


МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ СИСТЕМЫ ПОЛИЗ

Н.А. Резник

МИНИСТЕРСТВО
ВЫСШЕГО И СРЕДНЕГО СПЕЦИАЛЬНОГО ОБРАЗОВАНИЯ СССР



НАУЧНО-ИССЛЕДОВАТЕЛЬСКИЙ
ИНСТИТУТ
ПРОБЛЕМ ВЫСШЕЙ ШКОЛЫ

СПРАВКА
О ДЕПОНИРОВАНИИ РУКОПИСИ

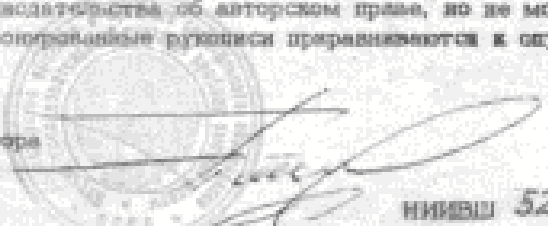
Выдана настоящей гр. Резник Н.А.
(фамилия, имя, отчество автора)

в том, что
Отделом научной информации НИИ ВШ депонирована в справочно-информационном фонде
рукопись Методическое обеспечение системы ПОЛИЗ
(название рукописи)
(№ 1225-88, деп. от 25.07.88)

Реферат (библиографическое описание) настоящей рукописи опубликован в
библиографическом указателе НИИ ВШ,
(наименование и номер реферативного издания, библиографического указателя)
серия "Средства обучения в высшей школе".
1988, вып. 6, п.39

В соответствии с Инструкцией о порядке депонирования рукописных работ по естественным, техническим и общественным наукам, утвержденной постановлением Государственного комитета Совета Министров СССР по науке и технике, Президиума Академии наук СССР, Министерства высшего и среднего специального образования СССР и Главного управления по охране государственных тайн в печати при Совете Министров СССР от 14 мая 1971 г. № 157/13, авторы депонированных рукописей сохраняют права, вытекающие из законодательства об авторском праве, но не могут претендовать на выплату гонорара; депонированные рукописи направляются в опубликованном печатном издании.

Заместитель директора



НИИ ВШ 52 Тир. 500

Ссылка: Резник Н.А. Методическое обеспечение системы ПОЛИЗ (депонир. рукопись). – НИИ ВШ, №1225, 1988, №0180047050. – 14 с.

Появление в учебном процессе электронно-вычислительных машин позволяет по-новому взглянуть на формирование математических умений и навыков. Активное развитие получила последнее время «кибернетическая психология» (М. Минский, П. Уитсон, С.М. Шапиро), в которой посредством изучения эвристических алгоритмов на ЭВМ анализируются возможные мыслительные механизмы человека.

Алгоритм перевода скобочной записи арифметических операций в обратную польскую (ПОЛИЗ – польская инверсная запись) возник при разработке трансляторов и нашел отражение в стековом устройстве распространенных в нашей стране ПМК.

Хотя необходимость «ручной» реализации этого алгоритма – явление временной, данный алгоритм позволяет вскрыть многие трудности овладения алгебраическими преобразованиями.

Предлагаем методическое обеспечение системы ПОЛИЗ в курсе «Основы информатики и вычислительной техники» средней школы с применением основных положений о визуальных математических стандартах. Все выкладки будем сопровождать подробными обоснованиями, чтобы продемонстрировать не подачу готовых образов, а формирование их в процессе обучения.

1. Визуальные инструкции по вводу числовой информации

Рассмотрим «Общий принцип составления программы вычислений на ПМК в автономном режиме по заданной формуле». Начнем с понятия инверсии.

Инверсия – перестановка слов, нарушающая обычный порядок их в предложениях, имеющая только оттенки смысла и применяемая в стилистических целях.

Данное определение удивительно точно отражает сущность польской инверсной записи математических формул, положенной в основу составления программ вычислений на программируемых микрокалькуляторах.

Принцип ее, предложенный инженером Лукашевичем (Польша), основан на усилении команды:

над числами-операндами действия-операторы осуществить!

Для наглядности проведем параллели:

Обычная математическая запись	Польская инверсная запись
$a + b$ a сложить с b	$a b +$ a с b сложить
$x : y$ x разделить на y	$x y :$ x на y разделить
$- 1,25$ Сменить знак числа $1,25$	$1,25 -$ Знак числа $1,25$ сменить
$\sin 45^{\circ}$ Вычислить синус 45°	$45^{\circ} \sin$ 45° синус вычислить

И так далее.

При инверсии типа « $a * b$ », где символ «*», есть знак арифметической (двуместной) операции, возникает необходимость отделения чисел a и b друг от друга. В противном случае, в частности, в записи « $a b *$ » символ « a » может быть воспринят как количество десятков, а символ « b » - количество единиц некоторого соответствующего числа.

Клавиша $\boxed{B\uparrow}$ выполняет роль разделителя при вводе данных. Таким образом, модель программирования выражения « $a b *$ » имеет вид: $a \boxed{B\uparrow} b \boxed{*}$.

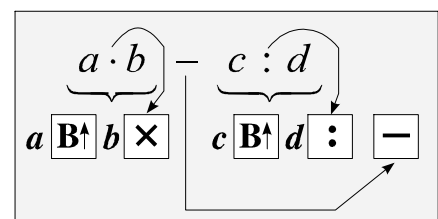
Например:

Задание	Программа
$3,25 - 1,45$	$3,25 \boxed{B\uparrow} 1,45 \boxed{-}$
$0,35 \cdot 12,5$	$0,35 \boxed{B\uparrow} 12,5 \boxed{\times}$

Рассмотрим простейшее алгебраическое выражение $ab - \frac{c}{d}$.

Элементарные рассуждения формируют следующий порядок действий:

- $a \cdot b \Rightarrow a$ на b умножить \Rightarrow результат [1]
- $c / d \Rightarrow c$ на d разделить \Rightarrow результат [2]
- из результата [1] результат [2] вычесть.



Уже здесь очевидны важные моменты:

- при предъявлении задания в *линеаризированном виде* (в строчку) порядок записи числовых величин в программе в точности соответствует последовательности их положения в исходной формуле,
- порядок записи операторов устанавливается согласно *правилам* очередности использования арифметических действий.

Отсюда легко просматривается конечный перечень команд, позволяющий осуществить программу вычислений по формулам подобного содержания.

А. Простейшие программы

Изложим предписания с одновременной демонстрацией каждого из них при составлении программы по конкретному заданию (рис. 2).

Пример 1. Составить программу вычислений по формуле

$$\frac{a - b}{m + n} - c \cdot \frac{l}{k}$$

№ команды	Реализация
1. Записать формулу в линеаризованном виде.	$(a - b) : (m + n) - (c \cdot l) : k$
2. Применить инверсию к конструкции $x * y$	$a \quad b \quad m \quad n \quad c \quad l \quad k$
3. Фиксировать операнды по порядку их следования в /1/	$a \boxed{\text{В}\uparrow} b \boxed{-} m \boxed{\text{В}\uparrow} n \boxed{+} \quad c \boxed{\text{В}\uparrow} l \boxed{+} k$
4. Выписать в столбик неиспользованные операторы по порядку их исполнения и указать их положение в /3/	
5. Оформить программу	$a \boxed{\text{В}\uparrow} b \boxed{-} m \boxed{\text{В}\uparrow} n \boxed{+} \boxed{\div} c \boxed{\text{В}\uparrow} l \boxed{+} k \boxed{\div} \boxed{-}$

Большинство одноместных операторов ПМК осуществляют вычисления значений элементарных функций от заданного аргумента (\lg , \cos , tg^{-1} , $\sqrt{\quad}$...)

В программе они всегда предваряются префиксной клавишей \boxed{F} , которая сама по себе никакого действия над числовой величиной не оказывает.

Таким образом, для конструкций типа « $f(x)$ »

модель программирования имеет вид: \boxed{Ff} .

Например:

Задание	$\arcsin 30$	$\ln 1,05$
Программа	$30 \boxed{F \sin^{-1}}$	$1.05 \boxed{F \lg}$

При программировании выражений типа « $f(a)*$ » надобность в клавише $\boxed{B\uparrow}$ отпадает.

В данном случае a и $*$ есть операнды и (что нагляднее) уже в самой линейризованной записи отделены операторами f и g .

К примеру:

Задание	$\cos \alpha / \sin \beta$	$15^2 + \sqrt{15}$
Программа	$\alpha \boxed{F \cos} \beta \boxed{F \sin} \boxed{\div}$	$15 \boxed{F x^2} 15 \boxed{F \sqrt{\quad}} \boxed{+}$

Более точное и подробное обоснование данного положения можно получить при знакомстве с работой стека МК [2].

Очевидно, что команда №3, рассмотренной выше серии, здесь нуждается в расширении.

Здесь она звучит так:

применить инверсию к конструкции типа « $f(x)$ ».

Пример 2. Составить программу вычислений по формуле:

$$\frac{\cos \alpha - \sin \beta}{\sin \gamma - \cos \delta}.$$

1. Запишем формулу в линеаризованном представлении:

$$(\cos \alpha - \sin \beta) : (\sin \gamma - \cos \delta)$$

2. Фиксируем операнды по порядку их следования:

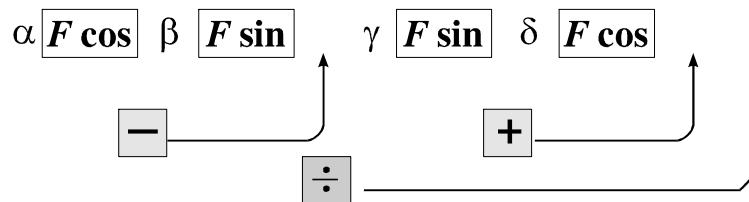
$$\alpha \quad \beta \quad \gamma \quad \delta$$

3. Применим инверсию к конструкциям типа « $f(x)$ »:

$$\alpha \boxed{F \cos} \beta \boxed{F \sin} \quad \gamma \boxed{F \sin} \delta \boxed{F \cos}$$

4. Выпишем неиспользованные операторы по порядку их исполнения и укажем их положение в [3]:

5.



5. Оформим программу:

$$\alpha \boxed{F \cos} \beta \boxed{F \sin} \boxed{-} \gamma \boxed{F \sin} \delta \boxed{F \cos} \boxed{+} \boxed{\div}$$

Итак, мы выявили два важных элемента работы с формульной информацией: линеаризация операндов и упорядочение очередности действий над ними. Перевод этих операций с одного способа записи на другой помогает осмыслить конструктивные особенности алгебраических выражений и является, таким образом, средством коррекции неправильно сформированных учений и навыков чтения математических формул.

Б. Особые и стандартные элементы программы.

Условимся в дальнейшем конструкции « $a * b$ » и « $f(x)$ » именовать **стандартными элементами программы**. Данная договоренность позволяет перейти к очередному этапу совершенствования формируемого алгоритма.

Обработку навыков программирования стандартных элементов можно приурочить к началу изучения курса «Основы информатики и вычислительной техники».

Знакомство с понятиями операнда, одноместного и двуместного операторов, инверсии даст учащимся первоначальные сведения об особенностях языка ЭВМ. Перевод данных с формульного языка предъявления на язык ПМК явится хорошей демонстрацией свойств понятия «алгоритм»: точность, конечность, дискретность, детерминированность, понятность, выполнимость и т.д.

Пример 3. Составить программу вычислений по формуле:

$$\frac{\lg(x+y)}{\lg x - \lg y}$$

Решение:

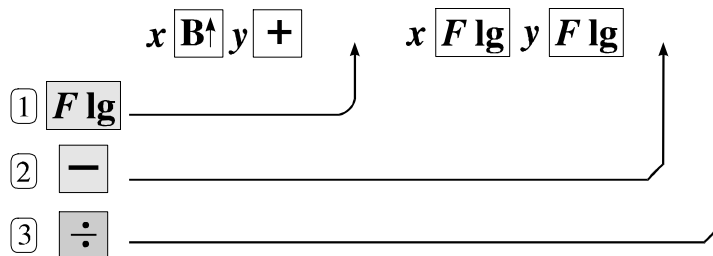
1. Представим формулу в линейризованном виде и отметим в ней

а) стандартные элементы: $x + y$, $\lg x$, $\lg y$,

б) порядок исполнения операторов, не входящих в список стандартных элементов программы

$$\lg(x+y) : (\lg x - \lg y)$$

2. Осуществим инверсию стандартных элементов и укажем место в программе операторов, отмеченных номерами



3. Оформим программу:

$$x \boxed{B\uparrow} y \boxed{+} \boxed{F \lg} x \boxed{F \lg} y \boxed{F \lg} \boxed{-} \boxed{:}$$

Применим разработанные инструкции можно и в более быстром темпе.

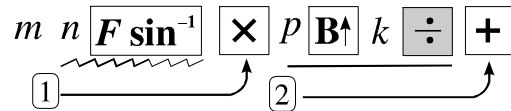
Пример 4. Составить программу вычислений по формуле:

$$m \arcsin n + (p : k)$$

1. Анализ формулы

$$m \cdot \arcsin n + (p : k)$$

2. Составление программы

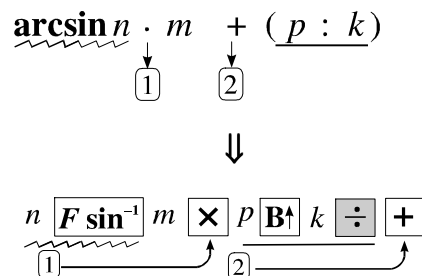


Налицо ошибка:

величины m и n не отделены друг от друга. Подобные случаи до сих пор в серии команд не учтены. Чтобы лишний раз не вводить клавишу $\boxed{B^{\uparrow}}$, придется пересмотреть ситуацию.

Известно, что от перестановки мест слагаемых и сомножителей результат не меняется.

Согласно этому имеем:



На сей раз программа вычислений по формуле $m \arcsin n + (p : k)$ составлена верно.

Итак, при наличии в исходном задании элементов вида

$$\langle a + f(x) \rangle \text{ и } \langle a \cdot f(x) \rangle$$

следует модифицировать (изменить) их на

$$\langle f(x) + a \rangle \text{ и } \langle f(x) \cdot a \rangle \text{ соответственно.}$$

Например:

Формула	$15 + \sqrt{121}$	$z \cdot e^p$
Модификация	$\sqrt{121} + 15$	$e^p \cdot z$
Программа	$121 \boxed{F \sqrt{}} 15 \boxed{+}$	$p \boxed{F e^x} z \boxed{\times}$

Для операторов $\boxed{-}$ и $\boxed{\div}$ подобная модификация приводит к перераспределению ролей членов конструкции вида « $a * f(x)$ », которую будем называть *особым элементом программы*. Действительно:

$$\begin{aligned} &\text{для «} a : f(x) \text{» имеем, «} f(x) : a \text{»} \\ &\text{аналогично, для «} a - f(x) \text{» имеем «} f(x) - a \text{»}. \end{aligned}$$

Чтобы этого не происходило, следует ввести дополнительный символ « \leftrightarrow », возвращающий компонентам особых элементов « $a : f(x)$ » и « $f(x) - a$ » их первоначальные значения:

$a + f(x)$	$f(x) + a$	$x \boxed{Ff} a \boxed{+}$
$a \cdot f(x)$	$f(x) \cdot a$	$x \boxed{Ff} a \boxed{\times}$
$a - f(x)$	$f(x) - a \leftrightarrow$	$x \boxed{Ff} a \leftrightarrow$
$a : f(x)$	$f(x) : a \leftrightarrow$	$x \boxed{Ff} a \leftrightarrow$

Таким образом, наряду с содержательными операциями (линеаризация и упорядочение действий) мы ввели в алгоритм операции, выполнение которых связано с экономией памяти МК.

Если считать, что число становится операндом только после выполнения над ним некоторой операции (разделитель, запись и считывание из памяти, вычисление значений элементарных функций и прочее), то в дальнейшем при использовании регистровой памяти микрокалькулятора понятие «особый элемент программы» становится излишним. То есть, если в выражении « $a * f(x)$ » число получено извлечением из памяти, то оно уже является операндом и модификация не требуется. Однако, на данном этапе мы составляем программу вычислений по формуле путем непосредственного ввода числовых данных в стек ПМК.

2. Алгоритм

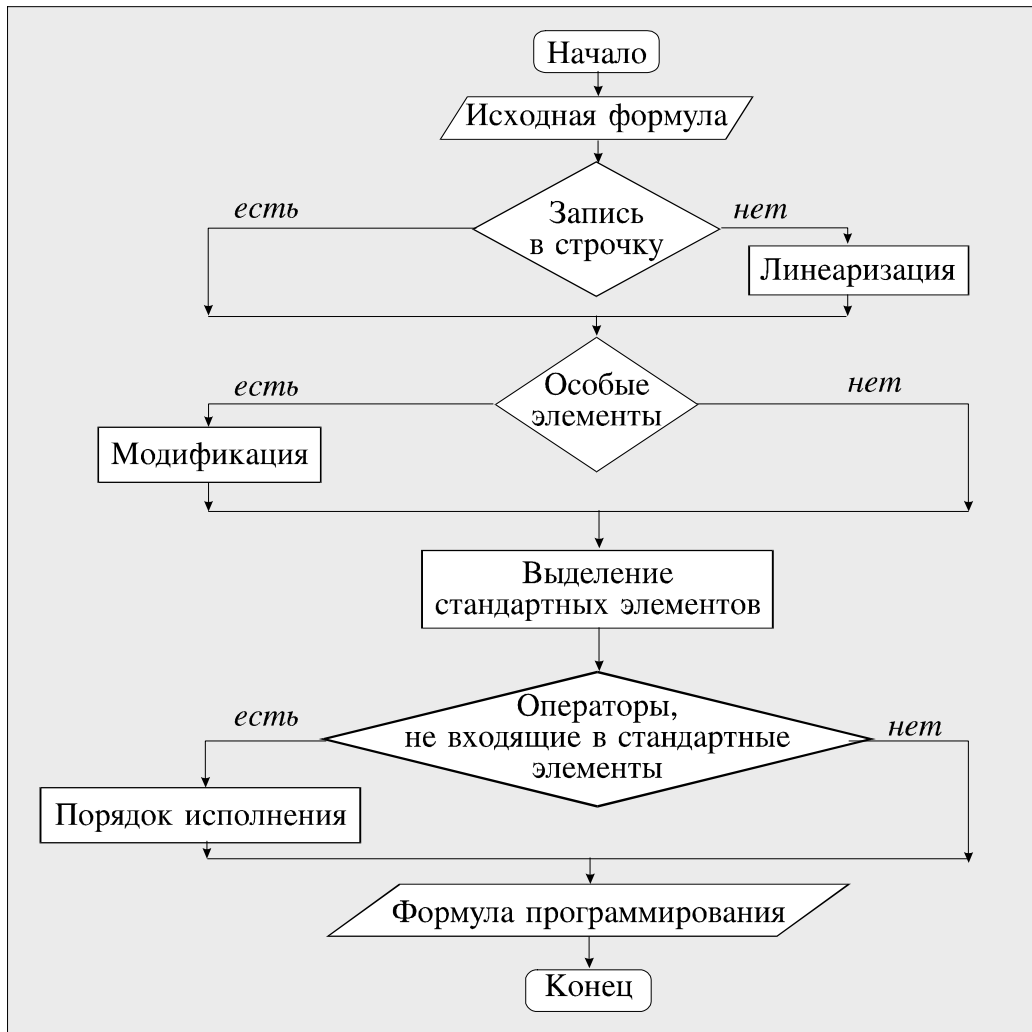
Подведем итоги. Весь процесс составления программ вычислений на микрокалькуляторах типа МК-56, МК-54 и т.д. в режиме АВТ назовем алгоритмом «ФОРМУЛА».

Он состоит из двух этапов:

1. анализ и преобразование формулы,
2. программирование.

А. Анализ и преобразование информации

Представим первый из них в виде блок-схемы алгоритма «Анализ и преобразование формулы».



В данной блок-схеме команда «Выделение стандартных элементов» не нуждается в блоке условия, поскольку практически невозможно найти математическую формулу, в которой не было бы « $a * b$ » и « $f(x)$ ».

Как уже упоминалось, команда «Особые элементы» не всегда обязательна. Хотя директива «Порядок исполнения операторов» здесь представлена в виде

простой команды (не требующей альтернатив), на самом деле – это, в отличие от всех остальных предписаний, составная команда.

Однако, поскольку мы ввели понятие стандартного элемента, на данном этапе такая директива допустима.

Полностью формализованный алгоритм перевода скобочной записи в обратную польскую приведен в книге «Машинный подход к решению математических задач» [1].

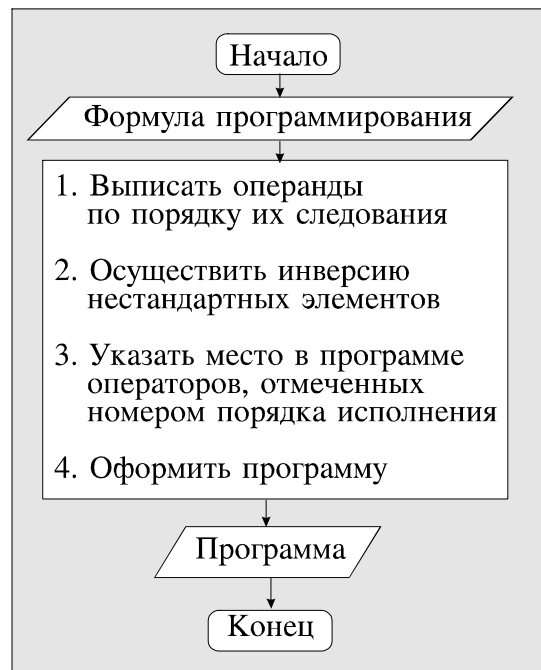
Структура и применение блок-схемы могут составить практическую иллюстрацию к теме «Простые и составные команды алгоритма».

Весь предыдущий материал текста усиленно акцентирует внимание на таком фундаментальном понятии, как «значение функции от заданного аргумента». Это следует иметь ввиду преподавателю, ведущему в одном и том же классе предметы «Алгебра и начала анализа» и «Основные информатики и вычислительной техники».

Б. Формула программирования.

Вторая часть общего алгоритма «ФОРМУЛА» представляется также в виде блок-схемы «Программирование».

Оба алгоритма объединяются в единый с помощью понятия «Формула программирования», т.е. формулы, подготовленной к переводу на язык ПМК.



Пример 5. Составить программу вычислений по формуле:

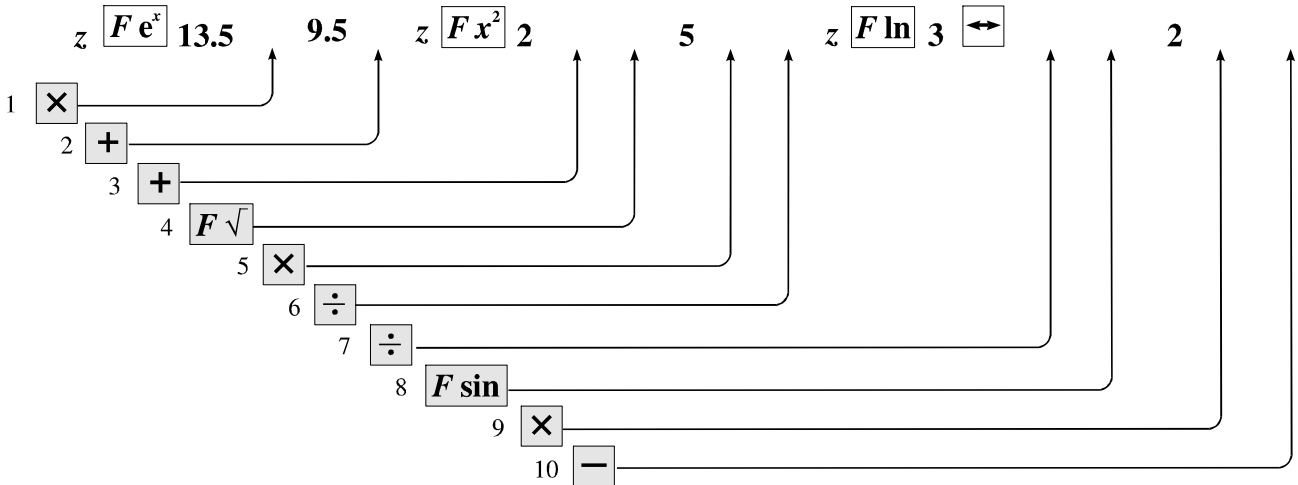
$$\frac{13,5 \cdot e^z + 9,5}{5 \cdot \sqrt{2 + z^2}} - 2 \sin \frac{3}{\ln z}$$

Решение:

1.

$$\begin{aligned} & \underbrace{(13,5 \cdot e^x + 9,5)}_{(1) \quad (2)} : \underbrace{(5 \cdot \sqrt{2 + z^2})}_{(6) \quad (4) \quad (3) \quad (5)} - \underbrace{2 \cdot \sin(3 : \ln z)}_{(10) \quad (8) \quad (7) \quad (9)} = \\ & = \underbrace{(e^x \cdot 13,5 + 9,5)}_{(1) \quad (2)} : \underbrace{(\sqrt{2 + z^2} \cdot 5)}_{(6) \quad (4) \quad (3) \quad (5)} - \underbrace{\sin(\ln z : 3 \leftrightarrow)}_{(10) \quad (8) \quad (7) \quad (9)} \cdot 2 = \end{aligned}$$

2.



Программа:

z [F e^x] 13.5 [×] 9.5 [+] z [F x²] 2 [+] [F √] 5 [×] ÷ z [F ln] 3 [↔] ÷ [F sin] 2 [×] [-]

Рассмотрим пример составления программы с применением регистров памяти.

Пример 6. Составить программу вычислений по формуле:

$$\frac{21}{p^2 - 4p + 10} - p^2 + 4p - 6$$

Последовательная реализация команд алгоритма «ФОРМУЛА» приводит к слишком длинной программе. Неоднократно придется задавать вычисления значений « p^2 » и « $4p$ ». Проанализируем структуру данного выражения, осуществив предварительно тождественные преобразования:

$$\frac{21}{p^2 - 4p + 10} - p^2 + 4p - 6 = \frac{21}{p^2 - 4p + 10} - (p^2 - 4p + 10) + 4 = \frac{21}{a} - a + 4,$$

$$\text{где } a = p^2 - 4p + 10.$$

Наметим план программирования вычислений по заданной формуле, учитывая, что в ней дважды повторяется многочлен « $a = p^2 - 4p + 10$ »:

1. Составить программы вычислений выражения $a = p^2 - 4p + 10$.

2. Занести результат вычислений выражения $a = p^2 - 4p + 10$ в один из регистров памяти, например, в RGa : $x \rightarrow Pa$.

3. Составить программу вычислений выражения $\frac{21}{a} - a + 4$, заменяя в нем

символ a на $P \rightarrow xa$ – вызов из памяти ПМК значения a .

В итоге:

P	F	x^2	4	$B \uparrow$	P	\times	$-$	10	$+$	$x \rightarrow Pa$	a	
21	$P \rightarrow xa$	\div	$P \rightarrow xa$	$-$	4	$+$						M

Итак, алгоритм «ФОРМУЛА» построен на «визуальном подходе» к процессу программирования. Он формирует необходимые навыки анализа исходного задания, что, в свою очередь, приводит к составлению наиболее компактных программ. Такой подход может быть внедрен в программное педагогическое обеспечение персонального компьютера, и тогда машина сможет помочь учащемуся или студенту самостоятельно усвоить необходимые навыки программирования вычислений по заданной формуле в режиме АВТ.

Использованная литература

1. Нивергельт Ю. и др. Машинный подход к решению математических задач / Ю. Нивергельт, Дж. Феррар, Э. Рейнгольт // Совр. математика: Вводные курсы / Пер. с англ. А.Л. Александрова и Н.Б. Лебедевой; Под. ред. В.Ш. Кауфмана. – М.: Мир, 1977. – 35 с.

2. Штернберг Л. «Зачем микрокалькулятору стек» – Квант №5, 1986 – С. 39-43.