

86-4  
17632-22

Министерство высшего и среднего специального образования  
Латвийской ССР  
Латвийский ордена Трудового Красного Знамени  
государственный университет имени Петра Стучки  
Вычислительный центр

Р.В.Фрейвалд, Д.Я.Таймля, А.И.Аузиньш,  
О.М.Елкина, П.Б.Кикуст

АЛГОРИТМИЧЕСКИЙ ЯЗЫК.

ЯЗЫК ПРОГРАММИРОВАНИЯ Р А П И Р А

Методические указания

Латвийский государственный университет им. П.Стучки  
Рига 1986

Издание представляет собой вторую часть методических указаний под редакцией Р.В.Фрейвалда для учителей средних школ, готовящих уроки по новому учебному предмету "Основы информатики и вычислительной техники" с использованием второй части пробного учебного пособия под редакцией А.П.Ершова и В.М.Монахова. Указания могут быть полезны также и студентам специальности 2013 - математика во время педагогической практики и при изучении спецкурса. В издании излагаются основные методические принципы построения нового учебного курса, даются комментарии к наиболее сложным местам учебного пособия, приводятся решения упражнений и дополнительные упражнения с решениями.

Издание подготовлено коллективом авторов:

Р.В.Фрейвалд - доктор физ.-мат.наук, зам.директора ВЦ

Д.Я.Таймина - аспирантка

А.И.Аузиньш - мл.научный сотрудник

О.М.Елкина - ст.математик

П.Б.Кикуст - кандидат физ.-мат.наук, зав.лабораторией.

УТВЕРЖДЕНО

на заседании секции по вопросам математического обеспечения  
Совета Вычислительного центра ЛГУ им.П.Стучки  
18 июня 1986 г., протокол № 3.

## АЛГОРИТМИЧЕСКИЙ ЯЗЫК

### §6. КОМАНДА ВЫБОРА (2ч.)

Этот и следующий параграф в наибольшей мере нарушают естественный порядок изложения материала в нашем двухлетнем учебном курсе. После увеличения объема времени, отведенного на изучение алгоритмического языка, стало возможно кроме команды ветвления ("если-то-иначе") подробно рассматривать и более мощную команду выбора, осуществляющую ветвление на более чем 2 случая. Если бы оба учебника писались одновременно, то команда выбора рассматривалась уже в 9 классе сразу после команды ветвления, обобщением которой она является. Действительно, эта команда только незначительно сложнее для изучения, чем команда ветвления, но позволяет легко осуществлять выполнение одной из нескольких серий команд (в зависимости от выполнения ряда логических условий) с дальнейшим переходом к общему продолжению работы.

Основная цель. Ввести понятие команды выбора; показать форму записи этой команды на примерах составления алгоритмов для решения конкретных задач.

Требования к знаниям и умениям. Ученики должны знать назначение команды выбора и ее структуру, должны уметь записывать эту команду на алгоритмическом языке и использовать ее при составлении алгоритмов.

#### Методические указания

На стр.27-28 учебника дано формальное описание выполнения команды выбора. Надо особо обратить внимание учащихся на то, что условия в команде выбора только кажутся равноправными, а на самом деле порядок, в каком записаны условия даже очень существенен. Сначала исполнитель проверяет, соблюдается ли условие I команды:





выбор

при условии 1 : серия 1

при условии 2 : серия 2

...

при условии N : серия N

иначе серия A

все

Если условие 1 соблюдается, то исполнитель выполняет команды, входящие в серию 1, и на этом выполнение команды выбора заканчивается. Если условие 1 не соблюдается, то исполнитель проверяет, соблюдается ли условие 2. Если это условие соблюдается, то он выполняет команды, входящие в серию 2, и на этом выполнение команды выбора заканчивается. Если же условие 2 не соблюдается, то исполнитель проверяет условие 3. Если оно соблюдается, то он выполняет серию 3 и т.д. Если оказывается, что ни одно из условий 1, 2, ..., N не соблюдается, то исполнитель выполняет серию A.

В примере 3 параграфа 6 учебника (строки 1-10 снизу стр.30 учебника) показано, как эту особенность определения команды выбора можно использовать для сокращения записи алгоритма. Вместе с тем хочется посоветовать учителю и ученикам не очень увлекаться таким сокращением. Дело в том, что алгоритмы должны быть удобочитаемыми, а это сокращение в некоторых случаях вызывает у читателя при разборе алгоритма необходимость помнить, какие условия уже были проверены и с каким результатом. Если условия 1,2, ...,N составлены так, что никакие два условия не могут соблюдаться одновременно, то читатель может помнить, что уже проверено, может не помнить, может помнить неточно, - от этого ничто не зависит. Если же эти условия составлены иначе, то читатель обязан все помнить точно, а это трудно.

Заметьте, что в отличие от обозначений, использованных в учебнике для 9 класса, теперь начало и конец составной команды мы соединяем вертикальной линией. Это делает текст алгоритма более наглядным и уменьшает число ошибок по пропуску конца составной команды.

Д о м а ш н е е з а д а н и е : к уроку 1 - §6(с.26-29), №1; к уроку 2 - §6(с.29-31), №5 §6.

ВОПРОСЫ И ОТВЕТЫ

1. Как записывается команда выбора? Как она выполняется? Приведите примеры.

Ответ. Команда выбора может записываться в полной и сокращенной форме. Сокращенная форма команды:

выбор

при условии 1: серия 1

при условии 2: серия 2

...

при условии N : серия N

все

Выполняя команду выбора, исполнитель последовательно проверяет все условия команды, пока не обнаружит первое из них, которое соблюдается. Найдя первое такое условие, исполнитель выполняет стоящую за ним серию команд, и на этом выполнение команды выбора заканчивается.

В сокращенной форме команды выбора не предусмотрено никаких действий на случай, если ни одно из условий команды выбора не соблюдается. В этом случае исполнитель просто пропускает команду выбора. Если же мы хотим предусмотреть какое-то действие и на этот случай, мы должны записать команду выбора в такой форме:

выбор

при условии 1: серия 1

при условии 2: серия 2

...

при условии N : серия N

иначе серия A

все

Пример. Записать алгоритм определяющий число корней уравнения  $\cos x = a$  в полуоткрытом промежутке  $[0, 2\pi[$  в зависимости от  $a$ .

алг число корней (вещ  $a$ , лит  $A$ )

арг  $a$

рез  $A$

нач

выбор

при  $a < -1$  или  $a > 1$ :  $A :=$  "корней нет"

при  $a = 1$  или  $a = -1$ :  $A :=$  "один корень"

при  $-1 < a < 1$ :  $A :=$  "два корня"

все

кон

2. В каких случаях исполнитель выполняет серию команд, стоящих после слова иначе в команде выбора?

Ответ. Исполнитель выполняет серию команд, стоящих после слова иначе в команде выбора в тех случаях, когда ни одно из условий команды выбора не соблюдается.

3. Может ли команда ветвления рассматриваться как частный случай команды выбора? Приведите примеры.

Ответ. Да, может. Например, с помощью команды ветвления алгоритм решения линейного уравнения  $ax = 1$  может быть записан так:

если  $a = 0$

то  $A :=$  "корней нет"

иначе  $x := \frac{1}{a}$ ;  $A :=$  "корень есть"

все

С помощью команды выбора это можно написать следующим образом:

выбор

при  $a = 0$ :  $A :=$  "корней нет"

иначе  $x := \frac{1}{a}$ ;  $A :=$  "корень есть"

все

Можно написать и так:

выбор

при  $a = 0$ :  $A :=$  "корней нет"

при  $a \neq 0$ :  $A :=$  "корень есть";  $x := \frac{1}{a}$

все

## УПРАЖНЕНИЯ И РЕШЕНИЯ

1. Запишите в виде команды выбора фразу из сказки: "Направо пойдешь - коня потеряешь, налево пойдешь - смертью умрешь, прямо пойдешь - друга найдешь".

Решение.

выбор

при  $a =$  "направо пойдешь" :  $A :=$  "коня потеряешь"

при  $a =$  "налево пойдешь" :  $A :=$  "смертью умрешь"

при  $a =$  "прямо пойдешь" :  $A :=$  "друга найдешь"

все

2. Напишите алгоритм вычисления значений функции, график которой изображен на рисунке 17.

Решение.

алг значение функции (вещ  $x$ , вещ  $y$ )

арг  $x$

рез  $y$

нач

выбор

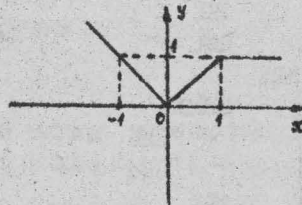
при  $x \leq 0$ :  $y := -x$

при  $0 < x$  и  $x \leq 1$ :  $y := x$

при  $x > 1$ :  $y := 1$

все

кон



3. Запишите алгоритм нахождения числа действительных корней уравнения  $x^2 = a + 1$  в зависимости от значения параметра  $a$ .



Решение.

алг число корней (вещ a, лит A)

арг a  
рез A

нач

выбор

при  $a < -1$  : A := "корней нет"  
при  $a = -1$  : A := "один корень"  
при  $a > -1$  : A := "два корня"

все

кон

4. Запишите алгоритм вычисления абсолютной величины (модуля) вещественного числа с помощью команды выбора.

Решение.

алг модуль (вещ x, вещ y)

арг x  
рез y

нач

выбор

при  $x < 0$  : y := -x  
при  $x \geq 0$  : y := x

все

кон

5. Нарисуйте график функции, значения которой вычисляются таким алгоритмом:

алг значение функции (вещ x, вещ y)

арг x  
рез y

нач

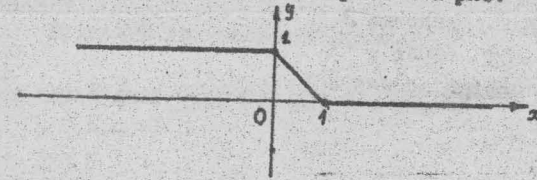
выбор

при  $x \leq 0$  : y := 1  
при  $0 < x$  и  $x \leq 1$  : y := 1-x  
при  $x > 1$  : y := 0

все

кон

Решение. График функции изображен на рис.



6. В тексте приведены два варианта алгоритма "запуск ракеты" с использованием команды выбора. Поменяв в каждом из них местами два первых условия команды выбора, записав:

а) выбор

при  $7,8 \leq u$  и  $u < 11,2$  : A := "ракета станет спутником Земли"

при  $u < 7,8$  : A := "ракета упадет на Землю"

б) выбор

при  $u < 11,2$  : A := "ракета станет спутником Земли"

при  $u < 7,8$  : A := "ракета упадет на Землю"

Останутся ли алгоритмы правильными? Почему?

Решение.

а) Алгоритм останется правильным, так как при несоблюдении первого условия, исполнитель проверит второе условие и выдаст правильный ответ.

б) Алгоритм не останется правильным. Условия в команде выбора проверяются последовательно. При этом второе условие будет проверяться только в том случае, если первое условие не соблюдается, т.е. в случае  $u \geq 11,2$ . Следовательно, при  $u < 7,8$  исполнитель выдаст неправильный ответ.

7. Запишите с помощью команды выбора:

если условие 1  
то если условие 2  
то  
серия 1  
все

иначе  
     если    условие 3  
         то серия 2  
         иначе серия 3  
     все  
все

Решение.

выбор  
     при условие 1 и условие 2 : серия 1  
     при условие 3 : серия 2  
     иначе серия 3  
все

8. Напишите алгоритм нахождения числа корней уравнения  $ax + b = 0$  в зависимости от значений параметров  $a, b$ . Не забудьте рассмотреть случай  $a = 0$ , а также  $a = 0$  и  $b = 0$ .

Решение.

алг число корней (вещ  $a$ , вещ  $b$ , лит  $A$ )  
     арг  $a, b$   
     рез  $A$

нач  
     выбор  
         при  $a = 0$  и  $b = 0$  :  $A :=$  "корней бесконечно много"  
         при  $a = 0$  и  $b \neq 0$  :  $A :=$  "корней нет"  
         при  $a \neq 0$  :  $A :=$  "один корень"  
     все  
кон

9. Напишите алгоритм нахождения числа дней в месяце, если даны: номер месяца  $n$  - целое число от 1 до 12; целое число  $a$ , равное 1 для високосного года и равное 0 в противном случае.

II

Решение.

Приведем два варианта решения:

1)

алг число дней (цел  $n$ , цел  $a$ , цел  $x$ )  
     арг  $n, a$   
     рез  $x$

нач  
     выбор  
         при  $n = 2$  : выбор  
             при  $a = 0$  :  $x := 28$   
             при  $a = 1$  :  $x := 29$   
             все  
         при  $n = 4$  или  $n = 6$  или  $n = 9$  или  $n = 11$  :  $x := 30$   
         иначе  $x := 31$   
     все  
кон

2)

алг число дней (цел  $n$ , цел  $a$ , цел  $x$ )  
     арг  $n, a$   
     рез  $x$

нач  
     выбор  
         при  $n = 1$  :  $x := 31$   
         при  $n = 2$  и  $a = 0$  :  $x := 28$   
         при  $n = 2$  и  $a = 1$  :  $x := 29$   
         при  $n = 3$  :  $x := 31$   
         при  $n = 4$  :  $x := 30$   
         при  $n = 5$  :  $x := 31$   
         при  $n = 6$  :  $x := 30$   
         при  $n = 7$  :  $x := 31$   
         при  $n = 8$  :  $x := 31$   
         при  $n = 9$  :  $x := 30$   
         при  $n = 10$  :  $x := 31$   
         при  $n = 11$  :  $x := 30$   
         при  $n = 12$  :  $x := 31$   
     все  
кон



10. Напишите алгоритм вычисления функции  $y = |x - 1| + |x - 4|$ , различающий случаи  $x \leq 1$ ,  $1 < x \leq 4$  и  $x > 4$ .

Решение.

алг вычисление функции (вещ  $x$ , вещ  $y$ )

арг  $x$

рез  $y$

нач

выбор

при  $x \leq 1$  :  $y := 5 - 2x$

при  $1 < x$  и  $x \leq 4$  :  $y := 3$

при  $x > 4$  :  $y := 2x - 5$

все

кон

11. Чтобы определить, на какую цифру оканчивается квадрат целого числа, достаточно знать лишь последнюю цифру самого числа. Напишите алгоритм, который по одной из цифр  $0, 1, 2, \dots, 9$  - последней цифре числа  $n$  - находит последнюю цифру числа  $n^2$ .

Решение. Обозначим последнюю цифру числа  $n$  через  $a$ , а через  $A$  обозначим последнюю цифру числа  $n^2$ .

алг последняя цифра (цел  $a$ , цел  $A$ )

арг  $a$

рез  $A$

нач

выбор

при  $a = 0$  :  $A := 0$

при  $a = 1$  или  $a = 9$  :  $A := 1$

при  $a = 2$  или  $a = 8$  :  $A := 4$

при  $a = 3$  или  $a = 7$  :  $A := 9$

при  $a = 4$  или  $a = 6$  :  $A := 6$

при  $a = 5$  :  $A := 5$

все

Вполне законно и следующее решение:  
алг последняя цифра (цел  $a$ , цел  $A$ )

арг  $a$

рез  $A$

нач

выбор

при  $a=0$  :  $A:=0$

при  $a=1$  :  $A:=1$

при  $a=2$  :  $A:=4$

при  $a=3$  :  $A:=9$

при  $a=4$  :  $A:=6$

при  $a=5$  :  $A:=5$

при  $a=6$  :  $A:=6$

при  $a=7$  :  $A:=9$

при  $a=8$  :  $A:=4$

при  $a=9$  :  $A:=1$

все

кон

12. Напишите алгоритм, который по номеру дня недели - целому числу от 1 до 7 - выдает в качестве результата количество уроков в вашем классе в соответствующий день.

Решение. Количество уроков для различных классов и школ будет различное. В решении рассматривается один пример распределения уроков.

алг количество уроков (нат  $x$ , нат  $y$ )

арг  $x$

рез  $y$

нач

выбор

при  $x = 1$  :  $y := 6$

при  $x = 2$  :  $y := 5$

при  $x = 3$  :  $y := 7$

при  $x = 4$  :  $y := 6$

при  $x = 5$  :  $y := 7$

при  $x = 6$  :  $y := 5$

при  $x = 7$  :  $y := 0$

все

кон

### § 7. КОМАНДА ПОВТОРЕНИЯ С ПАРАМЕТРОМ (3 ч.)

**Основная цель.** Ввести понятие команды повторения с параметром; показать форму записи этой команды на примерах составления алгоритмов для решения конкретных задач.

**Требования к знаниям и умениям.** Ученики должны знать назначение команды повторения с параметром, уметь записывать эту команду на алгоритмическом языке и использовать ее при составлении алгоритмов, в частности, в алгоритмах, работающих с линейными и прямоугольными таблицами.

#### Методические указания

Команда повторения с параметром логически более проста чем команда повторения, которая рассматривалась в учебнике 9 класса. Во всяком случае, изучение циклов естественно было бы начинать именно с циклов "для", а не с циклов "пока".

Построение циклов - один из основных вопросов нашего учебного предмета. Вряд ли будет предвечением говорить, что непрограммист становится программистом в тот момент, когда осваивает построение цикла в цикле.

Метод обучения умению построения циклов естественен: сначала ученики должны внимательно рассмотреть во всех деталях как работает один, другой, третий алгоритм (составляя и заполняя таблицу значений и особенно внимательно прослеживая как происходит выход из цикла) и лишь потом приступить к самостоятельному составлению алгоритмов с циклами (снова рисовать таблицы значений).

Можно заранее предсказать, что ученики будут совершать много ошибок, неправильно выбирая параметр цикла, вследствие чего цикл будет оканчиваться либо слишком рано, либо слишком поздно. Поэтому при проверке работ учеников учитель должен обращать на это особое внимание.

Учитель должен обратить внимание учеников на случай, когда цикл "для" начинает выполняться от  $X_{min}$  до  $X_{max}$ , где  $X_{min} > X_{max}$ . Об этом случае в учебнике сказано (см. стр. 39):

"Дело в том, что в алгоритмическом языке принято такое соглашение: в команде вида

для  $x$  от  $X_{min}$  до  $X_{max}$

нц

серия

кц

при  $X_{min} > X_{max}$  серия команд не выполняется ни разу."

Может показаться, что это исключительный случай и ученику незачем обращать на него внимание. Однако при построении циклов в циклах такие значения  $X_{min}$  и  $X_{max}$  возникают сплошь и рядом, и ученик должен быть к этому готов.

Заметьте, что в этом параграфе содержится больше упражнений чем обычно. Честно говоря, и этого количества мало. Как бы то ни было, ученики должны получить очень много задач на составление алгоритмов, содержащих ветвления и циклы. Именно здесь решается вопрос, научатся ли ученики программировать.

При прохождении этой темы учитель может использовать следующий методический прием: дать ученикам условие задачи и одновременно её решение, алгоритм с неправильно организованным циклом. Ученики должны самостоятельно эту ошибку найти и предложить способ её устранения.

Рассмотрим следующий пример. Требуется построить алгоритм, который в квадратной таблице заменяет элементы, находящиеся ниже главной диагонали на нули. Предлагается ошибочный алгоритм:

алг нули под диагональ (цел  $n$ , вещ таб  $A[1:n, 1:n]$ )

арг  $n, A$

рез  $A$

нач цел  $i, j$

для  $i$  от 1 до  $n$

нц

для  $j$  от 1 до  $i$

нц

$A[i, j] := 0$

кц

нц

кон



Ученикам предлагается найти ошибку, исполнив этот алгоритм для небольших значений  $n$ , скажем  $n=2$  или  $n=3$ .

В этом алгоритме ошибка состоит в том, что неправильно организован внутренний цикл. В итоге, алгоритм заменяет нулями не только элементы, находящиеся ниже главной диагонали, но и элементы на главной диагонали. Ошибку можно исправить, изменяя седьмую строчку алгоритма на правильную

для  $j$  от  $i$  до  $i-1$ .

(Заметьте, что при  $i=1$  получится цикл

для  $j$  от  $i$  до  $0$ ,

который не выполнится ни разу.)

Еще один полезный методический прием – доделка алгоритма. Ученикам предлагается задача и почти полностью написанный алгоритм её решения, в котором остается дописать одну строчку, например, строку с параметрами цикла.

Домашнее задание: к уроку 1 – § 7 (с.33-36), № 2, § 7; к уроку 2 – § 7 (с.36-39), № 4, § 7; к уроку 3 – § 7 (с.39-41), № 7, § 7.

## ВОПРОСЫ И ОТВЕТЫ

1. Какова форма записи команды повторения с параметром?

Ответ. Общий вид команды повторения с параметром:

для  $x$  от  $X_{min}$  до  $X_{max}$

нц

серия

кц

В этом случае при исполнении команды повторения с параметром значение параметра всякий раз увеличивается на 1. Чтобы шаг изменения параметра был бы произвольным, то команда повторения с параметром записывается так:

для  $x$  от  $X_{min}$  до  $X_{max}$  шаг  $X_{шаг}$

нц

серия

кц

2. Приведите примеры применения команды повторения с параметром.

Ответ. Команду повторения можно применить при нахождении, например,  $x^{10}$ :

$y:=1$

для  $i$  от  $1$  до  $10$

нц

$y:=y * x$

кц

Команду повторения используют при суммировании чисел, например:

алг сумма (вещ таб  $a[1 : 1986]$ , вещ  $y$ )

арг  $a$

рез  $y$

нач цел  $i$

$y := 0$

для  $i$  от  $1$  до  $1986$

нц

$y := y + a[i]$

кц

кон

3. Может ли команда повторения с параметром содержать внутри себя команду выбора или другую команду повторения с параметром? Приведите примеры.

Ответ. Да, может. Приведем два примера.

Пример 1. Составить алгоритм, который считает, сколько раз в таблице из 1986 целых чисел встречается число 17.

алг учет числа 17 (цел таб а [1:1986], цел S )

```

арг а
рез у
нач цел
  S := 0
  для i от 1 до 1986
  нц
    выбор
      при а[i] = 17 : S := S + 1
    все
  кц
кон

```

В этом случае вместо команды "выбор" можно использовать и команду "если".

Пример 2. Составить алгоритм, который переписывает в прямоугольную таблицу b элементы прямоугольной таблицы а, т.е. делающий вторую таблицу копией первой.

алг копирование таблицы (вещ таб а 1:12, 1981:1985 ,  
b 1:12, 1981:1985 )

```

арг а
рез b
нач цел i, j
  для i от 1 до 12
  нц
    для j от 1981 до 1985
    нц
      b[i, j] := a[i, j]
    кц
  кц
кон

```

4. Приведите пример применения команды повторения с параметром и шагом, отличным от 1.

Ответ. Используя команду повторения с параметром и шагом 2, можно найти, например, сумму чисел, стоящих на местах с четными номерами в линейной таблице:

```

алг сумма (вещ таб а [1:19], вещ у)
  арг а
  рез у
  нач цел
    у := 0
    для i от 2 до 19 шаг 2
    нц
      у := у + а[i]
    кц
  кон

```



## УПРАЖНЕНИЯ И РЕШЕНИЯ

1. Используя команду повторения с параметром, составьте алгоритм, который прибавляет по I ко всем элементам таблицы а. Его заголовок должен иметь вид

алг прибавление I (цел таб а [I : 100])  
арг а  
рез а

Решение.

алг прибавление I (цел таб а [I : 100])  
арг а  
рез а  
нач цел i  
для i от I до 100  
нц  
кн а [i] := а [i] + I  
кн  
кон

2. Составьте алгоритм заполнения таблицы умножения чисел от I до 99 двумя способами: с использованием и без использования команды повторения с параметром. Заголовок должен иметь вид:

алг таблица умножения (цел таб произведение [I : 99, I : 99])  
рез произведение  
 Какой из двух способов вам кажется более удобным?

Решение.

а) алг таблица умножения (цел таб произведение [I:99, I:99])  
рез произведение  
нач цел i  
для i от I до 99  
нц  
для j от I до 99  
нц  
кн произведение [i,j] := i \* j  
кн  
кон

б) алг таблица умножения (цел таб произведение [I:99, I:99])  
рез произведение

нач цел i, j  
 i := I  
пока i ≠ 99  
нц  
 j := I  
пока j ≠ 99  
нц  
 произведение [i,j] := i \* j  
 j := j + 1  
кн  
 i := i + 1  
кн  
кон

Способ а) имеет более удобную форму записи.

3. Какие изменения нужно сделать в алгоритмах "копирование таблицы" и "обращение", чтобы фигурирующие в них таблицы из 100 элементов заменить таблицами из 200 элементов?

Решение. В алгоритме "копирование таблицы" нужно 1) заменить описание таблиц а и b на (цел таб а [I:200], цел таб b [I:200]) и 2) изменить границы изменения параметра для i от I до 200.

В алгоритме "обращение" нужно 1) изменить описание таблиц: (цел таб а [I:200], цел таб b [I:200]);  
 2) изменить границы изменения параметра для i от I до 200;  
 3) изменить описание серии:  
 $b[i] := a[201-i].$

4. Рассмотрим следующий алгоритм:

```

алг обработка1(цел таб а [1:10])
  арг а
  рез а
нач цел i
  для i от 1 до 10
  ни
    а[i] := а[11-i]
  кц
кон
  
```

Каковы будут результаты исполнения этого алгоритма для величины а, значения которой приведены в таблице 15?

Таблица 15

| i    | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 |
|------|----|----|----|----|----|----|----|----|----|----|
| a(i) | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |

Решение.

| i    | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 |
|------|----|----|----|----|----|----|----|----|----|----|
| a(i) | 20 | 19 | 18 | 17 | 16 | 16 | 17 | 18 | 19 | 20 |

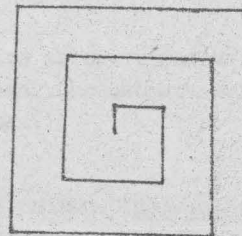
15. Выполните предыдущее упражнение для алгоритмов

```

а)
алг обработка 2 (цел таб а [1:10])
  арг а
  рез а
нач цел i, А
  для i от 1 до 10
  ни
    А := а[i]
    а[i] := а[11-i]
    а[11-i] := А
  кц
кон
  
```

7. Что нарисует исполнитель алгоритма "спираль" при  $n = 10$ , если заменить в нем команду "налево (90)" командой "направо (90)"?

Решение.



8. Исходными данными являются целочисленная таблица температура [1:31], в которой записана температура за каждый день января и величина S, равная средней январской температуре за последнее столетие. Составьте алгоритм, подсчитывающий, сколько в январе было дней с температурой большей, меньшей если равной средней. Его заголовок должен иметь вид:

```

алг подсчет (цел таб температура [1:31], цел S, цел бол,
  мен, рав)
  арг температура, S
  рез бол, мен, рав
  
```

Решение.

```

алг подсчет (цел таб температура [1:31], цел S, цел бол,
  мен, рав)
  арг температура, S
  рез бол, мен, рав
  
```

```

нач цел
  бол:=0; мен:=0; рав:=0
  для i от 1 до 31
  ни
    выбор
      при температура[i] < S : мен:=мен+1
      при температура[i] = S : рав:=рав+1
  
```



б) алг обработка 3 (цел таб а [1:10])

арг а  
рез а  
нач цел  $i$ , А  
 для  $i$  от 1 до 5  
 нц  
 А := а [ $i$ ]  
 а [ $i$ ] := а [ $n-i$ ]  
 а [ $n-i$ ] := А  
 кц  
кон

Решение.

в)

|        |    |    |    |    |    |    |    |    |    |    |
|--------|----|----|----|----|----|----|----|----|----|----|
| $i$    | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 |
| $a(i)$ | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |

В результате получаем исходную таблицу а.

б)

|        |    |    |    |    |    |    |    |    |    |    |
|--------|----|----|----|----|----|----|----|----|----|----|
| $i$    | 1  | 2  | 3  | 4  | 5  |    |    |    |    |    |
| $a(i)$ | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 |

6. Что нарисует исполнитель алгоритма "л-угольник", если а)  $л = 3$ ; б)  $л = 2$ ; в)  $л = 1$ ; г)  $л = 0$ ; д)  $л = -1$ ?

Решение. а) Правильный треугольник; б) отрезок длиной 1, при чем исполнитель по этому отрезку пройдет дважды: вперед и назад; в) отрезок длиной 1; в случаях г) и д) исполнитель ничего не нарисует, потому что, при  $x_{\min} > x_{\max}$  серия команд не выполняется ни разу.

при температура [ $i$ ] > 5 : бол := бол + 1  
все  
кц  
кон

9. Составьте алгоритм заполнения таблицы числа дней от 1801 до 1899 года. (Высокие годы - каждый четвертый, начиная с 1804 года.)

Решение.

алг заполнение таблицы (цел таб число дней [1801:1899])  
рез число дней  
нач цел  $i$   
 для  $i$  от 1801 до 1899  
 нц  
 число дней [ $i$ ] := 365  
 кц  
 для  $i$  от 1804 до 1899 шаг 4  
 нц  
 число дней [ $i$ ] := число дней [ $i$ ] + 1  
 кц  
кон

10. Составьте алгоритм заполнения таблицы числа дней от 1801 до 2000 года. Учтите, что 1900 год - не высокий, а 2000 год - высокий.

Решение.

алг заполнение таблицы (цел таб число дней [1801:2000])  
рез число дней  
нач цел  $i$   
 для  $i$  от 1801 до 2000  
 нц  
 число дней [ $i$ ] := 365  
 кц  
 для  $i$  от 1804 до 2000 шаг 4  
 нц

```

      число дней [i] := число дней [i] + 1
    КЦ
    число дней [1900] := 365
  КОН

```

11. Составьте алгоритм вычисления числа воскресений в 1991 году. (Первое воскресенье 1991 году - 6 января.)

Решение.

```

алг число воскресений (цел 5)
  рез 5
нач  цел i
      5 := 0
      для i от 6 до 365 шаг 7
        нц
          5 := 5 + 1
        кц
      кц
конт

```

12. Составьте алгоритм заполнения таблицы числа дней для всех лет с 1-го до 2000-го года. Указание: високосными являются годы, номер которых делится на 4, за исключением тех, номера которых делятся на 100, но не делятся на 400.

Решение.

```

алг заполнение таблицы (цел таб число дней [1:2000])
  рез  число дней
нач  цел i
      для i от 1 до 2000
        нц
          число дней [i] := 365
        кц
      для i от 4 до 2000 шаг 4
        нц
          число дней [i] := число дней [i] + 1
        кц
      для i от 100 до 2000 шаг 100
        нц

```

```

      число дней [i] := число дней [i] - 1
    кц
    для i от 400 до 2000 шаг 400
      нц
        число дней [i] := число дней [i] + 1
      кц
    кц
конт

```

13. Составьте алгоритм нахождения а) минимального ; б) максимального элемента таблицы а [1:100].

Решение.

```

а)
алг минимальный элемент (вещ таб а [1:100], вещ мин)
  арг а
  рез мин
нач  цел i
      мин := а [1]
      для i от 1 до 100
        нц
          выбор
            при а [i] < мин : мин := а [i]
          все
        кц
      кц
конт
б)
алг максимальный элемент (вещ таб а [1:100], вещ макс)
  арг а
  рез макс
нач  цел i
      макс := а [1]
      для i от 1 до 100
        нц
          выбор
            при а [i] > макс : макс := а [i]
          все
        кц
      кц
конт

```



14. Составьте алгоритмы для рисования пятиконечной звезды.

Решение.

```

алг звезда
нач цел i
  для i от 1 до 5
    нц
      вперед (6)
      направо (144)
    кц
конец

```

15. Составьте алгоритм вычисления суммы квадратов всех натуральных чисел от 1 до 50:  $S = 1^2 + 2^2 + 3^2 + \dots + 50^2$ .

Решение.

```

алг сумма квадратов (цел S)
рез S
нач нат n
  S := 0
  для n от 1 до 50
    нц
      S := S + n * n
    кц
конец

```

16. Составьте алгоритм вычисления первых 30 чисел Фибоначчи. (Эти числа определяются так:  $a_1 = 1$ ,  $a_2 = 1$ , а каждое следующее число равно сумме двух предыдущих:  $a_{n+1} = a_n + a_{n-1}$ . Вот первые несколько чисел Фибоначчи: 1, 1, 2, 3, 5, 8, 12, 21, ...) Ответ получите в виде таблицы а [1:30].

Решение.

алг числа Фибоначчи (нат таб а [1:30])

```

рез а
нач цел
  a[1] := 1
  a[2] := 1
  для i от 3 до 30
    нц
      a[i] := a[i-1] + a[i-2]
    кц
конец

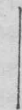
```

17. Что нарисует исполнитель алгоритма "n-угольник" (пример 7), если в нем угол  $360^\circ/n$  заменить вдвое большим углом  $720^\circ/n$ ? Рассмотрите случаи  $n = 3, 4, 5, 6, 7, 8$ .

Решение.



При  $n = 3$  исполнитель нарисует правильный треугольник.



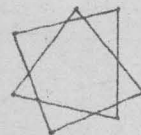
При  $n = 4$  исполнитель нарисует отрезок длиной 1.



При  $n = 5$  исполнитель нарисует пятиконечную звезду.



При  $n = 6$  исполнитель нарисует правильный треугольник.



При  $n = 7$  исполнитель нарисует фигуру, изображенную на рисунке.



При  $n = 8$  исполнитель нарисует квадрат.

18. Составьте алгоритм циклической перестановки элементов таблицы  $a[1:1000]$ , при которой  $a[i]$  перемещается в  $a[i+1]$ , а  $a[1000]$  перемещается в  $a[1]$ .

Решение.

```

алг циклическая перестановка (вещ таб a [1:1000])
  арг а
  рез а
нач цел i, вещ R
  R := a[1000]
  для i от 1 до 999
    нц
      a[1001-i] := a[1000-i]
    кц
  a[1] := R
кон
  
```

При решении этой задачи обязательно начинать циклическую перестановку справа, ибо в противном случае, т.е. начиная перестановку слева, получим таблицу, в которой все элементы будут равны.

19. Составьте алгоритм циклической перестановки элементов таблицы  $a[1:1000]$  на заданное число  $k$  шагов, так что элемент  $a[i]$  перемещается в  $a[i+k]$ , а последние  $k$  элементов таблицы, которым "не хватило места", перемещаются в "освободившиеся" первые  $k$  элементов таблицы.

Решение.

```

алг циклическая перестановка (вещ таб a [1:1000], цел k)
  арг а, k
  рез а
нач цел i, вещ таб в [1:k]
  для i от 1000-k+1 до 1000
    нц
      в [i-1000+k] := a[i]
    кц
  
```

```

  для i от 1 до 1000-k
    нц
      a[1000-i+1] := a[1000-k-i+1]
    кц
  для i от 1 до k
    нц
      a[i] := в[i]
    кц
кон
  
```

20. Дана таблица  $a[1:500]$ . Напишите алгоритм, записывающий в качестве  $i$ -го элемента этой таблицы сумму стоящих в ней чисел от  $a[1]$  до  $a[i]$  включительно.

Решение.

```

алг таблица сумм (вещ таб a [1:500])
  арг а
  рез а
нач цел i
  для i от 2 до 500
    нц
      a[i] := a[i] + a[i-1]
    кц
кон
  
```

Поясним это решение на числовом примере. Пусть дана таблица:

| $i$    | 1 | 2  | 3   | 4    | 5     |
|--------|---|----|-----|------|-------|
| $a(i)$ | 1 | 20 | 300 | 4000 | 50000 |

После выполнения алгоритма получим следующую таблицу:

| $i$    | 1 | 2  | 3   | 4    | 5     |
|--------|---|----|-----|------|-------|
| $a(i)$ | 1 | 21 | 321 | 4321 | 54321 |



## Самостоятельная работа

1. Составьте алгоритм для вычисления суммы

$$1 + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{n}.$$

2. Составьте алгоритм, проверяющий, являются ли элементы таблицы вещ таб a [1:100] арифметической прогрессией с разностью d. Алгоритм должен иметь следующий вид:

```

алг проверка (вещ таб a [1:100], вещ d, лит:ответ)
  арг a, d
  рез ответ
нач
  ...
кон

```

Литерная величина ответа после выполнения алгоритма должна получить одно из двух значений "является прогрессией" или "не является прогрессией".

## Решения

1. При решении задачи используем команду повторения с параметром, и предполагаем, что сумма определена для  $n > 0$ .

```

алг сумма (цел n, вещ S)
  арг n
  рез S
  нач цел i
  S := 1
  для i от 2 до n
  нц
  S := S +  $\frac{1}{i}$ 
  кц
кон

```

2. Члены арифметической прогрессии  $a_1, a_2, \dots, a_{100}$  должны удовлетворять равенству  $a_{i+1} = a_i + d$ , где  $1 \leq i \leq 99$ .

```

алг проверка (вещ таб a [1:100], вещ d, лит:ответ)
  арг a, d
  рез ответ
  нач цел i
  ответ := "является прогрессией"
  для i от 1 до 99
  нц
  если  $a[i+1] \neq a[i] + d$ 
  то ответ := "не является прогрессией"
  все
  кц
кон

```

## § 8. ВСПОМОГАТЕЛЬНЫЕ АЛГОРИТМЫ ВЫЧИСЛЕНИЯ ЗНАЧЕНИЙ ФУНКЦИЙ (2 ч.)

Основная цель. Ввести новый тип вспомогательных алгоритмов: вспомогательные алгоритмы вычисления значения функций; показать форму записи таких алгоритмов на примерах составления алгоритмов для решения конкретных задач.

Требования к знаниям и умениям. Ученики должны знать назначение вспомогательных алгоритмов вычисления значений функций, должны уметь составлять и исполнять алгоритмы и исполнять алгоритмы, содержащие вызовы таких вспомогательных алгоритмов.

### М е т о д и ч е с к и е    у к а з а н и я

Использование вспомогательных алгоритмов разнообразно. Во-первых, обращение к вспомогательному алгоритму полезно, когда у нас имеется ранее составленный алгоритм, который может войти в качестве составной части в алгоритм, составляемый теперь. Во-вторых, еще при изучении материала 9 класса неоднократно отмечалось, что сложные алгоритмы следует составлять методом последовательного уточнения "с в е р х у в н и з", т.е. сначала продумать из каких составных частей алгоритм будет состоять и оформить алгоритм в виде последовательности обращений к вспомогательным алгоритмам, которые еще не составлены. Потом для завершения всей работы остаются эти вспомогательные алгоритмы составить, но это уже более простая задача, чем составление исходного алгоритма.

Например, если требуется составить алгоритм заполнения таблицы умножения (см. пример 3 на стр. 35 учебника), то внутренний цикл можно оформить в виде вспомогательного алгоритма. Таким образом, более сложный алгоритм, требующий применения цикла в цикле сводится к построению двух алгоритмов, состоящих из одного цикла каждый.

В § 8 учебника рассматривается еще одно применение вспомогательных алгоритмов. Предлагается способ более коротко и наглядно записывать алгоритмы, вычисляющие функции.

Это позволяет записывать алгоритмы более привычным для математика способом. Например, при вычислении значений функции трех аргументов

$$u = ||x - |y+1|| + |x + |y|| + 1$$

с помощью вспомогательного алгоритма вычисления модуля действительного числа (см. с. 90 учебника 9 класса) в 9 классе мы строим алгоритмы следующим образом.

алг вычисление (вещ  $x, y, z, u$ )

арг  $x, y, z$

рез  $u$

нач

вещ  $a, b, c, d$

МОД  $(y+1, a)$

МОД  $(x-a, b)$

МОД  $(y, c)$

МОД  $(z+c, d)$

МОД  $(b+d, u)$

$u := u + 1$

кон

Вспомогательные алгоритмы вычисления значений функций, введенные в § 8 учебника 10 класса, позволяют этот же алгоритм записать намного более наглядно.

алг вычисление (вещ  $x, y, z, u$ )

арг  $x, y, z$

рез  $u$

нач

$u := abs(abs(x - abs(y+1)) + abs(z + abs(y))) + 1$

кон

Учитель легко заметит, что приведенный нами пример напоминает пример I из § 8 учебника. Большая выразительность примера достигнута тем, что здесь в одной формуле, выражающей зависимость значения функции от значений аргументов, содержится больше вхождений имени функции, вычисляемой вспомогательным алгоритмом. Если учителю понадобится еще более



убедительный пример, то можно взять функцию, определяемую формулой с 8, 15, 20 или еще большим числом вхождений обозначения одной и той же функции. Например,

$$u = \sin(\sin(\sin(x) + \sin(y)) - \sin(y - \sin(z))) + \sin(z) - \sin(x).$$

Надо обратить внимание учеников на то, что заголовок вспомогательного алгоритма вычисления значения функции отличается от заголовков любых других алгоритмов. Отличие состоит в том, что в заголовке перед названием функции указывается тип значений функции.

Для учеников несколько неожиданным может сказаться употребление служебного слова знач. Это первый случай в нашем курсе, когда служебным словом обозначается переменная. В учебнике (стр. 46) сказано: "Эта величина используется как переменная, тип которой совпадает с типом значений функции. Указывать тип этой величины не следует". Не следует ее указывать и в заголовке в качестве результата алгоритма.

В примерах 1-3 из § 8 учебника переменная знач получает свое значение только один раз и потом уже это значение не меняет. Однако такое ограниченное использование переменной знач не обязательно. В остальных примерах из § 8 учебника переменная знач свое значение меняет неоднократно. Так в примере 4 переменная знач последовательно принимает значения

$$0, a[1], a[1] + a[2], a[1] + a[2] + a[3], \dots, a[1] + a[2] + \dots + a[100], \\ a[1] + a[2] + \dots + a[100].$$

100

Учитель может предложить в качестве дополнительного упражнения задачу построения алгоритма нахождения большего из четырех чисел, алгоритмов нахождения большего из пяти, шести и т.д. чисел. При этом можно спрашивать учеников, как они предполагают это организовать. Например, большое число из восьми чисел можно находить, сперва найдя большее число из первых четырех, потом большее из последних четырех чисел, и, наконец, большее из этих двух промежуточных результатов.

Домашнее задание: к уроку 1 - § 8 (с. 44-47), № 1 § 8; к уроку 2 - § 8 (с. 47-51), № 5, § 8.

## ВОПРОСЫ И ОТВЕТЫ

1. Как используются вспомогательные алгоритмы вычисления значений функций? Приведите примеры.

Ответ

Для многократного вычисления значений функций в программе используют вспомогательный алгоритм особого вида. Название такого алгоритма используется как обозначение вычисляемой им функции. Например:

$$y := \text{факториал}(3) \quad z := \text{max } 3(?, II, 9)$$

2. Как составляется заголовок алгоритма вычисления значений функции?

Ответ

Заголовок алгоритма вычисления значений функции начинается со служебным словом алг. Далее указывается тип значений функции (например: вещ, цел) и название функции. Затем в скобках список ее аргументов с указанием типа.

3. Для чего используется величина знач в алгоритме вычисления значений функций?

Ответ

Величина знач используется как переменная, в которую помещается значение функции (результат выполнения вспомогательного алгоритма). Тип этой величины совпадает с типом значений функции, указанном в заголовке.

## УПРАЖНЕНИЯ И РЕШЕНИЯ

1. Напишите алгоритмы вычисления значения выражения  $y = |x + 1| + |x - 1|$ , используя в качестве вспомогательного алгоритма вычисления значений функции abs(x).

Решение

алг вещ вычисление I(вещ x)

нач

знач := abs(x+1) + abs(x-1)

кон

2. Напишите алгоритмы вычисления значений функции  $y = \max 4(a, b, c, d)$ , значение которой равно большому из четырех ее аргументов.

Решения

алг вещ мах 4 (вещ  $a, b, c, d$ )

нач

знач: = мах2(мах2( $a, b$ ), мах2( $c, d$ ))

кон

алг вещ мах 4 (вещ  $a, b, c, d$ )

нач

знач: = мах2( $a, b$ )

знач: = мах2(знач,  $c$ )

знач: = мах2(знач,  $d$ )

кон

алг вещ мах 4 (вещ  $a, b, c, d$ )

нач

знач: = мах 2 (мах 3 ( $a, b, c$ ),  $d$ )

кон

3. Напишите алгоритмы алг вещ больший (вещ таб  $a[1:100]$ ), вычисляющий значения функции больший ( $a$ ), значением которой является наибольший из элементов таблицы  $a$ .

Решение

алг вещ больший (вещ таб  $a[1:100]$ )

нач цел  $i$

знач: =  $a[1]$

для  $i$  от 1 до 100

нц

если  $a[i] > \text{знач}$

то знач: =  $a[i]$

все

кц

кон

4. Напишите алгоритмы алг вещ сумма (вещ таб  $a[1:100]$ ), вычисляющий сумму всех элементов таблицы  $a$ .

Решение

алг вещ сумма (вещ таб  $a[1:100]$ )

нач цел  $i$

знач: = 0

для  $i$  от 1 до 100

нц

знач: = знач +  $a[i]$

кц

кон

5. Используя алгоритмы вычисления функции факториал ( $n$ ) в качестве вспомогательного, напишите алгоритмы вычисления суммы факториалов всех четных чисел от 2 до 100.

Решение

алг сумма факториалов (цел  $S$ )

рез  $S$

нач цел  $i$

$S$ : = 0

для  $i$  от 2 до 100 шаг 2

нц

$S$ : =  $S$  + факториал ( $i$ )

кц

кон

6. Напишите алгоритмы нахождения приближенного значения функции  $y = \sin x$  при малых  $x$  по формуле

$$y = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \frac{x^9}{9!}$$

Решение

алг вещ sin (вещ  $x$ )

нач цел  $i$

знач: = 0

для  $i$  от 1 до 5

нц

кц



$$\text{знач:} = \text{знач} + \frac{x \text{ иж } (2^i - 1) \text{ и } (-1) \text{ иж } (i + 1)}{\text{факториал}(2^i - 1)}$$

кц

кон

Пояснение. Применение цикла от 1 до 5 не обязательно. Можно все выражения записать как оператор присваивания.

Наилучшим можно считать следующий:

алг вещ sin (вещ x)

нач вещ y

знач: = x

y: = x

y: =  $\frac{y * x * x}{2 * 2}$

знач: = знач - y

y: =  $\frac{y * x * x}{4 * 5}$

знач: = знач + y

y: =  $\frac{y * x * x}{6 * 7}$

знач: = знач - y

y: =  $\frac{y * x * x}{8 * 9}$

знач: = знач + y

кон

Такой код вычислений можно записать и с помощью цикла

алг вещ sin (вещ x)

нач цел i, вещ y

знач: = x

y: = x

для i от 1 до 4

ны

y: =  $\frac{y * x * x}{2 * i * (2 * i + 1)}$

знач: = знач + y \* (-1) \* i

кц

кон

7. Напишите алгоритм, находящий а) последнюю цифру десятичной записи натурального числа x; б) предпоследнюю цифру десятичной записи натурального числа x. При этом можно использовать алгоритм-остаток, имеющийся в библиотеке алгоритмов.

Решение

алг цел остаток (цел делимое, делитель)

нач

знач: = делимое

пока знач ≥ делитель

ны

знач: = знач - делитель

кц

кон

а) алг цел последняя цифра (цел x)

нач

знач: = остаток (x, 10)

кон

б) алг цел предпоследняя цифра (цел x)

нач

x: =  $\frac{x - \text{остаток}(x, 10)}{10}$

знач: = остаток (x, 10)

кон

8. Напишите алгоритм, находящий сумму цифр десятичной записи трехзначного натурального числа x.

Решение

алг нат сумма цифр (нат x)

нач нат a, b, c

c: = остаток (x, 10)

x: =  $\frac{x - c}{10}$

b: = остаток (x, 10)

a: =  $\frac{x - b}{10}$

знач: = a + b + c

кон

9. Напишите алгоритм, находящий сумму трехзначного числа  $N$  с трехзначным числом  $N$ , полученным из первого перестановкой его цифр в обратном порядке.

Решение

Пояснение.  $a$  - первая,  $b$  - вторая,  $c$  - третья цифра числа.

алг нат сумма 2 (нат  $N$ )

нач цел  $a, b, c$

$c := \text{остаток}(N, 10)$

$b := \text{остаток}(\frac{N - c}{10}, 10)$

$a := \frac{N - c - 10 * b}{100}$

знач:  $= N + a + 10 * b + 100 * c$

кон

Если сначала решить упражнение математически, то можно найти, что результат должен быть равен  $101(a+c) + 20b$ , где  $a, b, c$  - соответственно цифры.

алг нат сумма 2 (нат  $N$ )

нач цел  $b$

$b := \text{предпоследняя цифра}(N)$

знач:  $= 101 * \text{сумма цифр}(N) - 81 * b$

кон

10. Напишите алгоритм подсчета количества "счастливых" автобусных билетов. (Билет называется счастливым, если сумма первых трех цифр номера билета равна сумме последних трех цифр).

Решение

алг билет (цел  $x$ )

рез  $x$

нач цел таб сумма [0:27]

цел  $a, b, c, s, i$

$x := 0$

для  $i$  от 0 до 27

нц

$\text{сумма}[i] := 0$

кц

для  $a$  от 0 до 9

нц

для  $b$  от 0 до 9

нц

для  $c$  от 0 до 9

нц

$S := a + b + c$

сумма ( $S$ ):  $= \text{сумма}(S) + 1$

кц

кц

кц

для  $i$  от 0 до 27

нц

$x := x + \text{сумма}(i) * \text{сумма}(i)$

кц

кон

Пояснение: В  $i$ -тый элемент таблицы сумма подсчитывается число трехзначных чисел с суммой цифр  $i$ . Например сумма [26] = 3 дает числа 899, 989, 998. Каждый из этих чисел может быть первым или вторым. Значит счастливых билетов с суммой 26 всего 9. В общем случае число счастливых билетов с суммой цифр  $i$  равно квадрату числа трехзначных чисел с суммой цифр  $i$ . Подсчет по всем возможным суммам от 0 до 27 дает общее число счастливых билетов.

Этот алгоритм можно реализовать и на машине и достаточно быстро получить результат.

Однако можно написать более простые алгоритмы, которые хуже тем, что будут работать 1000 раз дольше.

Два таких примера.

алг билет (цел  $x$ )

рез  $x$

нач цел  $y, z, i$

$x := 0$

для  $i$  от 0 до 999999

нц

$y := \text{остаток}(i, 1000)$

$z := \frac{i - y}{1000}$



```

если сумма цифр (y) = сумма цифр (z)
  то x := x + 1
все
кц
кон
алг билет (цел x)
  рез x
  нач цел a, b, c, d, e, i
  x := 0
  для a от 0 до 9
    нц
    для b от 0 до 9
      нц
      для c от 0 до 9
        нц
        для d от 0 до 9
          нц
          для e от 0 до 9
            нц
            для i от 0 до 9
              нц
              если a + b + c = d + e + i
                то x := x + 1
              все
            кц
          кц
        кц
      кц
    кц
  кц
кон

```

## ДОПОЛНИТЕЛЬНЫЕ УПРАЖНЕНИЯ И РЕШЕНИЯ

1. Напишите алгоритм создания таблицы значений выражения  $y = |x-1| + |x+1|$  для целых  $x$  от  $-10$  до  $+10$ .

Решения.

алг таблица (вещ таб A [-10:+10])

```

рез A
нач цел i
  для i от -10 до +10
    нц
    A[i] := вычисление I(i)
  кц
кон

```

2. Напишите алгоритм находящий сумму цифр данного числа.

Решение.

алг цел сумма - ц (нат x)

```

нач цел y
  знач := 0
  пока x > 0
    нц
    y := остаток(x, 10)
    знач := знач + y
    x := (x - y) / 10
  кц
кон

```

3. Напишите алгоритм находящий квадрат целого числа с максимальной суммой цифр среди квадратов меньших  $10^6$ .

алг квадрат - ц (нат x)

```

рез x
нач цел i
  x := 0
  для i от 1 до 1000
    нц
    если сумма -ц(i*i) > сумма -ц(x)

```

кон

кц

всв

то x: i i

## § 9. АЛГОРИТМЫ РАБОТЫ С ЛИТЕРНЫМИ ВЕЛИЧИНАМИ (2 ч.)

Основная цель. Ознакомить учеников с простейшими операциями над литерными величинами.

Требования к знаниям и умениям. Ученики должны знать операции соединения (склеивания) и вырезки текста, должны уметь правильно применять эти операции при составлении простейших алгоритмов обработки текстов.

### Методические указания

Материал, изучаемый в § 9 учебника, призван скорее привлечь внимание учеников, чем дать конкретные знания.

ЭВМ (в отличие, например, от калькуляторов) могут обрабатывать текстовую и графическую информацию. В учебнике 9 класса уже рассматривались простейшие алгоритмы работы с изображениями. В § 9 учебника 10 класса рассматриваются самые простые операции со словами. Новизна этого материала для учеников состоит не в том, что эти операции чем-то необычны. Из курса математики ученики знают намного более сложные операции над числовыми величинами. Новизна состоит в том, что до сих пор в школе ученики не пользовались *о с о з н а н и* о никакими алгоритмами над текстами. Между тем даже в детских играх со словами содержится много действий, которые могут быть формализованы с помощью операций соединения, вырезки текста, обращения слов. Поэтому рекомендуем учителю в классе задавать вопросы, какие игры такого типа ученики помнят, и выбирать задачи для второго урока по этой теме с учетом названных учениками игр.

В § 9 рассматриваются самые простые операции над словами. Ученики должны не столько запомнить, что допустимы такие и только такие команды, но понять, как происходит в ЭВМ простейшая работа с текстами. Можно ученикам дополнительно рассказать, что в современных системах автоматизации труда канцелярии (некоторую информацию о та-



ких системах можно найти на 82-83 с. учебника) хранятся заготовки деловых писем, в которых оставлено место для некоторых имен (с указанием падежа, числа, рода). Достаточно ввести нужное имя и компьютер автоматически печатает полностью подготовленное письмо.

Нужно, чтобы ученики получили представление (хотя бы в самых общих чертах), как это делается.

Все же полезно заметить, что с помощью рассмотренных операций над словами можно осуществлять довольно сложные преобразования текстов. Интересно отметить, что почти все читатели при первом чтении серьезно недооценивают возможностей приведенных операций.

Дополнительную информацию о применении алгоритмов работы с литерными величинами читатель может найти в § 17 учебника (см. с. 82-83, 89-92).

Наконец, одно предупреждение учителя. На стр. 54 учебника рассматривается алгоритм обращения слова. Естественно, учитель будет демонстрировать на примерах конкретных слов, что означает эта операция. Ученики также будут с удовольствием приводить такие примеры. Все же учитель должен продумать все (1) предлагаемые примеры заранее. Дело в том, что в русском языке неожиданно много слов имеют неблагозвучные, а то и просто неприличные обращения.

Д о м а ш н е е з а д а н и е: к уроку 1 - § 9, № 3 § 9; к уроку 2 - № 12 § 9.

## ВОПРОСЫ И ОТВЕТЫ

1. Как применяется операция соединения (склеивания) к литерным величинам? Приведите примеры.

**Ответ.** Операция склеивания применяется для соединения двух текстов в один. Например, "a1" + "a2" = "a1a2".

2. Как применяется операция вырезки к литерным величинам? Приведите примеры.

**Ответ.** Операция вырезки применяется для получения фрагмента текста. Например, если A = "ответ", то A [4:5] = "ет", A [1:7] = "о", A [2:3] = "тв".

3. Как применяется команда частичного изменения значения литерной величины? Приведите примеры.

**Ответ.** Команды частичного изменения значения литерной величины позволяют заменить часть слова. Например, если A = "квант", то после выполнения команды A [4:5] := "рк" значением A станет слово "кварк". Новая часть слова должна иметь ту же длину, что и старое.

## УПРАЖНЕНИЯ И РЕШЕНИЯ

1. Пусть A = "компьютеризация". Найдите значения:

а) A [4:4] + A [2:2] + A [7:9] + A [15:15],

б) A [4:4] + A [9:11]

**Решение.**

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

к о м п ь ю т е р и з а ц и я

а) потеря

б) приз

2. Составьте алгоритмы подсчета суммарного числа букв "а" и букв "б" в данной литерной величине.

**Решение.**

алг цел число букв (лит A)

нач цел i

```

знач:=0
для i от 1 до длин(A)
нц
  выбор
    при A [i:i] = "а": знач:=знач+1
    при A [i:i] = "б": знач:=знач+1
  конец выбора
кц
кон

```

3. Составьте алгоритмы замены в какой-нибудь литерной величине всех букв "а" на буквы "б" и наоборот (при полной замене, например, из слова "баба" должно получиться слово "абаб").

Решение.

алг замена (лит x)

```

арг x
рез x
нач цел i
для i от 1 до длин(x)
нц
  выбор
    при x[i:i] = "а": x[i:i] := "б"
    при x[i:i] = "б": x[i:i] := "а"
  конец выбора
кц
кон

```

4. Составьте алгоритм, удваивающий каждую букву в заданном тексте (при этом, например, из слова "бейсик" должно получиться слово "ббейсикк").

Решение.

алг лит удвоение (лит x)

```

нач цел i
для i от 1 до длин(x)
нц
  знач:=знач + x[i:i] + x[i:i]
кц
кон

```

5. Составьте алгоритм, выясняющий, является ли данное слово "перевертышем" (так называются слова, читающиеся одинаково слева направо и справа налево, например: ТУТ, ПОТОП, КАЗАК).

Решения.

алг лит проверка (лит x)

```

нач лит y
y:=обращение(x)
если x=y
  то знач:="да"
  иначе знач:="нет"
все
кон

```

Пояснение. В решении использован алгоритм обращения приведенный в примере 2 учебного пособия.

Заметим, что слово является "перевертышем", если его первая буква совпадает с последней, вторая с предпоследней и т.д. Можно написать алгоритм такой проверки. Удобно использовать два индекса  $i$  и  $j$ , которые указывают на сравниваемые буквы. Можно организовать цикл с командой

алг проверка (лит x, y)

```

арг x
рез y
нач цел i, j
y:="да"
для i от 1 до длин(x)
нц
  j:=длин(x)-i+1
  если x[i:i] ≠ x[j:j]
    то y:="нет"
  все
кц
кон

```

Другой вариант, более быстрый:



```

нач цел i, j
у:="да", i:=1, j:=длин(x)
пока i=j и у="да"
  ни
  если x[i:i] ≠ x[j:j]
  то у:="нет"
  всё
  i:=i+1
  j:=j-1
кц
кон

```

6. Составьте алгоритм, вычеркивающий из данного слова все буквы "а" (так, чтобы, например, из слова "заноа" получилось слово "знос").

Решение

```

алг лит вычеркивание (лит x)
нач цел i
для i от 1 до длин(x)
  ни
  если x[i:i] ≠ "а"
  то рез:=рез+x[i:i]
  всё
кц
кон

```

7. Составьте алгоритм, который каждую встреченную в слове букву "б" заменял бы сочетанием букв "ку" (так, чтобы, например, из слова "абракадабра" получалось слово "акуракадакура").

Решение

```

алг б-ку (лит x, y)
  арг x
  рез y
нач цел i
для i от 1 до длин(x)

```

```

  ни
  если x[i:i] = "б"
  то y:=y+"ку"
  иначе y:=y+x[i:i]
  всё
кц
кон

```

8. С помощью операций вырезки и склеивания составьте из частей слова "интеграл" слова "гаители", "рентген", "тигр", "агент".

Решение.

A = "и н т е г р а л"  
 1 2 3 4 5 6 7 8

"гаители" = A [5:5] + A [7:7] + A [2:2] + A [3:4] + A [9:9] + A [1:1]

"рентген" = A [6:6] + A [4:4] + A [2:3] + A [5:5] + A [4:4] + A [2:2]

"тигр" = A [3:3] + A [1:1] + A [5:6]

"агент" = A [7:7] + A [5:5] + A [4:4] + A [2:3]

9. Напишите алгоритм, проверяющий, является ли частью данного слова "сок". Ответ должен быть "да" или "нет". (Например, для слов "сокол" и "кусок" ответ "да", а для слова "кокос" - "нет").

Решение.

```

алг лит сок (лит x)
нач цел i
  знач:="нет"
  для i от 1 до длин(x)-2
    ни
      если x [i:i+2] = "сок"
      то знач:="да"
      всё
    кц
  кон

```

Можно заметить, что если знач получил значение "да" то повторять цикла дальше нет необходимости. С помощью команды пока этот алгоритм можно записать так:

```

алг лит сок (лит x)
нач цел
  знач:="нет", i:=1
  пока i ≤ длин(x)-2 и знач="нет"
  ни
    если x[i:i+2] ="сок"
    то знач:="да"
  все
кн
кон

```

При использовании переменных в операции вырезки из литерной переменной A надо следить, чтобы значения переменных находились в интервале I и длин(A). Поэтому верхняя граница цикла установлена  $\text{длин}(x)-2$ . Посмотрим как будет работать алгоритм, если дано слово из трех букв. Цикл будет выполняться только один раз с  $i = 1$ .  $x[1:3]$  действительно все слово x. Если данное слово содержит один или две буквы, то цикл не будет выполнен ни одного раза.

10. Напишите алгоритм, подсчитывающий, сколько раз в данном слове встречается сочетание "со" (например, в слове "согласование" оно встречается 2 раза).

Решение.

```

алг цел со (лит x)
нач цел i
  знач:=0
  для i от 1 до длин(x)-1
  ни
    если x[i:i+1] ="со"
    то знач:=знач+1
  все
кн
кон

```

11. Напишите алгоритм, подсчитывающий, сколько раз в данном слове x встречается (в качестве его части) данное слово y. Примеры: если x="аттестат", y="т", результатом должно быть число 4; если x="абракадабра", y="бра" - результат равен 2; если x="забой", y="еж" - результат равен 0, если x="аааа", y="ааа", то результат равен 3.

Решение.

```

алг цел подслово (лит(x, y))
нач цел i
  знач:=0
  для i от 1 до длин(x) - длин(y) + 1
  ни
    если x[i:i+длин(y)-1] = y
    то знач:=знач+1
  все
кн
кон

```

Пояснение. Алгоритм подслово является обобщением алгоритма со. Проверка появления сочетания "со" заменено словом y. Наиболее трудное - правильно указать верхнюю границу цикла и значения параметров вырезки.

Рассмотрим например случай когда x и y равной длины. Тогда цикл должен выполняться I раз. В команде если получим  $x[1:1+\text{длин}(y)-1] = y$

Действительно все слово x задают X [1:длин(y)], так как  $\text{длин}(x) = \text{длин}(y)$ .

Если  $\text{длин}(x) < \text{длин}(y)$ , то цикл не будет выполнен ни раз и знач останется равным 0.

Можно записать границы цикла и по другому.

```

алг цел подслово(лит x, y)
нач цел i
  знач:=0
  для i от длин(y) до длин(x)
  ни
    если x[i+1-длин(y):i] = y
    то знач:=знач+1
  все
кн
кон

```



## ДОПОЛНИТЕЛЬНЫЕ УПРАЖНЕНИЯ И РЕШЕНИЯ

I. Составьте алгоритм, выясняющий можно ли данное слово  $y$  составить из частей данного слова  $x$ , если каждую букву слова  $x$  можно использовать

- несколько раз
- только один раз

Например, если  $y$  = "крик" и  $x$  = "кирпич", то в случае

- ответ "да" в случае б) ответ "нет".

Решения.

- алг лит проверка I (лит  $x, y$ )

```

нач цел i, j
лит z
знач := "да"
i := 1
пока знач = "да" и i ≤ длин(y)
ни
  z := "нет"
  j := 1
  пока z = "нет" и j ≤ длин(x)
  ни
    если y[i:i] = x[j:j]
    то z := "есть"
  все
  j := j + 1
кн
если z = "нет"
то знач := "нет"
все
i := i + 1
кн

```

кон

- алг лит проверка 2 (лит  $x, y$ )

```

нач цел i, j
лит z

```

знач := "да"

i := 1

пока знач = "да" и i ≤ длин(y)

ни

z := "нет"

j := 1

пока z = "нет" и j ≤ длин(x)

ни

если y[i:i] = x[j:j]

то z := "есть"

x := x [1:j-1] + x [j+1:длин(x)]

все

j := j + 1

кн

если z = "нет"

то знач := "нет"

все

i := i + 1

кн

кон

- Составьте алгоритм выясняющий состоит ли данное слово из двух одинаковых частей.

Например: БАБА, ДЯДЯ.

Решение.

- алг лит проверка 3 (лит  $x$ )

нач цел i, j

i := остаток(длин(x), 2)

если i = 1

то знач := "нет"

иначе j := длин(x)

2

знач := "да"

для i от 1 до j

ни





В настоящее время существует большое число языков программирования высокого уровня. Но так как вычислительные машины могут выполнять программы только на машинном языке, то любая программа, написанная на языке высокого уровня, должна переводиться на язык, понятный машине — язык машинных команд. Эта работа поручена самой машине и выполняется программой, которая называется транслятором.

Прежде, чем перейти к разбору отдельных параграфов данного раздела, изложим общие правила записи программы на РАПИРЕ.

Текст программы на языке РАПИРА состоит из отдельных предложений, называемых предписаниями. Предписание — это аналог команды в алгоритмическом языке.

При написании программы используют следующие основные предписания РАПИРА:

присваивание;  
ветвление;  
цикл;  
ввод;  
вывод;  
вызов.

На одной строке можно задавать несколько предписаний. Разделяются они точкой с запятой, например:

X:=2; Y:=2

В языке РАПИРА так же как и в алгоритмическом языке используются служебные слова, которые имеют строго определенный смысл и назначение. В языках программирования такие слова называются ключевыми. Составляющий программу должен знать все ключевые слова и их употребление.

При записи текста программы все знаки записываются в одну строку, использование подстрочных и надстрочных индексов не допускается ввиду фиксированного набора символов, определенного возможностями техники.

Например:

X<sub>1</sub> записывается как X1,

B<sup>2</sup> записывается как B\*2.

Основу любого языка составляет алфавит. Алфавит языка РАПИРА — это русские и латинские заглавные буквы, цифры, специальные знаки, такие, как:

+ - \* / . , : ; ( ) > < " % ? ] [ и другие.

Имена в РАПИРЕ (имена переменных, процедур, функций и т.д.) выбираются программистом произвольно, но рекомендуется использовать такие имена, которые отражают смысл, назначение переменной величины, процедуры или функции.

Имя в РАПИРЕ состоит из букв и цифр, и знаков подчеркивания (  ), причем на первом месте должна стоять буква. Допускается использование только заглавных букв. В начале и конце имени знак подчеркивания указывать нельзя! Обычно знак подчеркивания используется в именах, состоящих из нескольких смысловых частей.

Примеры имен:

X

X12

ОЦЕНКА\_ОТЛИЧНО

Примеры ошибочных имен:

6AB (имя не может начинаться с цифры)

\_ОБЪЕКТ (знак подчеркивания не может стоять в начале имени)

Скорость (строчные буквы не допустимы)

При написании программы элементы текста (ключевые слова, имена, константы) отделяются друг от друга разделителями. Разделителями могут быть знаки операции, запятая, двоеточие, точка с запятой, скобки, знак равенства, пробел. Пробел (пропуск) как разделитель элементов текста может свободно использоваться в программе. Но внутри ключевого слова, имени, числа использовать пробелы запрещено. Пробел в языке РАПИРА считается специальным символом (позиция, которая соответствует этому символу, остается свободной).

Над числами в РАПИРЕ так же как и в алгоритмическом языке определены арифметические операции и операции сравнения, для обозначения которых используются знаки:

- + - сложение;
- - вычитание;
- \* - умножение;
- / - деление;
- \*\* - возведение в степень.

Еще раз обращаем ваше внимание на то, что  $A^2$  на языке РАПИРА будет записано как  $A**2$ .

Операции сравнения:

- = - равно;
- /= - не равно (вместо  $\neq$  в алгоритмическом языке);
- > - больше;
- < - меньше;
- <= - меньше или равно;
- >= - больше или равно;

Все эти операции имеют обычный математический смысл.

При записи дробных чисел для отделения целой части от дробной используется точка (вместо запятой в алгоритмическом языке).

Например, число 12,58 записывается как 12.58.

Это можно объяснить тем, что первые языки программирования возникли в англоязычных странах, где принято использовать в качестве разделителя целой и дробной частей числа десятичную точку. Эта традиция поддерживается во всех языках программирования.

Никакой знак операции не может быть опущен! Выражение  $4ac$  на языке РАПИРА должно быть записано как  $4*A*C$ . Запись  $4AC$  ошибочна!

Каждая программа должна быть записана в точном соответствии с правилами, принятыми для записи программы на конкретном языке программирования. Всякое отступление от этих правил квалифицируется как ошибка. Как вы помните, при записи алгоритма с использованием алгоритмического языка не выдвигалось требование строго следовать правилам оформления текста алгоритма. И в этом одно из главных отличий языка программирования и алгоритмического языка.

## § 10. ЗАПИСЬ АЛГОРИТМОВ В ВИДЕ ПРОЦЕДУР НА РАПИРЕ (I ч.)

Основная цель. Достичь понимания учениками необходимости в специальных языках программирования. Дать ученикам первое представление о языке программирования Рапира и продемонстрировать способы реализации некоторых конструкций алгоритмического языка средствами Рапиры.

Требования к знаниям и умениям. Ученики должны знать основные отличия записи алгоритмов на Рапире от записи алгоритмов на алгоритмическом языке.

### Методические указания

Процедура в языке РАПИРА определяется как программа или часть программы, имеющая самостоятельное имя - имя процедуры.

Разберем особенности записи алгоритмов в виде процедур на РАПИРЕ на примере алгоритма решения квадратного уравнения, приведенного в учебном пособии.

В последовательности предписаний, составляющих программу на языке РАПИРА, первым должен быть заголовок программы, имеющий вид:

ПРОЦ имя-процедуры (список параметров)

В заголовке

ПРОЦ РЕШЕНИЕ\_КВАДРАТНОГО\_УРАВНЕНИЯ(=>A,=>B,=>C,  
X1=>,X2=>,Y=>)

именем процедуры является имя

РЕШЕНИЕ\_КВАДРАТНОГО\_УРАВНЕНИЯ

и список параметров состоит из имен A, B, C, X1, X2, Y. Все имена записаны в соответствии с изложенными правилами записи имен на РАПИРЕ.

Имена в списке параметров отделяются друг от друга запятыми. Рядом с именем ставится стрелка (стрелка - это комбинация двух символов: "=" и ">", ее назначение будет пояснено ниже).



Таким образом, при записи алгоритма с использованием алгоритмического языка в заголовке указывается служебное слово алг, в заголовке программы на РАПИРЕ указывается ключевое слово ПРОЦ.

При описании алгоритма отдельно указываются исходные данные для алгоритма - аргументы и результаты алгоритма.

В процедуре на языке РАПИРА аргументы и результаты отдельно не описываются. Список параметров уже содержит имена аргументов и результатов. В списке параметров перед параметром-аргументом (его называют входным параметром) ставится стрелка "=>", для результата стрелка "=>" ставится после имени параметра-результата (это выходной параметр). Если какой-то параметр является и аргументом и результатом, то стрелка ставится до и после него. Например:

=>ТАБЛИЦА=>

Описание промежуточных переменных, используемых для обозначения вычисляемых промежуточных величин, в алгоритмическом языке приводится после служебного слова нач, при этом необходимо указать и тип такой переменной. В языке РАПИРА описание промежуточных (локальных) переменных внутри процедуры состоит из ключевого слова ИМЕНА, за которым через двоеточие перечисляются имена локальных переменных. Имена отделяются друг от друга запятыми.

Примеры:

ИМЕНА: А, В, С

ИМЕНА: D

Обратите внимание на то, что заголовок этой конструкции дается во множественном числе, даже если имя одно.

И еще одна особенность РАПИРЫ: при описании переменных не указываются типы переменных. Дело в том, что любой переменной можно присвоить значение любого типа. Кроме того, одной и той же переменной можно присваивать значения различных типов, например:

X:=1

X:=2.56

X:="КРОКОДИЛ".

Таким образом, имена в РАПИРЕ не типизированы, т.е. тип имеют не переменные, а их значения.

Д о м а ш н е е   з а д а н и е : §10, №5 §10.

## ВОПРОСЫ И ОТВЕТЫ

1. Приведите примеры различий между алгоритмическим языком и языком Рапира. Чем объясняются эти различия?

Ответ. Имеются несколько различий между алгоритмическим языком и языком Рапира. Назовем некоторые из них:

- произвольный алфавит для записи конструкций алгоритмического языка и строго фиксированный алфавит языка Рапира, обусловленный ограниченностью клавиатуры машины;
- в отличие от алгоритмического языка при образовании составных имен на Рапире обязательно использование символов подчеркивания, так как пробел воспринимается машиной как признак конца имени;
- в отличие от алгоритмического языка при описании переменных на Рапире не указываются типы переменных, так как на Рапире позволено любой переменной присвоить значение любого типа.

2. Как указываются аргументы и результаты алгоритма в записи на алгоритмическом языке и на Рапире?

Ответ. Аргументы и результаты на алгоритмическом языке перечисляются после ключевых слов арг и рез соответственно. На Рапире в перечислении величин в заголовке процедуры перед аргументами и после результатов ставится стрелка = > .

## УПРАЖНЕНИЯ И РЕШЕНИЯ

1. Запишите выражение

$$b^3 + \sqrt{c} - \sqrt[3]{a}$$

на языке РАПИРА.

Решение

$$B := 3 + \text{SQRT}(C) - A ** (1/3)$$

2. Заголовок алгоритма, записанного на РАПИРЕ:

ПРОЦ ПРИМЕР(=&gt;A, B=&gt;, =&gt;C=&gt;).

Укажите аргументы и результаты этого алгоритма.

Решение

Аргументом (входным параметром) этой процедуры является переменная A: указана стрелка перед именем A. Результатом (выходным параметром) - переменная B: стрелка указана после имени B. Переменная C является аргументом и результатом: стрелка указана перед именем C и после имени C.

3. Арифметическое выражение имеет вид:

$$\frac{1024 \cdot 2^5 + 128 \cdot 4^2}{(16 - 2 \cdot 4) - 2^2}$$

Запишите на РАПИРЕ команду, которая вычисляет значение этого выражения и присваивает его переменной с именем ВЕЛИЧИНА.

Решение

$$\text{ВЕЛИЧИНА} := (1024/2 ** 5 + 128/4 ** 2) / ((16 - 2 * 4) - 2 ** 2)$$

4. Запишите на РАПИРЕ команду, которая вычисляет величину  $h$  по формуле:

$$h = h_0 + \frac{9,81 \cdot t^2}{2}$$

Решение

$$H := H_0 + 9,81 \cdot T ** 2 / 2 \quad \text{или}$$

$$H := H_0 + 9,81 \cdot T \cdot T / 2$$

5. Составьте описание процедуры РАССТОЯНИЕ, которая имеет в качестве аргументов координаты двух точек:  $(X1, Y1)$ ,  $(X2, Y2)$  и вычисляет расстояние между этими точками.

Решение

ПРОЦ РАССТОЯНИЕ(=&gt;X1, =&gt;Y1, =&gt;X2, =&gt;Y2, R=&gt;)

НАЧ

$$R := \text{SQRT}((X2 - X1) ** 2 + (Y2 - Y1) ** 2)$$

КОН

6. Напишите процедуру с заголовком ПРОЦ ПРОВЕРКА(=>P, =>Q, =>R, Z=>), где P, Q, R - коэффициенты квадратного уравнения, а Z - литерная переменная, принимающая одно из трех значений

"КОРНИ ОДНОГО ЗНАКА"

"КОРНИ ПРОТИВОПОЛОЖНЫХ ЗНАКОВ"

"Вещественных корней нет"

Предусмотрите вызов процедуры РЕШЕНИЕ\_КВАДРАТНОГО\_УРАВНЕНИЯ внутри описания процедуры ПРОВЕРКА.

Процедура ПРОВЕРКА должна вычислять корни, и если они существуют, выяснять совпадение или несовпадение их знаков.

Решение

ПРОЦ ПРОВЕРКА(=&gt;P, =&gt;Q, =&gt;R, Z=&gt;)

НАЧ ИМЕНА X1, X2, Y

РЕШЕНИЕ\_КВАДРАТНОГО\_УРАВНЕНИЯ(P, Q, R, X1, X2, Y)

ЕСЛИ Y = "НЕТ РЕШЕНИЯ"

ТО Z := "Вещественных корней нет"

ИНАЧЕ

ЕСЛИ R/P &gt; 0

ТО Z := "КОРНИ ОДНОГО ЗНАКА"

ИНАЧЕ Z := "КОРНИ ПРОТИВОПОЛОЖНЫХ ЗНАКОВ"

ВСЕ

ВСЕ

КОН



## § II. ЗАПИСЬ АЛГОРИТМОВ ВЫЧИСЛЕНИЯ ЗНАЧЕНИЙ ФУНКЦИИ НА РАПИРЕ (I ч.)

Основная цель. Ознакомить учеников со способом реализации на Рапире функций и показать возможность снабжения текста программы комментариями.

Требования к знаниям и умениям. Ученики должны знать правила записи на Рапире алгоритмов вычисления значений функций. Ученики должны уметь записывать на Рапире простейшие алгоритмы вычисления значений функций.

### Методические указания

Функция как и процедура – это программа или часть программы, имеющая самостоятельное имя.

Основная особенность функции состоит в том, что функция после выполнения списка предписаний выдаёт единственный результат. Это делается с помощью конструкции вида:

ЗНАЧ:= выражение

Разберём запись алгоритма нахождения наибольшего общего делителя двух чисел в виде функции на языке РАПИРА. Текст функции приведён на стр.61 учебного пособия.

Особенности записи:

1. В заголовке функции используется ключевое слово ФУНК вместо служебного слова алг в алгоритмическом языке.

2. В заголовке функции указывается имя функции со списком параметров, заключённым в круглые скобки. Тип результата в заголовке функции не указывается, тогда как в заголовке алгоритма после служебного слова алг указывается тип значения функции (в рассматриваемом примере нат).

3. Список параметров функции содержит только имена переменных, типы переменных так же как и для процедуры не задаются. И так как все параметры функции являются аргументами, т.е. входными параметрами, то нет необходимости специально указывать на это, поэтому стрелки перед параметрами не ставятся.

4. Результат выполнения предписаний внутри функции описывается конструкцией.

ЗНАЧ:=X

Но в отличие от алгоритмического языка тип результата заранее не описывается и поэтому результатом функции может быть величина любого типа.

Алгоритм нахождения наибольшего общего делителя двух чисел может быть оформлен как процедура или как функция. Но так как в результате выполнения этого алгоритма должен быть получен единственный результат, то удобно записать этот алгоритм в виде функции.

Рассмотрим, в чём состоит принципиальное отличие процедуры и функции на РАПИРЕ.

Над процедурами и функциями выполняется единственная операция – вызов. Вызов – это исполнение всех предписаний процедуры или функции. Вызов процедуры – это отдельное предписание, которое состоит из имени процедуры, за которым в круглых скобках указываются фактические параметры, т.е. конкретные значения, которые будут переданы в процедуру или конкретные переменные, которым будет присвоен результат.

Пример вызова процедуры РЕШЕНИЕ-КВАДРАТНОГО-УРАВНЕНИЯ для нахождения корней уравнения.

$$x^2 - 5x + 6 = 0$$

РЕШЕНИЕ-КВАДРАТНОГО-УРАВНЕНИЯ (1,5,6,X1,X2,Y).

При вызове процедуры

РЕШЕНИЕ-КВАДРАТНОГО-УРАВНЕНИЯ выполняются следующие действия:

1. Осуществляется передача фактических параметров в процедуру, т.е.

входным параметрам процедуры

A,B,C присваиваются значения соответствующих фактических параметров.

A получает значение 1,

B –"– значение 5,

C –"– значение 6.

2. Исполняются все предписания процедуры в порядке их следования.

3. Вычисленные значения присваиваются соответствующим фактическим параметрам, т.е. переменные X1, X2 получают значения действительных корней уравнения, а значением переменной Y становится текст.

Вызов функции по своему значению — это операция, а не предписание, как вызов процедуры. Поэтому вызов функции может стоять везде, где допустимо выражение.

Поясним это на примере вызова функции, реализующей алгоритм нахождения наибольшего общего делителя двух чисел для значений 18 и 108. Вызов функции НОД (18, 108) задается в выражении. При вызове функции НОД осуществляется следующее.

1. Передача фактических параметров, т.е. параметру M присваивается значение 18, параметру N — присваивается значение 108.

2. Исполняются все предписания функции в порядке их следования.

3. Вычисляется значение результата, т.е. выражения записанного после

"ЗНАЧ:="

4. Вызов функции заменяется значением результата.

Таким образом, если выполняется, например, предписание  $A := \text{НОД}(18, 108) + 5$ , то результат вызова функции НОД — значение 18, и в итоге переменная A получит значение  $18 + 5 = 23$ .

Следовательно, процедура и функция отличаются — способом вызова — способом передачи результата.

Прежде чем перейти к следующему параграфу раздела рассмотрим возможности работы с литерными величинами. В PASCAL так же как и в алгоритмическом языке предусматривается работа с текстами.

Текст — это произвольная последовательность символов. Текст может включать буквы, цифры, специальные символы, в том числе пробел. При записи в программе текст заключается в кавычки.

Примеры:

"ДАННЫЕ"

"ИВАНОВ И.И."

"ТАБЛИЦА 1"

"СЕНТЯБРЬ 1985 ГОДА"

Символы в тексте нумеруются в направлении слева направо. Нумерация начинается с единицы. Над текстами так же, как и над числовыми переменными, могут выполняться операции. Рассмотрим операции.

1. Определение длины текста.

Обозначается операция словом ДЛИН. Например, после выполнения предписаний

$X := \text{"АБРАКАДАБРА"}$

$D := \text{ДЛИН}(X)$

переменная D получит значение 11.

2. Слияние, соединение текстов. Обозначается операция знаком "+". Например, после выполнения предписания

$X := \text{"ВЕЛО"} + \text{"ТОНКА"}$

X получит значение "ВЕЛОТОНКА". Длина литерной переменной X будет равна 9.

3. Выборка — извлечение символа по его номеру в тексте. Обозначается операция квадратными скобками, в которых указывается номер символа. Например, если

$X := \text{"ВЕЛОТОНКА"}$

то  $X[5]$  — выборка 5-го символа из текста X. Следовательно, результат выборки — это один символ. Номер символа может задаваться и выражением с целыми положительными значениями, не превосходящими длину текста.

В результате выполнения предписания

$Y := X[N-1] + X[N-2] + X[N-3]$

переменная Y получит значение "ГОЛ" при  $N=6$ , и значение "ЛЕВ" при  $N=4$ .

4. Вырезка — извлечение части текста.

Обозначается операция квадратными скобками, в которых указывается два номера, разделяемые двоеточием, номер начального и номер конечного символов.



Например, если

$Y := X[3 : 6]$ , то результатом будет текст

"ЛОГО" - название языка программирования.

Замечание: если в состав текста необходимо включить кавычку, то её нужно повторить дважды, например,

"КОМАНДА ""СПАРТАК"" "

обратите внимание на то, что последняя кавычка - это кавычка, окаймляющая текст.

Д о м а ш н е е з а д а н и е : §II, №4а §II.

### ВОПРОСЫ И ОТВЕТЫ

1. Как записываются алгоритмы вычисления значений функций на языке РАПИРА? Приведите пример.

Ответ.

При записи алгоритма вычисления значений функций на языке РАПИРА последовательность предписаний, реализующих алгоритм оформляется как функция.

Покажем это на примере алгоритма "прямого" вычисления значения многочлена  $2x^3 - 3x + 5$ .

ФУНК МНОГОЧЛЕН (X)

НАЧ

ЗНАЧ:=2·X·X·3-3·X+5

КОН

В заголовке функции записывается ключевое слово ФУНК, имя функции - МНОГОЧЛЕН и в скобках единственный параметр. Тип значения функции не задаётся. Так как для функции все параметры являются аргументами, то стрелка перед именем не ставится.

Результат, возвращаемый функцией - это значение выражения, записанного после "ЗНАЧ:=". Для функции это единственный способ задать возвращаемое значение.

2. Почему возникает необходимость использовать в языке РАПИРА два слова ФУНК и ПРСЦ вместо одного слова алг в алгоритмическом языке?

Ответ.

В алгоритмическом языке запись алгоритмов вычисления значений функций отличается тем, что в заголовке алгорит-

ма после служебного слова алг указывается тип вычисляемого значения функции. В РАПИРЕ нет специального указания на тип вычисляемого значения функции. Как уже отмечалось в РАПИРЕ имена не типизированы. Алгоритм вычисления значений функции оформляется в виде функции, в заголовке которой записывается ключевое слово ФУНК.

### УПРАЖНЕНИЯ И РЕШЕНИЯ

1. Запишите на языке РАПИРА алгоритм вычисления функции  $\text{MAX2}(M, N)$ , значение которой равно большему из чисел  $M$  и  $N$ . Заголовок должен быть таким:

ФУНК MAX2(M, N)

Решение.

ФУНК MAX2(M, N)

НАЧ (\* НАХОЖДЕНИЕ БОЛЬШЕГО ИЗ ДВУХ ЧИСЕЛ \*)

ЕСЛИ  $M > N$

ТО ЗНАЧ:=M

ИНАЧЕ ЗНАЧ:=N

ВСЯ

КОН

2. Используя алгоритм предыдущего упражнения как вспомогательный, запишите алгоритм вычисления функции  $\text{MAX3}(M, N, S)$ , значение которой равно большему из трёх чисел  $M, N$  и  $S$ .

Решение

ФУНК MAX3(M, N, S)

НАЧ (\* НАХОЖДЕНИЕ БОЛЬШЕГО ИЗ ТРЕХ ЧИСЕЛ \*)

ЗНАЧ:=MAX2(MAX2(M, N), S)

КОН

В этом упражнении нахождение максимального из трёх чисел осуществляется в форме двукратного обращения к вспомогательному алгоритму MAX2.

3. Спишите алгоритм вычисления функции, значение которой равно числу букв "А" в заданном тексте СЛОВО.

Решение.

```

ФУНК ЧИСЛО_БУКВ_А(СЛОВО)
НАЧ ИМЕНА: I
ЗНАЧ:=0
ДЛЯ I ОТ 1 ДО ДЛИН(СЛОВО)
НЦ
    ЕСЛИ СЛОВО [ I ]="А"
        ТО ЗНАЧ:=ЗНАЧ+ 1
ВСЕ
КЦ
КОН

```

4. Расширьте возможности алгоритма предыдущего задания так, чтобы функция могла вычислять:

- число вхождений любой буквы, заданной в качестве аргумента алгоритма, в тексте СЛОВО;
- процентный состав задаваемой буквы в тексте СЛОВО.

Решение.

```

а) ФУНК ЧИСЛО_БУКВ(А,СЛОВО)
НАЧ ИМЕНА: I
ЗНАЧ:=0
ДЛЯ I ОТ 1 ДО ДЛИН(СЛОВО)
НЦ
    ЕСЛИ СЛОВО [ I ]= А
        ТО ЗНАЧ:=ЗНАЧ + 1
ВСЕ
КЦ

```

```

б) ФУНК ПРОЦЕНТНЫЙ_СОСТАВ(А,СЛОВО)
НАЧ
    ЗНАЧ:=(ЧИСЛО_БУКВ(А,СЛОВО)/ДЛИН(СЛОВО)) * 100
КОН

```

5. Спишите функцию ДЕНЬ\_НЕДЕЛИ, которая по аргументу ДЕНЬ литерной величине, принимающей значения "ПОНЕДЕЛЬНИК", "ВТОРНИК", ..., "ВОСКРЕСЕНЬЕ" вычисляет номер дня недели - целое число 1, 2, ..., 7 соответственно. Воспользуйтесь командой ВЛЕСР.

Решение.

```

ФУНК ДЕНЬ_НЕДЕЛИ (ДЕНЬ)
НАЧ
    ВЫБОР
        ПРИ ДЕНЬ="ПОНЕДЕЛЬНИК":ЗНАЧ:=1
        ПРИ ДЕНЬ="ВТОРНИК":ЗНАЧ:=2
        ПРИ ДЕНЬ="СРЕДА":ЗНАЧ:=3
        ПРИ ДЕНЬ="ЧЕТВЕРГ":ЗНАЧ:=4
        ПРИ ДЕНЬ="ПЯТНИЦА":ЗНАЧ:=5
        ПРИ ДЕНЬ="СУББОТА":ЗНАЧ:=6
        ИНАЧЕ ЗНАЧ:=7
    КОНЕЦ ВЫБОРА
КОН

```

## § 12. КОРТЕЖИ НА РАПИРЕ (I ч.)

Основная цель. Дать ученикам понятие о реализации табличных величин на Рапире и ознакомить их с обработкой данных сравнительно сложной структуры.

Требования к знаниям и умениям. Ученики должны:

Знать, что такое кортеж в Рапире, и понять, что его элементы могут принимать значения различных типов.

Уметь при помощи угловых скобок задавать кортежи различной сложности, в том числе прямоугольные таблицы и, например, списки класса, подобные разобранным в учебном пособии.

Уметь пользоваться функцией ДЛИН и обращаться к отдельным элементам кортежа, с целью извлечения и обработки информации.

Уметь пользоваться операцией вырезки и соединения.



### Методические указания

Кортеж в РАПИРЕ определяется как упорядоченная совокупность (упорядоченное множество) произвольных элементов. Элементы кортежа нумеруются, начиная с единицы.

По правилам РАПИРЫ элементы явно задаваемого кортежа заключаются в угловые скобки  $\langle \rangle$  и внутри скобок разделяются запятыми.

Например,

$K := \langle 10, 20, 30, 40 \rangle$

После выполнения этого предписания значением переменной  $K$  станет кортеж из 4-х элементов: чисел 10, 20, 30, 40.

Операции над кортежами.

Над кортежами в РАПИРЕ определены такие же операции, как и над текстами.

1. Определение длины кортежа. Обозначается операция ключевым словом ДЛИН. Так, если выполнено предписание

$D := \text{ДЛИН}(K)$ ,

то значением переменной  $D$  будет число 4 (4 Элемента).

2. Соединение. Обозначается операция знаком "+". После выполнения предписания

$K := \langle 10, 20, 30, 40 \rangle + \langle 50, 60 \rangle$

значением переменной  $K$  станет кортеж из 6-ти элементов:

$\langle 10, 20, 30, 40, 50, 60 \rangle$

3. Выборка - извлечение элемента кортежа. Обозначается операция квадратными скобками, в которых указывается номер элемента кортежа. После выполнения предписания

$K := \langle 10, 20, 30, 40 \rangle$

$\text{ВЫБОРКА} := K[2]$

значением переменной  $\text{ВЫБОРКА}$  будет число 20.

Результатом выборки является заданный элемент кортежа.

4. Вырезка. Обозначается операция квадратными скобками,  $[N1:N2]$ , где

$N1$  - номер начального элемента вырезки

$N2$  - номер конечного элемента вырезки.

Если выполнено предписание

$\text{ВЫРЕЗКА} := K[2:3]$ , то значением имени  $\text{ВЫРЕЗКА}$  будет

кортеж  $\langle 20, 30 \rangle$ .

Для кортежей результатом вырезки всегда будет кортеж.

Ограничиваться явным заданием кортежей при помощи угловых скобок в реальных программах совершенно не достаточно. Так, если нам необходим кортеж, содержащий 1000 первых подряд идущих натуральных чисел, то можно воспользоваться многократным повторением соединения:

$K := \langle 1 \rangle$

ДЛЯ I ОТ 2 ДО 1000

нц

$K := K + \langle I \rangle$

кц

Двумерную таблицу  $T[1:100, 1:1000]$ , содержащую нули, можно создать следующим образом.

$T := \langle \langle 0 \rangle \rangle$

ДЛЯ I ОТ 2 ДО 100

нц

$T := T + \langle \langle 0 \rangle \rangle$

кц

ДЛЯ I ОТ 1 ДО 100

нц

ДЛЯ J ОТ 2 ДО 1000

нц

$T[I] := T[I] + \langle 0 \rangle$

кц

кц

Однако, предпочтительнее обращаться к специальным средствам формирования кортежей, которыми, как правило, снабжены машины, выполняющие программы, написанные на языке Рапира.

В учебном пособии сказано: "табличные величины называются в Рапире кортежами". Однако, это определение не раскрывает все богатство конструкции "кортеж". В алгоритмическом языке описание табличной величины содержит указание на тип элементов таблицы. Таким образом, таблица определяется как совокупность элементов одного типа.

Элементами одного и того же кортежа могут быть величины различных типов: числа, тексты, а также другие кортежи.

Именно эта возможность продемонстрирована примером учебного пособия. Список класса призван иллюстрировать применение вычислительной техники в обработке данных достаточно сложной структуры. Такого рода данные характерны для планово-управленческих задач, возникающих в т.н. автоматизированных системах управления (АСУ – общепринятое сокращение этого понятия).

Имея вычислительную машину, и сам учитель может в аналогичном списке своего класса накапливать реальную информацию об успеваемости своих учеников, а также множество других данных. Процедуры учебника дают примеры извлечения из таких данных обобщающих сведений для отчетов.

Домашнее задание: § 12, § 3 § 12.

#### ВОПРОСЫ И ОТВЕТЫ

1. Что соответствует в РАПИРЕ имеющимся в алгоритмическом языке табличным величинам?

Ответ.

Табличные величины, имеющиеся в алгоритмическом языке, могут быть описаны в языке РАПИРА с помощью кортежей.

2. Могут ли быть элементами кортежа  
а) литерные величины?

Ответ.

Элементами кортежа могут быть величины различных типов, в том числе и литерные величины.

б) кортежи?

Ответ.

Элементами кортежа могут быть и кортежи.

3. Какие операции над кортежами вам известны?

Ответ.

Над кортежами выполняются те же операции, что и над текстами:

определение длины кортежа,  
слияние,  
выборка,  
вырезка.

#### УПРАЖНЕНИЯ И РЕШЕНИЯ

1. Составьте алгоритм вычисления произведения всех элементов кортежа. Его заголовок должен иметь вид:  
ФУНК ПРОИЗВЕДЕНИЕ(X)

(ПРОИЗВЕДЕНИЕ ВСЕХ ЭЛЕМЕНТОВ КОРТЕЖА X.)

Решение

ФУНК ПРОИЗВЕДЕНИЕ(X)

(ПРОИЗВЕДЕНИЕ ВСЕХ ЭЛЕМЕНТОВ КОРТЕЖА X.)

НАЧ ИМЕНА: I

ЗНАЧ:=I

ДЛЯ I ОТ 1 ДО ДЛИН(X)

НЦ

ЗНАЧ:=ЗНАЧ \* X[I]

КЦ

КОН

2. Чему равно A[3], если  
A=<"КОТ", "СЛОН", "ТИГР", "КРОЛИК">?

Решение

A[3]="ТИГР"

3. Составьте процедуру, определяющую по списку класса (в форме кортежа) число учеников, у которых отметки по математике ниже отметки по информатике.



Решение

```

ПРОЦ ОТМЕТКА (=>СПИСОК, ЧИСЛО=>)
НАЧ ИМЕНА:НОМЕР
    ЧИСЛО:=0
    ДЛЯ НОМЕР ОТ 1 ДО ДЛИН(СПИСОК)
        НЦ
            ЕСЛИ СПИСОК[НОМЕР][3]<СПИСОК[НОМЕР][4]
                ТО ЧИСЛО:=ЧИСЛО+1
        ВСЕ
    КЦ
КОН

```

4. Элементами кортежа ЧИСЛА являются числовые величины. Найдите наименьший из элементов кортежа.

Решение

```

ПРОЦ МИН (=>ЧИСЛА, МИНИМАЛЬНЫЙ=>)
НАЧ ИМЕНА:К
    МИНИМАЛЬНЫЙ:=ЧИСЛА[1]
    ДЛЯ К ОТ 2 ДО ДЛИН(ЧИСЛА)
        НЦ
            ЕСЛИ МИНИМАЛЬНЫЙ>ЧИСЛА[К]
                ТО МИНИМАЛЬНЫЙ:=ЧИСЛА[К]
        ВСЕ
    КЦ
КОН

```

5. В условии предыдущей задачи найти номер наименьшего элемента.

Решение

```

ПРОЦ МИН-НОМЕР (=>ЧИСЛА, НОМЕР=>)
НАЧ ИМЕНА:К
    НОМЕР:=1
    ДЛЯ К ОТ 2 ДО ДЛИН(ЧИСЛА)
        НЦ
            ЕСЛИ ЧИСЛА[НОМЕР]>ЧИСЛА[К]
                ТО НОМЕР:=К
        ВСЕ
    КЦ
КОН

```

6. Кортеж КЛАСС состоит из кортежей пар. Первый элемент пары - литерная величина, значением которой является фамилия школьника, а второй - литерная величина, принимающая значение "А" для мальчиков и "Б" для девочек. Опишите процедуру, которая сосчитает процентный состав девочек и мальчиков в классе.

```

ПРОЦ СОСТАВ_КЛАСС (=>КЛАСС, ПРОЦЕНТ1=>, ПРОЦЕНТ2=>)
НАЧ ИМЕНА:НОМЕР, ЧИСЛО
    ЧИСЛО:=0
    ДЛЯ НОМЕР ОТ 1 ДО ДЛИН(КЛАСС)
        НЦ
            ЕСЛИ КЛАСС[НОМЕР][2]="А"
                ТО ЧИСЛО:=ЧИСЛО+1
        ВСЕ
    КЦ
    ПРОЦЕНТ1:=ЧИСЛО/ДЛИН(КЛАСС)*100
    ПРОЦЕНТ2:=100-ПРОЦЕНТ1
КОН

```

7. Шахматную доску можно представить в виде кортежа кортежей. Кортёж ДОСКА состоит из 8 кортежей-горизонталей. Каждый кортеж - горизонталь, в свою очередь состоит из 8 элементов-полей.

Напишите процедуру ЛАДЬЯ, аргументами которой являются два целых числа  $M$  и  $N$ , между 1 и 8 - номер горизонтали и номер вертикали поля. Эта процедура должна "пометить" все поле, побиваемое ладьей, стоящей на поле  $(M, N)$ . "Пометить" поле - это значит присвоить единицу соответствующему элементу-полю. В исходном состоянии все элементы -поля содержат нулевые значения.

#### Решение

ПРОЦ ЛАДЬЯ ( $\Rightarrow M, \Rightarrow N, \Rightarrow ДОСКА = \gamma$ )

НАЧ ИМЕНА: I

ДЛЯ I ОТ 1 ДО 8

НЦ

ДОСКА[I][N] := 1

ДОСКА[M][I] := 1

КЦ

ДОСКА[M][N] := 0

КОН

### § 13. КОМАНДЫ ВВОДА И ВЫВОДА НА РАИРЕ (I ч.)

**Основная цель.** Дать ученикам представление о способах обмена информацией между машиной и человеком.

**Требования к знаниям и умениям.** Ученики должны знать назначение и форму записи команд ВВОД, ВВОД ДАННЫХ, ВЫВОД и знать, что при выполнении команд ВВОД, ВВОД ДАННЫХ машина приостанавливает работу программы до завершения человеком ввода информации. Ученики должны уметь пользоваться командами ВВОД, ВВОД ДАННЫХ и ВЫВОД в простых программах.

#### Методические указания

Для вычислительных машин существует большое разнообразие технических средств ввода-вывода информации. В учебном пособии рассматривается лишь самое простое из них, тем не менее самое распространенное как среди малых, так и больших компьютеров. Рассмотренная техника ввода-вывода при помощи клавиатуры и экрана предоставляет человеку все удобства в непосредственном общении с машиной. Недаром клавиатура и экран стали неотъемлемой принадлежностью всех современных вычислительных машин и являются даже доминирующей чертой в их внешнем облике.

Ввод и вывод - важнейшие операции, без которых работа компьютера практически лишена смысла. Операция ввода иногда может отсутствовать, но тогда ввод присутствует неявно - исходные данные заложены непосредственно в текст программы, например, в виде констант. Операция вывода совершенно необходима, ибо без нее человек так и не узнает чем же кончились вычисления.

Встает законный вопрос - каким же образом мы до сих пор обходились без упомянутых операций? Дело в том, что изучалось только внутреннее строение алгоритмов вычислений, которое, разумеется, является определяющим. Но по сути "ввод-вывод" уже присутствовал в виде обмена инфор-



мацией между процедурами и функциями при помощи механизмов передачи параметров (как аргументов, так и результатов) и возвращения значения функций. При помощи этих средств формируются промежуточные результаты вычислений. Непосредственно ввод-вывод необходим, как правило, соответственно в начале и в конце счета.

Предложенные в учебном пособии команды Рапиры ВВОД, ВВОД ДАННЫХ, ВЫВОД являются достаточно мощными. При их помощи можно не только просто ввести с клавиатуры исходные данные и вывести на экран результаты работы программы (процедура НАИБОЛЬШИЙ\_ОБЩИЙ\_ДЕЛИТЕЛЬ на стр.68), но даже организовать диалог между человеком и машиной. В диалоге указанную процедуру учебника можно использовать, например, следующим образом.

```

ПРОЦ НАИБОЛЬШИЙ_ОБЩИЙ_ДЕЛИТЕЛЬ_В_ДИАЛОГЕ
НАЧ ИМЕНА:ПРИЗНАК
    ПРИЗНАК:="ДА"
    ПОКА ПРИЗНАК="ДА"
        НЦ
            НАИБОЛЬШИЙ_ОБЩИЙ_ДЕЛИТЕЛЬ
            ВЫВОД:"ЖЕЛАЕТЕ ЛИ ДРУГОЙ ПРИМЕР?"
            ВВОД:ПРИЗНАК
        КЦ
    КОН

```

Пока на вводимый программой вопрос человек введет ответ "ДА", программа предложит ему ввести и очередную пару чисел для вычисления их наибольшего общего делителя.

Разумеется, как и вспомогательный алгоритм вычисления функции НОД(A,B), так и процедура НАИБОЛЬШИЙ\_ОБЩИЙ\_ДЕЛИТЕЛЬ, должны быть заранее сообщены компьютеру (выражение из стр.70 учебника).

Следует подчеркнуть, что последние слова имеют чисто технический смысл, зависящий от конкретных особенностей используемой машины. В большинстве случаев "сообщить компьютеру программы" означает, что они просто набраны одна за другой на его клавиатуре.

Особое внимание обращаем на отличительную особенность команд ВВОД, ВВОД ДАННЫХ. Они вызывают приостановку работы программы до завершения человеком ввода информации. В получении этой информации суть команд ввода.

Домашнее задание: § 13, № 4 § 13.

#### ВОПРОСЫ И ОТВЕТЫ

1. Для чего применяются команды ввода и вывода?

Ответ.

Команды ввода и вывода являются основным средством организации диалога пользователя с компьютером. Команда ввода даёт возможность ввести с клавиатуры в компьютер, требуемые значения. Команда вывода обеспечивает вывод на экран дисплея значений.

2. Как выполняются команды ввода и вывода?

Ответ.

По команде ВВОД:Х на экране появляется знак "?" - приглашение к вводу. Выполнение программы приостанавливается. После набора на клавиатуре некоторой последовательности символов и нажатия клавиши перевода строки (именно эта клавиша говорит о завершении ввода), эта последовательность символов, которая воспринимается как текст, будет присвоена переменной Х. Команда

ВВОД ДАННЫХ:Х

обеспечивает ввод числовых значений.

Команда

ВЫВОД:Х

обеспечивает вывод на экран значения переменной Х.

При выводе каждое новое выполнение предписания ВЫВОД обеспечивает вывод на экран значения с новой строки.

Например, при выполнении предписания:

ВЫВОД:3; ВЫВОД:"ТЕКСТ"

на экране появится число 3 и слово ТЕКСТ:

3

ТЕКСТ

Для того, чтобы результаты выдавались на одной строке, необходимо выводимые значения указывать в одном предписании ВЫВОД, разделяя их запятыми.

Например,

ВЫВОД:3, "ТЕКСТ"

В результате выполнение этого предписания на экран будет выдано

3 ТЕКСТ

В предписании вывода можно также указывать выражения, значения которых будут выведены на экран дисплея.

### УПРАЖНЕНИЯ И РЕШЕНИЯ

1. Составьте программу, которая вводила бы два числа и выводила бы их произведение.

#### Решение

ПРОЦ ПРОИЗВЕДЕНИЕ

НАЧ ИМЕНА: X, Y

ВЫВОД: "ВВЕДИТЕ ПЕРВОЕ ЧИСЛО"

ВВОД ДАННЫХ: X

ВЫВОД: "ВВЕДИТЕ ВТОРОЕ ЧИСЛО"

ВВОД ДАННЫХ: Y

ВЫВОД: X \* Y

КОН

2. Составьте программу, которая вводила бы две логических величины и выводила бы их соединение.

#### Решение

ПРОЦ СОЕДИНЕНИЕ

НАЧ ИМЕНА: T1, T2

ВЫВОД: "ВВЕДИТЕ ПЕРВОЕ ЗНАЧЕНИЕ"

ВВОД: T1

ВЫВОД: "ВВЕДИТЕ ВТОРОЕ ЗНАЧЕНИЕ"

ВВОД: T2

ВЫВОД: T1 + T2

КОН

3. Программа должна ввести два числа - координаты левого нижнего угла прямоугольника, а затем ещё два числа - координаты правого верхнего угла этого прямоугольника. После этого программа запрашивает у ученика значение пе-

риметра этого прямоугольника и в зависимости от введённого значения либо хвалит ученика, выводя текст "МОЛОДЕЦ!", либо сообщает о первом результате, выводя текст "НЕВЕРНО".

#### Решение

ПРОЦ ПЕРИМЕТР

НАЧ ИМЕНА: P, X1, Y1, X2, Y2, P1

ВЫВОД: "ВВЕДИТЕ КООРДИНАТУ X1"

ВВОД ДАННЫХ: X1

ВЫВОД: "ВВЕДИТЕ КООРДИНАТУ Y1"

ВВОД ДАННЫХ: Y1

ВЫВОД: "ВВЕДИТЕ КООРДИНАТУ X2"

ВВОД ДАННЫХ: X2

ВЫВОД: "ВВЕДИТЕ КООРДИНАТУ Y2"

ВВОД ДАННЫХ: Y2

$P := 2 * (X2 - X1 + Y2 - Y1)$

ВЫВОД: "ЧЕМУ РАВЕН ПЕРИМЕТР ПРЯМОУГОЛЬНИКА?"

ВВОД ДАННЫХ: P1

ЕСЛИ P = P1

ТО ВЫВОД: "МОЛОДЕЦ!"

ИНАЧЕ ВЫВОД: "НЕВЕРНО"

ВСЕ

КОН

4. На соревнованиях по прыжкам в высоту установлена квалификационная норма - 180 см. По программе надо ввести фамилию спортсмена и взятую им высоту.

Программа проверяет результат и выдает фамилию прыгуна и одно из двух сообщений:

"НОРМА ВЫПОЛНИЛА"

или

"ВЫСОТА НЕ ВСЯКА"



Решение

ПРОЦ СОРЕВНОВАНИЯ

НАЧ ИМЕНА:ФАМИЛИЯ,ВЫСОТА

ВЫВОД:"ФАМИЛИЯ СПОРТСМЕНА?"

ВВОД:ФАМИЛИЯ

ВЫВОД:"РЕЗУЛЬТАТ?"

ВВОД ДАННЫХ:ВЫСОТА

ЕСЛИ ВЫСОТА &lt; 180

ТО ВЫВОД:"ВЫСОТА НЕ ВЗЯТА"

ИНАЧЕ ВЫВОД:"НОРМА ВЫПОЛНЕНА"

ВСЕ

КОН

5. Перепишите процедуру

РЕШЕНИЕ\_КВАДРАТНОГО\_УРАВНЕНИЯ так, чтобы коэффициенты уравнения можно было не задавать в виде аргументов, а вводить по ходу выполнения программ.

Решение.

ПРОЦ РЕШЕНИЕ\_КВАДРАТНОГО\_УРАВНЕНИЯ (X1=?, X2=?, Y=? )

НАЧ ИМЕНА:Д,А,В,С

ВЫВОД:"ВВЕДИТЕ КОЭФФИЦИЕНТ А"

ВВОД ДАННЫХ:А

ВЫВОД:"ВВЕДИТЕ КОЭФФИЦИЕНТ В"

ВВОД ДАННЫХ:В

ВЫВОД:"ВВЕДИТЕ КОЭФФИЦИЕНТ С"

ВВОД ДАННЫХ:С

 $D := B * 2 - 4 * A * C$ 

ЕСЛИ D &lt; 0

ТО Y:="НЕТ РЕШЕНИЯ"

ИНАЧЕ

Y:="ЕСТЬ РЕШЕНИЕ"

 $X1 := (-B + \sqrt{D}) / (2 * A)$  $X2 := (-B - \sqrt{D}) / (2 * A)$ 

ВСЕ

КОН

## СОДЕРЖАНИЕ

|   |    |
|---|----|
| АЛГОРИТМИЧЕСКИЙ ЯЗЫК .....                            | 3  |
| §6. КОМАНДА ВЫБОРА .....                              | 3  |
| ВОПРОСЫ И ОТВЕТЫ .....                                | 5  |
| УПРАЖНЕНИЯ И РЕШЕНИЯ .....                            | 7  |
| §7. КОМАНДА ПОВТОРЕНИЯ С ПАРАМЕТРОМ .....             | 14 |
| ВОПРОСЫ И ОТВЕТЫ .....                                | 17 |
| УПРАЖНЕНИЯ И РЕШЕНИЯ .....                            | 20 |
| Самостоятельная работа .....                          | 32 |
| §8. ВСПОМОГАТЕЛЬНЫЕ АЛГОРИТМЫ ВЫЧИСЛЕНИЯ ЗНАЧЕНИЙ     |    |
| ФУНКЦИЙ .....   | 34 |
| ВОПРОСЫ И ОТВЕТЫ .....                                | 37 |
| УПРАЖНЕНИЯ И РЕШЕНИЯ .....                            | 37 |
| ДОПОЛНИТЕЛЬНЫЕ УПРАЖНЕНИЯ И РЕШЕНИЯ .....             | 45 |
| §9. АЛГОРИТМЫ РАБОТЫ С ЛИТЕРНЫМИ ВЕЛИЧИНАМИ .....     | 47 |
| ВОПРОСЫ И ОТВЕТЫ .....                                | 49 |
| УПРАЖНЕНИЯ И РЕШЕНИЯ .....                            | 49 |
| ДОПОЛНИТЕЛЬНЫЕ УПРАЖНЕНИЯ И РЕШЕНИЯ .....             | 56 |
| ЯЗЫК ПРОГРАММИРОВАНИЯ РАПИРА .....                    | 59 |
| §10. ЗАПИСЬ АЛГОРИТМОВ В ВИДЕ ПРОЦЕДУР НА РАПИРЕ .... | 63 |
| ВОПРОСЫ И ОТВЕТЫ .....                                | 65 |
| УПРАЖНЕНИЯ И РЕШЕНИЯ .....                            | 66 |
| §11. ЗАПИСЬ АЛГОРИТМОВ ВЫЧИСЛЕНИЯ ЗНАЧЕНИЙ ФУНКЦИЙ    |    |
| НА РАПИРЕ .....                                       | 68 |
| ВОПРОСЫ И ОТВЕТЫ .....                                | 72 |
| УПРАЖНЕНИЯ И РЕШЕНИЯ .....                            | 73 |
| §12. КОРТЕЖИ НА РАПИРЕ .....                          | 75 |
| ВОПРОСЫ И ОТВЕТЫ .....                                | 78 |
| УПРАЖНЕНИЯ И РЕШЕНИЯ .....                            | 79 |
| §13. КОМАНДЫ ВВОДА И ВЫВОДА НА РАПИРЕ .....           | 83 |
| ВОПРОСЫ И ОТВЕТЫ .....                                | 85 |
| УПРАЖНЕНИЯ И РЕШЕНИЯ .....                            | 86 |