной программы, когда необходимо быстрое или частое применение группы преобразований, не совпадающей с перечисленными выше группами. Такая группа преобразований задается вектором, имя которого указывается при обращении к соответствующей проблемной канонической форме. Каждому соотношению группы соответствуют четыре компоненты вектора: левая и правая части соотношения, модификация алгоритма применения соотношения и ограничения на значения свободных переменных (пп. 5.4.4). Если правая часть в соотношении отсутствует, то соответствующая компонента вектора остается пустой, а выражение, которое соответствует левой части, вычисляется. Преобразования с помощью такой канонической формы, благодаря специальным приемам реализации, значительно эффективнее циклического применения заданной группы формул.

#### 5.2.1.9. Оператор КАНОНИЧЕСКАЯ ФОРМА

Оператор имеет вид

$$K\Phi(B),$$

где  $B$  – имя соответствующей канонической формы.

Оператор действует на все преобразования программы или ее части до отмены режима.

В отличие от соответствующей функции оператор не блокирует подстановки значений переменных, которые выполняются в соответствии с контекстом программы.

Пример.

$K\Phi(3);$

$x:b;$

$p:x;$

$$z := 2^*p + 3^*p; \quad \rightarrow \quad 5^*b.$$

#### 5.2.1.10. Функция ветвления

Функция имеет вид

$$ЕСЛИ(B_1 \{ \alpha \}, B_2, B_3),$$

где  $B_1$  – предикат. Функция возвращает  $B_2$ , если  $B_1 = 1$  (истина), или  $B_3$ , если  $B_1 = 0$  (ложь),  $\alpha$  – задает множество допустимых при вычислении преобразований (имя канонической формы).

Пример.

A:  $x^*(x+1);$

$B: x^{**2} + x;$

$Y1:=ЕСЛИ(A=B, 1+5, 3+7); \quad \rightarrow \quad 10$

$Y2:=ЕСЛИ((A=B|4), 1+5, 3+7); \quad \rightarrow \quad 6.$

Для  $Y1$  вычисление предиката происходит по умолчанию при  $\alpha = 2$ ,  $A$  не равно  $B$  в силу того, что скобки не раскрываются. Для  $Y2 A = B | 4$  даст положительный результат, и мы пойдем по другой ветке.

#### 5.2.1.11. Оператор и функция оптимизации вычислений

Оператор может иметь вид:

1. ОПТ(1).

Обеспечивает вычисление значения МИС без повторного вычисления значений переменных.

2. ОПТ(2).

В процессе вычисления выполняет процедуру «склеивания» подобных подвыражений объекта и вычисляется только по одному разу.

3. ОПТ(B, И).

Также производит «склеивание» подобных подвыражений объекта  $B$  и полученную в результате МИС сохраняет с именем  $I$ .

### 5.2.2. Операторы, управляющие порядком выполнения программы

#### 5.2.2.1. Оператор переопределения операций и функций

Оператор имеет вид

$$ОПЕР(\otimes (П_1 \{ П_2, П_3, \dots \}), \{ \oplus (П_1 \{ П_2, П_3, \dots \}) \}),$$

где  $\otimes$  и  $\oplus$  – символы операций или функций, описанные в языке.

Оператор устанавливает режим выполнения программы, при котором операция (или функция)  $\otimes$  выполняется как операция (или функция)  $\oplus$ . Если второй операнд в операторе отсутствует, операция (или функция)  $\otimes$  в дальнейшем не выполняется. Действие оператора ОПЕР распространяется на всю последовательность операторов, выполняемых после него в программе.

Пример.

РАЦ;

КФ(4);

ОПЕР(sin(a), a - a\*\*3/6);

$b := (x - \sin(x)) / x^{**2}; \quad \rightarrow \quad 1/6 * x.$

#### 5.2.2.2. Оператор безусловного перехода

Оператор имеет вид

$$HA(B_1 + B_2),$$

где  $ZH(B_1)$  или  $ZH(B_2)$  являются либо меткой, либо целым числом (в частности, нуль). Это связано с тем, что операторы программы именуются своими порядковыми номерами и метки имеют

целочисленные значения (как переменные). Таким образом, заданный оператором переход может осуществляться относительно указанной метки.

Пример.

i:0;

i:=i+2;

HA(i); → HA(2) - управление передается на 2-ой оператор программы

или

HA(Lab+3); → управление передается на 3-ий оператор после оператора с именем Lab

Lab. a:2;

ВЫВ(: a);

i:0;

a:22; → оператор с именем Lab+3

ВЫВ(: a); → 22.

### 5.2.2.3. Оператор ветвления

Оператор имеет вид

$$\text{ЕСЛИ } (B, \{O_1\}, \{O_2\}).$$

Если  $3H(B) = 1$  (истина), выполняется оператор  $O_1$  (или пустой оператор, при отсутствии  $O_1$ ).

Если  $3H(B) = 0$  (ложь), выполняется оператор  $O_2$  (или пустой оператор, при отсутствии  $O_2$ ).

После этих преобразований выполняется следующий за ЕСЛИ оператор, если операторами  $O_1$  или  $O_2$  не предусмотрено другое продолжение программы. Если  $3H(B)$  не 0 и не 1, выполняется следующий за ЕСЛИ оператор.

Пример.

A:=3;

ЕСЛИ(A>100,ВЫВ(:'Истина'),ВЫВ(:'Ложь'))\$ → Ложь.

### 5.2.2.4. Операторы цикла

В А-2007 определены три вида оператора цикла: ЦИКЛ, ПОКА и ПОВТ.

#### 5.2.2.4.1. ЦИКЛ

Оператор имеет вид

$$\text{ЦИКЛ}(П, B_1, B_2, \{B_3\}, O).$$

Здесь последовательность  $П = (a, a + h, a + 2h, \dots, a + N \cdot h)$ ,

где  $a = \text{ent}(3H(B_1))$  – начальное значение,  $h = \text{ent}(3H(B_3))$  – шаг,

$$N = \text{ent}\left(\left\lceil \frac{\text{ent}(3H(B_2)) - a}{h} \right\rceil\right) - \text{условие выхода.}$$

Если  $B_3 = 1$ , то  $B_3$  можно опустить.

Оператор  $O$  выполняется для каждого значения  $P$ . Этим оператором может быть произвольный оператор, в том числе и оператор цикла.

#### 5.2.2.4.2. ПОКА

Оператор имеет вид

$$ПОКА(B, O).$$

Пока  $ЗН(B) = 1$  (истина), повторяется оператор  $O$  (тело цикла).

#### 5.2.2.4.3. ПОВТОРИТЬ

Оператор имеет вид

$$ПОВТ(B, O).$$

Сначала выполняется оператор  $O$ . После этого вычисляется  $ЗН(B)$ . Оператор  $O$  повторяется, пока  $ЗН(B) = 0$  (ложь).

#### 5.2.2.5. ПРЕРЫВАНИЕ

Оператор прерывания имеет вид

$$ПРЕП\{(B \text{ или } *)\}.$$

Оператор осуществляет прерывание работы  $i = ЗН(B)$  циклов, в которых он находится, и передает управление следующему за  $i$ -тым ( $i > 0$ ) циклом оператору, либо на начало  $i$ -го цикла ( $i < 0$ ) (отсчет ведется от оператора ПРЕП вверх). Если  $B = 1$ ,  $B$  можно опустить. Если  $B = *$  или  $i = 0$ , то осуществляется выход из всех циклов, внутри которых находится оператор ПРЕП.

#### 5.2.2.6. ПРОДОЛЖИТЬ

Оператор имеет вид

$$ПРОД\{(B \text{ или } *)\}.$$

Оператор, находясь в теле составных операторов, прерывает их работу. Если  $i = ЗН(B) > 0$ , управление передается на конец  $i$ -го составного, а если  $i < 0$  – на начало  $i$ -го составного (отсчет ведется от оператора ПРОД вверх). Если  $B = 1$ ,  $B$  можно опустить. Если  $B = *$  или  $i = 0$ , то осуществляется выход из всех составных, внутри которых находится оператор ПРОД.

#### 5.2.2.7. ПРОГРАММА

Оператор имеет вид

$$ПРОГ(T).$$

Работа оператора заключается в выполнении программы языка А-2007, заданной с помощью текста  $T$  (происходит трансляция). Если  $B$  не является текстом программы, то происходит аварийный останов.

Пример.

```
a:=b:=10;c:=20;ВЫВ(:b, ' ', c);  
ПРОГ(a);      →      10' 20.
```

#### 5.2.2.8. СТЕРЕТЬ

Оператор имеет вид

$$\text{СТЕРЕТЬ}(B),$$

где  $B$  – имя МИС или части МИС, программы или части программы.

Оператор позволяет стереть объект или фрагмент объекта с именем  $ЗН(B)$ , если оставшаяся часть объекта – синтаксически правильная конструкция. В противном случае происходит диагностируемый АВОСТ.

#### 5.2.2.9. ДОПОЛНИТЬ

Оператор имеет вид

$$\text{ДОП}(B_1 \text{ ! } O \text{ или } B)$$

и обеспечивает выполнение дополнительного оператора при достижении оператора с именем  $B_1$ . Дополнительный оператор указывается или непосредственно оператором  $O$ , или является оператором с именем  $B$ .

Пример.

```
ДОП(m ! ВЫВ(:'Дополнительный оператор'));  
e:=1/100;  
ЕСЛИ(3/6>e, m. , c:=0 );
```

В результате выполнения этого фрагмента программы выводится надпись 'Дополнительный оператор'.

#### 5.2.2.10. ЗАМЕНИТЬ

Оператор имеет вид

$$\text{ЗАМ}(B, O),$$

где  $B$  – имя заменяемого оператора,  $O$  – заменяющий оператор.

Объектом применения оператора является программа. Он позволяет производить замену (или вставку) любых операторов программы.

Пример.

```
x:=3;  
ЗАМ(3, (x:=12; r:=3));  
x:=5;  
OUTPUT(:x);  →      12.
```

#### 5.2.2.11. ТРАССА

Оператор имеет вид

$$ТРАСС\{\{I\}\}.$$

Оператор устанавливает режим, при котором процесс выполнения программы сопровождается выводом в файл с логическим именем  $ЗН(I)$  исходного текста каждого выполняемого оператора. Выводу трассы в файл с логическим именем  $ЗН(I)$  должен предшествовать оператор открытия файла. Если логическое имя опущено, то трасса выводится на дисплей.

#### 5.2.2.12. ЗНАЧЕНИЕ

Оператор имеет вид

$$ЗНАЧ\{\{I\}\}.$$

Оператор устанавливает режим вывода результата работы каждого оператора в файл с логическим именем  $ЗН(I)$ . Если логическое имя опущено, то значение выводится на дисплей.

#### 5.2.2.13. ОСТАНОВКА

Оператор имеет вид

$$ОСТ,$$

приостанавливает выполнение программы после каждого оператора. Выполнение следующего оператора происходит после нажатия клавиши «Enter».

#### 5.2.2.14. ВЫПОЛНИТЬ

Оператор имеет вид

$$ВЫП(B_1\{B_2\dots\}),$$

где  $B_n$  – имя оператора. Именованье операторов номерами дает возможность с помощью оператора ВЫПОЛНИТЬ рассматривать любое множество операторов как подпрограмму с глобальными параметрами. Зная номера операторов из любой части программы можно инициировать выполнение такой «подпрограммы» над полученными в данный момент данными.

Пример.

ВЫП (1, 10, 3, 5);

Оператор инициирует выполнение операторов с порядковыми номерами 1, 10, 3, 5 из любой точки программы с уже «наработанными» данными.

#### 5.2.2.15. Процедура-функция

Процедура-функция имеет вид

$$И(П_1\{П_2\}..):O,$$

где  $И$  – идентификатор,  $П_1, П_2, \dots$  – имена переменных,  $O$  – составной оператор, который реализует процедуру. Значение переменной, вычисленное процедурой-функцией, возвращается

операторами ВОЗВ (раздел 5.6.4.8), количество которых и положение в теле функции может быть любым. Указатель процедуры-функции синтаксически не отличается от указателя пользовательской функции и может быть операндом любого выражения.

Описание процедуры-функции может находиться в любом месте программы. В процедуре-функции все имена, не являющиеся именами других процедур, локализованы.

Пример.

$R(x):(m:=5;d:=m*x;ВОЗВ(d));$

$x0 : -5;$

$Y:=R(x0)+10;$

$ВЫВ(:Y); \quad \rightarrow \quad -15.$

#### 5.2.2.16. Подпрограмма

Подпрограммой является составной оператор, именованный следующим образом:

$$@I(P_1\{P_2\}...):O.$$

Здесь @ – признак процедуры,  $I$  – идентификатор,  $P_1, P_2, \dots$  имена переменных,  $O$  – составной оператор (тело процедуры).

В процедуре все имена, не являющиеся именами других процедур (описанных вне тела данной подпрограммы), локализованы.

Выполнение процедуры с именем  $I$  организуется оператором  $@I(B_1\{B_2\}...)$ .

Выход из процедуры осуществляется либо при достижении конца тела процедуры, либо при выполнении оператора ВОЗВ.

Все процедуры в А-2007 допускают рекурсию.

#### 5.2.2.17. ВОЗВРАТ

Оператор имеет вид

$$ВОЗВ\{(B)\},$$

где  $B$  – имя переменной, значение которой возвращается в программу. Если выражение  $B$  в операторе опущено или оператор отсутствует, то возвращаемым значением является указатель процедуры-функции. Тело процедуры-функции может содержать любое число операторов ВОЗВ. Выход из процедуры осуществляется при достижении любого из операторов ВОЗВ.

#### 5.2.2.18. ВКЛЮЧИТЬ

Оператор имеет вид

$$ВКЛ(I),$$

позволяет включить в число функций языка АНАЛИТИК процедуру-функцию с именем  $I$ .

#### 5.2.2.19. ИСКЛЮЧИТЬ

Оператор имеет вид



### *ИСКЛ(И),*

исключает включенную ранее процедуру-функцию с именем *И* из языка АНАЛИТИК.

#### **5.2.2.20. СТОП**

Оператор имеет вид

*СТОП*{(B)}.

Оператор останавливает выполнение программы после выполнения оператора с именем ЗН(B). Если ЗН(B) не имя оператора, то оператор СТОП пропускается. Если *B* опущен, то происходит немедленный останов программы.

Продолжение выполнения программы осуществляется нажатием клавиши ENTER.

#### **5.2.2.21. РЕТРАНСЛЯЦИЯ**

Оператор задания режима ретрансляции имеет вид

РЕТРАН(У или Р или А).

Оператор сообщает ретранслятору язык, на который ретранслируется программа (У – украинский, Р – русский, А – английский). Наличие этого оператора в программе не обязательно. По умолчанию ретрансляция осуществляется на украинский язык.

#### **5.2.2.22. КОНЕЦ**

Оператор КОНЕЦ прекращает выполнение программы.

### **5.3. Функции и операторы, образующие новые объекты**

Средства преобразования образуют в системе данных новые объекты с новыми структурными или функциональными свойствами.

Некоторые функции и операторы, которые в основном относятся к другим типам и в результате работы которых образуются новые объекты, также являются средствами преобразования. Они описаны в соответствующих пунктах:

функции ПРИМЕНИТЬ (пп. 5.3.3.8), ВЫЧИСЛИТЬ (пп. 5.3.3.3), ПРЕВРАТИТЬ (пп. 5.3.4.11), ЕСЛИ (пп. 5.2.1.10), ПРИНАДЛЕЖНОСТЬ (пп. 5.4.4), КООРДИНАТЫ (пп. 5.1.2.2), УРОВЕНЬ (пп. 5.3.4.8), ДЛИНА (пп. 5.5.2), TIME (пп. 5.5.3) и операторы ЦИКЛ (пп. 5.2.2.4.1), ПОКА (пп. 5.2.2.4.2), ПОВТ (пп. 5.2.2.4.3), СТЕРЕТЬ (пп. 5.2.2.8), ПРОГ (пп. 5.2.2.7), ПРИ (пп. 5.3.3.4), ОПТ (пп. 5.2.1.11), процедуры-функции (пп. 5.2.2.15), подпрограммы (пп. 5.2.2.16), ВВОД (пп. 5.5.7).

В АНАЛИТИКе эти функции и операторы условно можно разделить на четыре группы:

- операции и функции, определенные над числовым подмножеством данных КА;
- символьные операции;
- основные средства формульных преобразований;
- преобразования, в основном связанные со структурными свойствами объектов.

#### **5.3.1. Операции и функции, определенные над числовым подмножеством данных КА**

Функции вычисляются в рациональном и десятичном режимах в соответствии с заданной

разрядностью. Трансцендентные функции от целых или рациональных аргументов в рациональном режиме не вычисляются (возвращается указатель функции).

Приведенные символы отношений и элементарных функций отвечают общепринятым значениям:

алгебраические операции: +, -, \*, /, \*\*;

математические функции: sin, cos, tg, ctg, arcsin, arccos, arctg, arcctg, exp, sh, ch, th, cth, abs, int, ent, frc, sign;

rac – преобразует десятичное число в рациональное;

sqrt – корень;

ln – натуральный логарифм;

lg – десятичный логарифм;

gcd – наибольший общий делитель двух целых чисел;

ing – определенный интеграл (по формуле Симпсона);

min(V) – минимальное значение на множестве всех компонент вектора V ;

max(V) – максимальное значение на множестве всех компонент вектора V ;

ПОРЯДОК (x) – двоичный порядок числа x, представленного во внутреннем формате языка A-2007;

МАНТИССА (x) – мантисса числа x, представленного во внутреннем формате языка A-2007;

отношения порядка: <, =, >, <= (≤), ^= (≠), >= (≥);

операции алгебры логики: ^ (не), & (и), \ (или).

В АНАЛИТИКе реализованы некоторые соотношения алгебры комплексных чисел, которые намного облегчают реализацию выкладок в комплексной области.

Пример.

**x:=sqrt(-7);** → .264575E1\*II

**x:=ln(-2.1);** → 0.741937 E0+ PI\* II

**x:=lg(-2.1);** → .322219E0+PI\* II.

### 5.3.1.1. Сумма

Функция имеет вид

$$sum(\Pi, B_1, B_2 \{ B_3 \}, B),$$

где  $\Pi$  – связанная переменная,  $B_1$  – нижний предел,  $B_2$  – верхний предел,  $B_3$  – шаг,  $B$  – выражение.

Если  $3H(B_1)$  и  $3H(B_2)$ , и  $3H(B_3)$  – числа, то

$$\Pi \Leftrightarrow \{\Pi_j\} \Leftrightarrow \{3H(B_1) + j * 3H(B_3)\},$$

где  $j = 0, 1, \dots \leq (3H(B_2) - 3H(B_1)) / 3H(B_3)$ , порождает последовательность соответствующих значений выражения  $B(\Pi_j)$ . Результатом вычисления функции является сумма членов такой

последовательности. Шаг  $B_3$  может зависеть от  $\Pi$ . Если  $B_3$  отсутствует, то по умолчанию  $B_3 = 1$ .

Если  $ZH(B_1)$ ,  $ZH(B_2)$  или  $ZH(B_3)$  не число, то возвращается указатель функции.

Пример.

$z := (\text{sum}(j, 3, 16, 4, n^{**}j)|5); \quad \rightarrow \quad n^{**}3 + n^{**}7 + n^{**}11 + n^{**}15.$

### 5.3.1.2. Произведение

Функция имеет вид

$$\text{prod}(\Pi, B_1, B_2, \{B_3\}, B),$$

где  $\Pi$  – связанная переменная,  $B_1$  – нижний предел,  $B_2$  – верхний предел,  $B_3$  – шаг,  $B$  – выражение.

Если  $ZH(B_1)$  и  $ZH(B_2)$ , и  $ZH(B_3)$  – числа, то

$$\Pi \Leftrightarrow \{\Pi_j\} \Leftrightarrow \{ZH(B_1) + j * ZH(B_3)\},$$

где  $j = 0, 1, \dots \leq (ZH(B_2) - ZH(B_1)) / ZH(B_3)$ , порождает последовательность соответствующих значений выражения  $B(\Pi_j)$ . Результатом вычисления функции является произведение членов такой последовательности. Шаг  $B_3$  может зависеть от  $\Pi$ . Если  $B_3$  отсутствует, то по умолчанию  $B_3 = 1$ .

Если  $ZH(B_1)$ ,  $ZH(B_2)$  или  $ZH(B_3)$  не число, то возвращается указатель функции.

Пример.

$y := (\text{prod}(m, 1, 10, 2, t^{**}2)|5); \quad \rightarrow \quad t^{**}2^{**}5.$

### 5.3.1.3. Агрегация

Функция имеет вид

$$\text{agr}(\Pi, B_1, B_2, \{B_3\}, B),$$

где  $\Pi$  – связанная переменная,  $B_1$  – нижний предел,  $B_2$  – верхний предел,  $B_3$  – шаг,  $B$  – выражение.

Если  $ZH(B_1)$  и  $ZH(B_2)$ , и  $ZH(B_3)$  – числа, то

$$\Pi \Leftrightarrow \{\Pi_j\} \Leftrightarrow \{ZH(B_1) + j * ZH(B_3)\},$$

где  $j = 0, 1, \dots \leq (ZH(B_2) - ZH(B_1)) / ZH(B_3)$ , порождает последовательность соответствующих значений выражения  $B(\Pi_j)$ . Функция возвращает эту последовательность в виде вектора.

Шаг  $B_3$  может зависеть от  $\Pi$ . Если  $B_3$  отсутствует, то по умолчанию  $B_3 = 1$ .

Если  $ZH(B_1)$ ,  $ZH(B_2)$  или  $ZH(B_3)$  не число, то возвращается указатель функции.

Пример.

$$c := (\text{agr}(j, 2, 6, 2, k^{**j}) | 5); \quad \rightarrow \quad (k^{**2}, k^{**4}, k^{**6})$$

#### 5.3.1.4. Символьное дифференцирование

Вычисление производных осуществляется функциями:

$$\text{производная} - \text{dif}(\Pi_1^{**} B_1 \{ \Pi_2^{**} B_2 \} \dots, B);$$

$$\text{частная производная} - \text{DIF}(\Pi_1^{**} B_1 \{ \Pi_2^{**} B_2 \} \dots, B),$$

где  $\Pi_i$  – переменные, по которым происходит дифференцирование,  $B_i$  – выражения, определяющие порядок производных ( $3H(B_i) = ЦЧ$ ),  $B$  – дифференцируемое выражение. Для производной выполняется проверка зависимости  $3H(B)$  от  $\Pi_i$ . Значение выражения, именующего производную, вычисляется в режиме, соответствующем шестой канонической форме.

Пример.

$$s: x^{**2} * y^{**3} + 4;$$

$$d1 := (\text{dif}(x, y, s) | 6); \quad \rightarrow \quad 6 * y^{**2} * x + 6 * y * x^{**2}$$

$$u: x^{**3} + a;$$

$$a: c;$$

$$c: x^{**2};$$

$$d2 := (\text{DIF}(x, u) | 6); \quad \rightarrow \quad 3 * x^{**2}.$$

#### 5.3.1.5. Символьное интегрирование

Символьное интегрирование выражения  $B$  по переменной  $\Pi$  производится функцией  $\text{in}(\Pi, \{B_1, B_2, \dots\} B)$ :

$$\int f(x) dx = F(x) \Leftrightarrow \text{in}(x, f) \rightarrow 3H(F(x)),$$

$$\int_a^b f(x) dx = F(b) - F(a) \Leftrightarrow \text{in}(x, B_1, B_2, f) \rightarrow 3H(F(b) - F(a)),$$

где  $3H(B_1) = a$ ,  $3H(B_2) = b$  и  $F$  – первообразная функции  $f(x)$ .

Значение выражения, именующего интеграл от функции  $f(x)$ , вычисляется в режиме, соответствующем седьмой канонической форме.

### 5.3.2. Символьные операции

#### 5.3.2.1. Текстовая кавычка

Операция «текстовая кавычка» определена над любым текстом и ее операндами являются символы текста.

Пример.

$$a: \text{'текст'};$$

$$x: a[3];$$

$$\text{ВЫВ}(x); \quad \rightarrow \quad 'к'.$$

### 5.3.2.2. Конкатенация текста

Операция «#» выполняет сцепление текстовых выражений.

Пример.

**a:** 'принад';

**b:** 'лежность';

**x:=a # b;**

**ВЫВ(:x);** → 'принадлежность'.

### 5.3.2.3. Векторная запятая

В АНАЛИТИКе компоненты вектора рассматриваются как операнды операции «,».

Пример.

**a:** (x,y,z);

**u:=a[2];**

**ВЫВ(:u );** → y.

### 5.3.2.4. Скалярное умножение векторов

Операция «<\*>» – сумма попарных произведений компонент двух векторов одинаковой размерности.

Пример.

**a:** (x, 3);

**b:** (y, 4);

**u:=a <\*> b;**

**ВЫВ(:u );** → 12+x\*y.

## 5.3.3. Основные средства формульных преобразований

### 5.3.3.1. ФОРМИРОВАТЬ

Функция формирования имеет вид

$$\text{ФОРМ}(\{B_1, \dots, B_n\}B).$$

Результатом выполнения функции является выражение, полученное из именуемого выражения  $B$  после рекурсивного  $i$ -кратного ( $i = \text{ent}(\text{ЗН}(B_1))$ ) выполнения подстановок<sup>2</sup> вместо переменных именуемого выражения  $B$  значений выражений, которые они именуют. Если  $B_1$  опущено, то подстановки выполняются рекурсивно на всех уровнях. Если  $i = 0$ , то результатом выполнения функции будет  $B$ , а если  $i < 0$  или  $i$  – не число, то результатом выполнения функции считается указатель функции.

Пример.

**x:** 1/k+xk\*(1- 2/k);

**k:** sqrt(( 3 - xk)\*\*2+2);

**xk:** 4\*sin(a);

**a: t+PI;**  
**t:PI/3;**  
**u:=УРОВ(x, t);**      →      3  
**x:=ФОРМ(u, x);**      →       $1/\sqrt{((3-4*\sin(a))^{**2}+2)+(4*\sin(t+PI))*(1-2/\sqrt{((3-4*\sin(a))^{**2}+2))}$ .

### 5.3.3.2. ПОДСТАВИТЬ

Функция управления подстановками имеет вид

$$\text{ПОДСТ}(\{B_1, \dots, B_n\}B).$$

Результатом выполнения функции является выражение, полученное из выражения  $B$ , именуемого МИС после рекурсивного  $i$ -кратного ( $i = \text{ent}(3H(B_1))$ ) выполнения подстановок, выполнения всех операций и вычисления указателей всех функций в соответствии с заданными режимами. Если  $i = 0$ , то результатом выполнения функции будет  $3H(B)$ . Если  $i < 0$  или  $i$  – не число, то возвращается указатель функции.

Пример.

**z:z1; z1:z2; z2:z3; z3:5;**  
**x:a; a:a1; a1:a2; a2:4;**  
**Y1:=ПРИВ(3, ПОДСТ(z)+x);**      →      5+a.

### 5.3.3.3. ВЫЧИСЛИТЬ

Функция глобального вычисления имеет вид

$$\text{ВЫЧ}(B),$$

где  $B$  – именуемое выражение.

Она обеспечивает выполнение всех операций и функций в тех случаях, когда другие правила вычисления  $B$  блокируют их выполнение. При этом подстановки осуществляются до конца, и при выполнении операций и функций игнорируются все блокирующие правила и режимы.

Пример.

**a:=РАЦ(2/3);**      →      2/3  
**x:=ВЫЧ(a);**      →      0.666666E0.

### 5.3.3.4. Оператор ПРИ

Оператор имеет вид

$$\text{ПРИ}(P_1 \{ P_2 \} \dots \{ O \}).$$

Оператор предписывает выполнять оператор  $O$  при присвоении нового значения хотя бы одной из переменных  $P_1, P_2, \dots$ . Если  $O$  отсутствует, то при присвоении нового значения хотя бы одной из указанных переменных осуществляется вывод на дисплей их значений.

Пример.

---

<sup>2</sup> Никакие другие операции, кроме постановки, не выполняются.

**ПРИ(x ! ВЫВ(:x));**

**РАЗР:=4;**

**ЦИКЛ(i, 0, 3, 0.5, x:= i);**      →      0 .5E0 .1E1 .15E1 .2E1 .25E1 .3E1.

### 5.3.3.5. Операторы присваивания

Язык АНАЛИТИК содержит три оператора присваивания. В результате работы этих операторов левая часть оператора получает новое значение. Если конструкция в левой части, согласно синтаксису языка, именем быть не может, оператор считается пустым.

Операторы имеют вид:

1.  $V : V_1$ .

В результате значением  $V$  (если оно может быть именем) становится выражение  $B_1$ .

2.  $V :: V_1$ .

В результате значением  $V$  (если оно может быть именем) становится результат выполнения операции подстановка на один шаг входящих в выражение  $B_1$  переменных.

3.  $V := V_1$ .

При выполнении оператора последовательно аналитически вычисляются правая и левая части конструкции (левая часть не вычисляется, если это переменная). В случае, если результатом вычисления левой части является конструкция, которая может быть именем (идентификатор, координаты или пользовательская функция), то эта конструкция именуется результатом вычисления правой части.

Пример.

**z:= 5;**      →      переменная  $z$  именуется числом 5;

**x: z\*\*2+5;**      →      переменная  $x$  именуется выражением  $z**2+5$ ;

**y:: x - z;**      →      переменная  $y$  именуется выражением  $z**2+5-5$ ;

**A:= sin(y);**      →      переменная  $A$  именуется числом  $-.132351E0$ .

Однако, если

**РАЦ;**

**A:= sin(y);**      →      переменная  $A$  именуется выражением  $\sin(25)$ .

Оператор присваивания может иметь вид

$$B_1 := B_2!(\Pi_1 := \{B_3\}; \Pi_2 := \{B_4\}; \dots).$$

При этом во время вычисления  $ZH(B_2)$  значениями переменных  $\Pi_1, \Pi_2, \dots$  становятся соответственно  $B_3, B_4, \dots$ .

Если выражение  $B$  именуется объектом, то после оператора  $B :=$ ; выражение  $B$  именуется само себя.

### 5.3.3.6. ЗАМЕНИТЬ

Функция замены имеет вид

$$ЗАМ(B, B_1, B_2 \{, B_3 \}).$$

При выполнении функции осуществляется замена в  $ZH(B)$   $i$ -го ( $i = ent(ZH(B_3))$ ) вхождения  $ZH(B_1)$  на  $ZH(B_2)$ . Номер вхождения определяется при просмотре линейной записи выражения слева направо. Если  $i < 0$  или  $i$  – не число, то результатом выполнения будет указатель функции. Если  $B_3$  опущено, то результатом будет выражение, полученное после замены в  $ZH(B)$  всех вхождений  $ZH(B_1)$  на  $ZH(B_2)$ .

Пример.

**x:** $a+b**d+2*tg(c)$ ;

**z:** $sin(c)/cos(c)$ ;

**R:=3AM(x, tg(c),z);**     $\rightarrow$      $a+b**d+2*(sin(c)*cos(c)**-1)$ .

### 5.3.3.7. Функция

В А-2007 описание «пользовательской» функции имеет вид

$$I(\Pi_1\{, \Pi_2\}..): B$$

либо

$$I(\Pi_1\{, \Pi_2\}..) := B.$$

Во втором случае значение  $B$  предварительно вычисляется,

где  $I$  – имя функции (идентификатор),  $\Pi_1, \Pi_2, \dots$  – свободные переменные, которые являются ее аргументами, выражение  $B$  – тело функции, является именной формой, оно обычно содержит свободные переменные. Числа и другие переменные являются параметрами функции. Свободные переменные локализованы и могут использоваться в программе. Обращение к функции происходит, если выражение содержит указатель функции с конкретными значениями ее аргументов. Если имя функции  $I$  используется без аргументов, его значением является тело функции.

Пример.

**F(x):**  $x+3*y$ ;

**z: = F(2);**

**Выв(:z);**     $\rightarrow$      $2 + 3*y$ .

### 5.3.3.8. ПРИМЕНИТЬ

Функция имеет вид

$$ПРИМ(B, \Phi_1, \Phi_2, \{B_1\}, \{П\}, \{Ц\}, \{ * или B_2 \}),$$

где  $B$  – имя преобразуемого выражения,  $\Phi_1$  и  $\Phi_2$  – формы (описание форм полностью совпадает с описанием тела функции (пп. 5.3.3.4)),  $B_1$  – логическое выражение, задающее ограничения на значения переменных формы,  $B_2$  – номер вхождения рассматриваемой структурной части,  $П$  – переменная,  $Ц$  – целочисленный параметр, признак модификации, который учитывает свойства 0 и 1 в операциях алгебры.



Функция осуществляет применение соотношений вида  $\Phi_1 = \Phi_2$  к выражению  $B$  или его структурным частям.

Работа функции состоит из трех этапов. На первом этапе распознается принадлежность объекта с именем  $B$  или его структурных частей множеству выражений  $G$ , которое порождается формой  $\Phi_1$  путем подстановки в  $\Phi_1$  вместо переменных формы произвольных выражений, удовлетворяющих условию  $B_1$ .

На втором этапе формируется новое выражение  $W$  подстановкой в форму  $\Phi_2$  вместо переменных формы значений, при которых был успешно выполнен первый этап.

На третьем этапе происходит замещение в выражении  $B$  подвыражения, успешно прошедшего первый этап, или всего выражения  $B$  на выражение  $W$ .

Параметр\* определяет применение соотношений ко всем подвыражениям  $ZH(B)$ , а  $B_2$  – к одному подвыражению. При этом  $ZH(B_2)$  задает номер вхождения такого подвыражения. Если  $B_2$  и \* опущены, то соотношения применяются только ко всему выражению  $ZH(B)$ .

Так как по правилам записи математического анализа нулевые и единичные операнды вместе с операциями сложения, умножения, степени могут опускаться, то при сопоставлении с формой  $\Phi_1$  выражение  $B$  (или его подвыражение) может рассматриваться как

$$1 \cdot 1 \cdot \dots \cdot 1 \cdot B^{1^{1^{1^1}}} + 0 + 0 + \dots + 0. \quad (1)$$

Множество  $G$  может быть расширено также за счет некоторых элементарных функций формированием уравнений вида

$$h(A) = c,$$

где  $A$  – свободная переменная формы  $\Phi_1$ ,  $h(A)$  – подвыражение формы  $\Phi_1$ ,  $c$  – 0 или 1 либо как в (1), либо как структурная часть исследуемого выражения. Такие уравнения могут строиться рекурсивно.

Когда соответствующее выражение или его структурная часть получается из формы  $\Phi_1$  с последующим упрощением, учитывающим свойства 0 и 1, параметр  $\Pi$  может иметь значения:

- 0 – учитываются свойства 0 для операций;
- 1 – учитываются свойства 1 для операций;
- 10 – учитываются свойства 0 и 1 для операций;
- 20 – учитываются свойства 0 для операций и функций;
- 21 – учитываются свойства 1 для операций и функций;
- 210 – учитываются свойства 0 и 1 для операций и функций.

Если  $\Pi$  опущено, то свойства 0 и 1 не учитываются.

Переменная  $\Pi$  получает значение 1 или 0 в зависимости от того, применилось или нет соотношение  $\Phi_1 = \Phi_2$ .

Пример.

$f1(a,b):=(a-b)*(a+b);$   
 $f2(a,b):=a**2-b**2;$   
 $w:(\cos(2*x)-1)*(\cos(2*x)+1)+(\sin(2*x)+2)*(\sin(2*x)-2);$   
 $z1:=\text{ПРИМ}(w,f1,f2,,r1,,*); \quad \rightarrow \quad r1=1 \text{ и } z1=\cos(2*x)**2-1+\sin(2*x)**2-4$   
 $g1(t):=\sin(t)**2+\cos(t)**2; \quad /* \quad g2(t):=1 */$   
 $z2:=\text{ПРИМ}(z1,g1,1,,r2,,1); \quad \rightarrow \quad r2=1 \text{ и } z2=-4$   
 или  
 $f(A, B, X): \text{Tet}(A*X+B);$   
 $f1(A, B, X): \text{dif}(X, \text{Tet}(A*X + B));$   
 $f2(A, B, X): 1;$   
 $z : \text{Tet}(x+1)*\text{Tet}(2-3*x);$   
 $dz:=\text{dif}(x, z |6);$   
 $r:=1;$   
 $\text{ПОКА}(r=1; dz:=\text{ПРИМ}(dz, f1, f2, X?ПЕРЕМ, r, 10, 1); \rightarrow \quad \text{Tet}(2-3*x) - 3*\text{Tet}(x+1).$

#### 5.3.3.9. ПРИМЕНИТЬ вектор соотношений

Функция имеет вид

$$\text{ПРИМ}(B, I, \Pi).$$

Указатель  $I$  задает применение системы соотношений, описанной вектором с именем  $I$ , к выражению  $ZH(B)$ . Переменная  $\Pi$  получает значение 1 или 0 в зависимости от того, применилось или нет какое-либо соотношение из  $I$  (формат вектора  $I$  дан в пп. 5.2.1.8 при описании проблемной формы).

#### 5.3.4. Преобразования, связанные со структурными свойствами объектов

##### 5.3.4.1. DEN

Функция выделения знаменателя имеет вид

$$\text{DEN}(B).$$

Функция возвращает произведение сомножителей знаменателя выражения  $ZH(B)$  с положительными показателями степени и сомножителей числителя с отрицательными показателями степени. Если таких сомножителей нет, то функция возвращает 1.

Пример.

$$a: \sin(x+y) / x;$$

$$D:=\text{DEN}(a); \quad \rightarrow \quad x.$$

##### 5.3.4.2. NUM

Функция выделения числителя имеет вид

$$\text{NUM}(B).$$

Функция возвращает произведение сомножителей числителя выражения  $ZH(B)$  с положительными показателями степени и сомножителей знаменателя с отрицательными показателями степени.

Пример.

**a: sin(x+y) / x;**

**N:=NUM(a);** →  $\sin(x+y)$ .

### 5.3.4.3. КООРДИНАТЫ

Функция имеет вид

$$KООP(B, B_1\{, B_2\}),$$

где  $B$  – исследуемое выражение,  $B_1$  – структурная часть выражения  $B$ ,  $B_2$  – номер вхождения  $B_1$  в структуру  $B$ .

Функция находит и именуует в структуре МИС заданную структурную часть.

Если  $ZH(B_2)$  не положительное число или  $ZH(B_1)$  не входит в иерархическую структуру выражения  $B$ , то функция возвращает соответствующий указатель функции.

Значением указателя функции будет вектор, число компонент которого совпадает с количеством вхождений объекта  $ZH(B_1)$  в иерархию выражения  $B$ , а его компонентами являются координаты (пп. 5.1.2.2), именующие  $ZH(B_1)$  в этой структуре. Поиск ведется на всю глубину МИС.

Число  $ZH(B_2)$  задает номер вхождения. Если  $B_2$  опущено, то ищутся все вхождения.

Пример.

**b:=x+y;**

**x:=z\*\*2\*k;**

**y:=t\*v;**

**v:=sin(z\*\*2);**

**a1:=KООP(b,z\*\*2);** →  $(b[1][1], b[2][2][1])$ .

### 5.3.4.4. Главная операция

Функция имеет вид

$$ГО(B),$$

где  $B$  – выражение.

Значением указателя функции будет текст, состоящий из символа главной операции.

Пример.

**w: a + t[1,2]\*\*2 +4\*sin(PI);**

**g1:=ГО(w);** →  $' + '$

**g2:=ГО(w[3, 2]);** →  $' \sin '$

Если значением  $B$  является вектор, то главной операцией считается операция агрегации «,». Для текста главной операцией является апостроф «'».

#### 5.3.4.5. Вектор операндов

Функция имеет вид

$$BO(B),$$

где  $B$  – выражение.

Функция формирует вектор, компонентами которого являются операнды главной операции  $ZH(B)$ .

Пример.

$$w := \cos(x) * u^{**2} * (t^{**2} + 2 * t + 1);$$

$$v := BO(w); \quad \rightarrow \quad (\cos(x), u^{**2}, 1 + t^{**2} + 2 * t).$$

#### 5.3.4.6. Количество операндов

Функция имеет вид

$$KO(B),$$

где  $B$  – выражение.

Функция определяет количество операндов главной (последней) операции выражения  $ZH(B)$ .

Пример.

$$k := \ln(x);$$

$$a := 1;$$

$$b := 3 * d^{**2} + 3 * a * k + k;$$

$$z := KO(b); \quad \rightarrow \quad 3.$$

$$z := KO(b | 3); \quad \rightarrow \quad 2.$$

#### 5.3.4.7. Количество вхождений

Функция имеет вид

$$KB(B, B_1),$$

где  $B$  – выражение,  $B_1$  – структурная часть  $B$ .

Функция возвращает целое число, определяющее количество вхождений  $ZH(B_1)$  в иерархическую структуру выражения  $B$ .

Пример.

$$b := 3 * d^{**2} + 3 * a * k + k;$$

$$k := \ln(x);$$

$$x := a + 1;$$

$$z := KB(b, a); \quad \rightarrow \quad 3.$$

#### 5.3.4.8. Уровень

Функция имеет вид

$$УРОВ(B\{B_1, B_2, \dots\}),$$

где  $B$  – выражение,  $B_i$  – структурная часть  $B$ .

Функция ставит в соответствие каждому именуемому выражению  $B_i$  вектор, компоненты которого – номера структурных уровней, на которых находится этот элемент в иерархии с именем  $B$ . Если  $B_i$  не входит в иерархическую структуру выражения  $B$ , то соответствующая компонента вектора принимает значение 0. Если  $B_i$  отсутствуют, функция возвращает число уровней этой иерархии.

Если проверяется вхождение только одного выражения, то результатом будет скаляр.

Пример.

**S: a+b\*\*3;**

**a: x\*cos(y)+y;**

**b: -x\*sin(z) + y\*cos(t);**

**R:=УРОВ(S, t);**      →      2.

#### 5.3.4.9. Вектор идентификаторов

Функция имеет вид

$$ВИ(ВП \text{ или } B).$$

Функция возвращает все идентификаторы выполняемой программы (ВП) или ее части ( $ЗН(B)$ ).

Пример.

**x: a\*\*2+sin(b)+c\*d;**

**z:=ВИ(x);**      →      (4, a, b, c, d).

Результат получаем в виде вектора, первая компонента которого указывает на количество идентификаторов, остальными компонентами являются идентификаторы исследуемого объекта.

#### 5.3.4.10. Количество операторов

Функция имеет вид

$$КОП(ВП \text{ или } И),$$

где ВП – выполняемая программа,  $И$  – имя процедуры.

Функция возвращает число, определяющее количество операторов в выполняемой программе (ВП) или процедуре с именем  $И$ . Здесь  $И$  может именовать либо процедуру, либо процедуру-функцию, либо процедуру с глобальными параметрами (пп. 5.6.5.2).

Пример.

**L. ( x:y; y:z; z:a; a:4; u:=x; ВЫВ(:u); );**

**P:=КОП(L);**      →      7.

#### 5.3.4.11. ПРЕВРАТИТЬ

Функция имеет вид

$$ПРЕВ(B),$$

где  $B$  – выражение или текст.

Функция применяется к  $B$ . При этом, если  $ЗН(B)$  – не текст, то это выражение превращается в текст. Если  $ЗН(B)$  – текст, то он превращается в аналитическое выражение или программу, которые проходят синтаксический контроль. Если результат не является синтаксически правильной конструкцией, происходит АВОСТ.

Пример.

$V := 'L. x:y; y:z; z:a; a:4; u:=x; ВЫВ(:u); '$ ;

$P := ПРЕВ(V); \rightarrow 4.$

#### 5.3.4.12. ПРОГРАММА

Оператор имеет вид

$$ПРОГ(T),$$

где  $T$  – текст.

Работа оператора заключается в выполнении программы языка А-2007, заданной с помощью текста  $B$ . Если  $B$  не является текстом программы, то происходит аварийный останов.

Пример.

$a := ' b:=10; c:=20; ВЫВ(:b, ',c) '$ ;

$ПРОГ(a); \rightarrow 10' 20.$

### 5.4. Функции и операторы – распознаватели функциональных и структурных свойств выражений

Некоторые функции и операторы, которые в основном относятся к другим типам и в результате работы которых происходит распознавание функциональных или структурных свойств объектов, также являются средствами распознавания. Они описаны в соответствующих пунктах:

операции отношений порядка  $<$ ,  $=$ ,  $>$ ,  $\leq$  ( $\leq$ ),  $\neq$  ( $\neq$ ),  $\geq$  ( $\geq$ ) (пп. 5.3.1), функции ЕСЛИ (пп. 5.2.1.10), главная операция (пп. 5.3.4.4), УРОВЕНЬ (пп. 5.3.4.8), количество операндов (пп. 5.3.4.6), количество операторов (пп. 5.3.4.10), количество записей (пп. 5.5.4), количество вхождений (пп. 5.3.4.7), ДЛИНА (пп. 5.5.2) и оператор ЕСЛИ (пп. 5.2.2.3).

#### 5.4.1. Проверка зависимости

Функция имеет вид

$$ЗАВ(B, B_1 \{ B_2 \dots \}),$$

где  $B$  – выражение,  $B_1, B_2, \dots$  – структурные части выражения  $B$ .

Функция проверяет, встречаются ли в  $ЗН(B)$  объекты  $ЗН(B_1), ЗН(B_2), \dots$

Функция возвращает в виде вектора последовательность из 0 (ложь) и 1 (истина), изоморфную последовательности имен распознаваемых объектов.

Если проверяется зависимость только от одного выражения, то результатом будет скаляр.

Пример.

**s: sqrt(x\*\*2 + y\*\*2);**

**x : a\*t+b;**

**y: c\*t +sin(d);**

**z1:=3AB(s, c);** → 1

**z2:=3AB(s, c, 1.7,sin(d));** → (1,0,1).

#### 5.4.2. Операция распознавания класса

Операция имеет вид

$$B?I,$$

где  $B$  – исследуемый объект, а  $I$  – имя класса. Функция возвращает значения: 1 (истина) или 0 (ложь) в зависимости от того, принадлежит ли выражение  $3H(B)$  соответствующему классу.

Всего из множества объектов А-2007 выделяется и распознается 18 классов (пп. 5.1.5).

Пример.

**s:= (x,2,y);**

**z:=s?ВЕКТ;** → 1

**z:=s?ТЕКСТ;** → 0.

#### 5.4.3. Оператор КЛАСС

Оператор имеет вид:

$$\text{КЛАСС}(I, P_1\{P_2, \dots\}),$$

где  $P_1, P_2, \dots$  – переменные, которым в дальнейшем (в ходе выполнения программы) допустимо присваивать значения только из класса  $I$  (пп. 5.1.5). Попытка присвоить таким переменным значения из других классов приводит к аварийному останову. Вместе с операцией распознавания (пп. 5.4.2) этот аппарат можно использовать аналогично объявлению типов в других языках.

Пример.

**КЛАСС(ЦЕЛ, z);**

**z:=3;** → программа выполняется дальше

**z:=3/7;** → происходит диагностируемый аварийный останов.

#### 5.4.4. ПРИНАДЛЕЖНОСТЬ

Функция имеет вид

$$\text{ПРИНАДЛ}(B, B_1, \{B_2\}, \{B_3\}, \{B_4\}),$$

где  $B$  – выражение,  $B_1$  – форма,  $B_2$  – логическое выражение, задающее ограничения на значения переменных формы,  $B_3$  – признак модификации, который учитывает свойства 0 и 1,  $B_4$  – номер вхождения рассматриваемой структурной части. Если  $B_4$  опущено, то сопоставление происходит только со всем выражением  $B$ .

Функция ПРИНАДЛЕЖНОСТЬ является предикатом и возвращает 0 или 1. Если сопоставление произошло успешно, то запоминаются новые значения переменных формы, соответствующие объекту.

Пример.

$f(A, B, X): A * X + B;$

$z1 := a * y + 3;$

$r1 := \text{ПРИНАДЛ}(z1, f, , , );$  → 1 при  $A = a, B = 3, X = y$

$z2 := y + 3;$

$r2 := \text{ПРИНАДЛ}(z2, f, , 1, );$  → 1 при  $A = 1, B = 3, X = y$

$r3 := \text{ПРИНАДЛ}(z2, f, X^{\wedge}=y \ \& \ B^{\wedge}=y, 1, );$  → 0.

### 5.5. Средства связи с системой

Некоторые операторы, которые в основном относятся к другим типам и в результате работы которых происходит распознавание функциональных или структурных свойств объектов, также являются средствами связи с системой. Они описаны в соответствующих пунктах:

ОСТАНОВ (пп. 5.2.2.13), СТОП (пп. 5.2.2.20), КОНЕЦ (пп. 5.2.2.22).

#### 5.5.1. Оператор ПАМЯТЬ

Оператор имеет вид

$$\text{ПАМ}\{\{P\}\},$$

где  $P$  – переменная.

Оператор выполняет очистку оперативной памяти от «мусора», а переменной  $P$  присваивается в качестве значения целое число, равное объему в байтах свободной оперативной памяти.

#### 5.5.2. Функция ДЛИНА

Функция имеет вид

$$\text{ДЛИН}(B),$$

где  $B$  – выражение.

Функция возвращает число – объем памяти в байтах, занимаемый объектом  $ZH(B)$ , в ОЗУ.

#### 5.5.3. Функция TIME

Функция имеет вид

$$\text{TIME}.$$

Функция возвращает вектор, содержащий текущее время процессора (часы, минуты, секунды, миллисекунды).

#### 5.5.4. Количество записей

Функция имеет вид

$$\text{КЗ}(I),$$



где  $I$  – имя файла.

Значением указателя функции является число, определяющее количество записей в файле с именем  $I$ .

### 5.5.5. ОТКРЫТЬ

Оператор имеет вид

$$ОТКР(I_1, I_2, Режим),$$

где  $I_1$  – логическое имя файла в программе,  $I_2$  – имя файла в системе, текстовое выражение, Режим – текстовая переменная, которая может принимать значения ‘С’ – режим “только чтения”, ‘З’ – режим “только для записи”, ‘Д’ – режим “дополнения”.

Оператор используется для обеспечения взаимодействия программы с внешним носителем информации (дисковый файл, устройство печати и др.).

Пример.

**ОТКР (И1, 'c:\abc\def.ant', 'С');** → файл *C:\abc\def.ant* открыт только для чтения.

### 5.5.6. ЗАКРЫТЬ

Оператор имеет вид

$$ЗАКР(I).$$

Оператор прекращает доступ к файлу с логическим именем  $I$ . Если операнд опущен, происходит закрытие всех открытых программой файлов.

### 5.5.7. Оператор ввода

Оператор имеет вид

$$ВВОД(П\{Г\}\{B\}:\{Э_1\{Э_2\}...\}),$$

где  $П$  – логическое имя файла, с которого производится ввод, а  $Г$  определяет диапазон записей, в котором происходит выборка информации. Параметр  $Г$  имеет вид  $B_i$  или  $B_i : B_j$  и задает выборку либо  $i$ -й записи ( $i = ЗН(B_i)$ ), либо последовательность всех записей в диапазоне с  $i$  по  $j$ .

Пример.

**ВВОД(И1[5:10]:X);** → переменная  $X$  приобретает значение в результате выборки всех записей с 5-й по 10-ю из ресурса, который определяется идентификатором  $И1$ .

Если в операторе ВВОД идентификатор ресурса имеет вид  $П$  (без координат), то осуществляется ввод данных с текущей позиции.

Если количество элементов ввода больше количества записей в заданном ресурсе системы или селектор указывает на номер записи, больший количества записей, то происходит АВОСТ.

В качестве имени устройства может быть использован идентификатор файла, полученный при его открытии.

Если идентификатор ресурса для ввода не задан, то происходит ввод данных с клавиатуры. При вводе могут быть использованы все возможности, которые предоставляет интерфейс системы А-2007.

Второй параметр оператора ВВОД управляет способом преобразования введенной информации. Он может принимать один из следующих параметров:

- '%С' – ввод одного символа. Все остальные символы игнорируются;
- '%А' – используется при вводе текста в формате А-2007.

После двоеточия в операторе ВВОД следует произвольное количество элементов, определяющих объекты, значениями которых становятся вводимые данные. Данные в объекты вводятся по очереди. Для каждого нового объекта при вводе с клавиатуры открывается новое диалоговое окно, которое закрывается при окончании ввода.

### 5.5.8. Оператор вывода

Оператор имеет вид

$$\text{ВЫВ}(\{I\}, B: \{\mathcal{E}_1, \mathcal{E}_2, \dots\}),$$

где  $I$  – это идентификатор ресурса, в который осуществляется вывод,  $B$  – управляющее выражение, которое указывает способ вывода,  $\mathcal{E}_1, \mathcal{E}_2, \dots$  – выводимые элементы.

В качестве идентификатора ресурса может быть использован дескриптор файла, полученный после использования оператора ОТКР. При отсутствии идентификатора ресурса вывод осуществляется в окно «Результаты».

Выражение, управляющее форматом вывода, имеет вид:

'%{\V}\{+\}{B<sub>1</sub>}\{B<sub>2</sub>\}\Ф%...!',

'%Б' для вывода выражения без кавычек, '%С' для программы.

Здесь выражение  $B$ , заключенное в две обратные косые черты, задает количество повторений формата. Если значение выражения  $B$  меньше количества элементов для вывода, то формат вывода определяется циклическим поиском, начиная с первого элемента.

Знак «+» предписывает обязательный вывод знака для чисел.

Значение выражения  $B_1$  есть минимальная длина ширины поля выводимого элемента, а  $B_2$  – количество цифр после запятой при выводе десятичных чисел.

Выражение  $\Phi$  определяет формат выводимой информации и имеет вид  $D$  или  $F$ , или  $R$ , или  $B$ , где  $D$  – для вывода целого числа,  $F$  – для вывода десятичного числа с фиксированной точкой,  $E$  – вывод десятичного числа с плавающей точкой,  $R$  – вывод рационального числа и  $B$  – текста, набора символов, выражения.

Формы %Б и %С означают, соответственно, вывод без ограничителя текста (апострофа) и вывод программы в символьном виде.

Если управляющее выражение опущено, то осуществляется вывод значения выражения на языке А-2007.

## 5.6. Разделители

### 5.6.1. Точка «.»

Используется на выделении имени оператора (метка), для разделения целой и дробной части десятичного числа.

### 5.6.2. Точка с запятой «;»

Используется как разделитель между операторами программы и как разделитель групп координат (пп. 5.1.2.2).

### 5.6.3. Двоеточие «:»

Используется для определения диапазона (пп. 5.1.2.2).

### 5.6.4. Круглые скобки «(» и «)»

Используются при описании функций и при описании составного оператора.

### 5.6.5. Квадратные скобки «[» и «]»

Используются при задании координат и являются символами операции «подстановка».

### 5.6.6. Признак подпрограммы «@»

Ставится в начале заголовка подпрограммы.

### 5.6.7. Вертикальная черта «|»

Используется как признак номера набора преобразования (пп. 5.2.1.8).

### 5.6.8. Комментарии

Последовательность знаков языка, заключенная между знаками /\* и \*/, является не объектом системы, а комментариями программиста. Комментарий может находиться в любом месте программы.

Пример.

**Z: a\*(b+c);**

**Distr(Z): (                    /\* Реализуется закон дистрибутивности \*/**

**U:= Z|4;**

**ВЫВ(:U));                    /\* Результатом применения процедуры к выражению Z  
является выражение a\*b+a\*c \*/.**

### 5.6.9. Признак конца программы «\$»

Ставится в конце программы или подпрограммы.

## 6. Выводы

Вместе с ростом технических возможностей современных ПЭВМ и усложнением научных и прикладных задач в настоящее время резко возросли роль и значение математического

моделирования [9, 10, 17]. При этом применение компьютерной алгебры перестало быть «экзотикой», доступной только элитным дисциплинам науки и техники, как это имело место в 60-х годах прошлого столетия, когда компьютерными символьными преобразованиями пользовались в основном только в теоретической физике, «небесной механике», связанной с применением ракет и т.п. В настоящее время автоматизация символьных преобразований доступна даже студентам ВУЗов. Однако их применение ограничено в силу недостаточно развитого уровня интеллекта соответствующих алгоритмических языков, что ведет к сложности написания программ и превалировании интерактивного режима решения задач.

Семейство языков АНАЛИТИК всегда развивалось в сторону роста интеллекта, когда наряду с совершенствованием диалогового аппарата решения задач особое внимание уделялось средствам создания автоматических стандартных программ. Представленная в настоящей статье версия впервые содержит классификацию средств компьютерной алгебры. Отличительным свойством языка является то, что управление символьными преобразованиями происходит с помощью специального аппарата управления режимами вычислений, в то время как в наиболее популярных языках компьютерной алгебры управление осуществляется с помощью типов данных [9, 18, 19]. Эта версия языка содержит описание новой, более развитой, ступени аппарата интерактивного решения задач. Язык АНАЛИТИК-2007 содержит развитый аппарат автоматического решения задач с более совершенными средствами распознавания структурных и функциональных свойств объектов компьютерной алгебры. Эти возможности языка опираются на радикальное обобщение понятий объектов компьютерной алгебры. Реализация языка опирается на эффективные алгоритмы, которые используют возможности современной компьютерной техники. Большая общность объектов и функций языка делает его более компактным (язык содержит около 130 различных функций и операторов, в то время как наиболее популярные зарубежные языки содержат несколько тысяч).

## СПИСОК ЛИТЕРАТУРЫ

1. АНАЛИТИК – язык для описания вычислительных процессов с применением аналитических преобразований / В.Г. Боднарчук, В.М. Глушков, Т.А. Гринченко и др. // Теория автоматов и методы формализованного синтеза вычислительных машин и систем // Труды семинара. – Киев, 1968. – Вып. 1. – С. 3–105.
2. АНАЛИТИК (Алгоритмический язык для описания вычислительных процессов с использованием аналитических преобразований) / В.М. Глушков, В.Г. Боднарчук, Т.А. Гринченко и др. // Кибернетика. – 1971. – № 3. – С. 102 – 134.
3. АНАЛИТИК-74 / В.М. Глушков, Т.А. Гринченко, А.А. Дородницына и др. // Кибернетика. – 1978. – № 5. – С. 114–147.
4. Глушков В.М. АНАЛИТИК-79 / В.М. Глушков, Т.А. Гринченко, А.А. Дородницына, А.М. Драж, В.П. Клименко, С.Б. Погребинский, О.Н. Савчак, Ю.С. Фишман, Н.П. Царюк. – Киев, 1983. – 72 с. – (Препринт / АН УССР. Ин-т кибернетики; 83-12).
5. Бублик Б.А. Алгоритмический язык АНАЛИТИК-89 / Б.А. Бублик, Ю.С. Фишман и др. – Киев, 1990. – 43 с. – (Препринт / АН УССР. Ин-т кибернетики им. В.М. Глушкова; 90-1).
6. Морозов А.А. Принципы построения системы программирования АНАЛИТИК-91 (ориентированный на более совершенную технологию программирования научных и прикладных задач) / А.А. Морозов, В.П. Клименко, Ю.С. Фишман // Управляющие системы и машины. – 1992. – № 3. – С. 60–69.
7. АНАЛИТИК-93 / А.А. Морозов, В.П. Клименко, Ю.С. Фишман и др. // Кибернетика и системный анализ. – 1995. – № 5. – С. 127–156.
8. АНАЛИТИК-2000 / В.П. Клименко, Ю.С. Фишман, А.Л. Ляхов и др. // Математичні машини і системи. – 2001. – № 1, 2. – С. 66–99.
9. Клименко В.П. Основные тенденции развития языков систем компьютерной алгебры / В.П. Клименко, А.Л. Ляхов, Ю.С. Фишман // Математичні машини і системи. – 2002. – № 2. – С. 29–64.
10. Благовещенский Ю.В., Теслер Г.С. Вычисление элементарных функций на ЭВМ. – Киев: Техника, 1977. – 208 с.

11. Клименко В.П. Основные принципы построения систем и интерпретации языков, проблемно-ориентированных на научные и инженерные задачи // Кибернетика. – 1990. – № 1. – С. 49–56.
12. Клименко В.П. Особенности структуры данных и их преобразования в системе компьютерной алгебры АНАЛИТИК / В.П. Клименко, Ю.С. Фишман, Т.Н. Швалюк // Математичні машини і системи. – 2004. – № 2. – С. 42–48.
13. Клименко В.П. К вопросу о распознавании функциональных свойств аналитических выражений на машинах серии МИР / В.П. Клименко, С.Б. Погребинский, Ю.С. Фишман // Кибернетика. – 1973. – № 2. – С. 43–52.
14. Фишман Ю.С. Основные особенности создания и применения средств реализации численно-аналитических методов на ЭВМ // Математические машины и системы. – 1998. – № 2. – С. 9–18.
15. Основные особенности реализации системы компьютерной алгебры (АНАЛИТИК–2000) / В.П. Клименко, Ю.С. Фишман, С.В. Кондратов и др. // Математичні машини і системи. – 2002. – № 1. – С. 14–28.
16. Об оценке эффективности применения линейных списков при реализации систем компьютерной алгебры / В.П. Клименко, Ю.С. Фишман, С.В. Кондрашов и др. // Математичні машини і системи. – 2005. – № 4. – С. 55–61.
17. Клименко В.П. Аналитическое моделирование решения некоторого класса краевых задач / В.П. Клименко, А.Л. Ляхов, Т.Н. Швалюк // Радіоелектроніка. Інформатика. Управління. – 2000. – № 2 (4). – С. 82–87.
18. Maple V Realise 5. Programming Guide / В. Monagan, К.О. Geddes, К.М. Heal etc. – Springer, 1998. – 380 p.
19. Дьяконов В.П. Системы символьной математики Mathematica 2 и Mathematica 3. – М.: СК ПРЕСС, 1998. – 320 с.

*Стаття надійшла до редакції 12.06.2007*