

Информатика 9 Б класс

<https://digital.prosv.ru/>, <https://media.prosv.ru/content/>, <https://media.prosv.ru/>

Группа компаний «Просвещение», открывает свободный доступ к электронным формам учебников

Дистанционные уроки на неделю с 6 по 10 апреля 2020, 1 час в неделю

Учитель физики информатики Гаджиагаев Тагир Гаджиагаевич

Учебник Семакин Залогова Информатика 9 класс

**Внимание! Ответы на вопросы и задания оформлять письменно в рабочих тетрадях.
Работы будут проверены**

1 занятие

§ 10 Линейные вычислительные алгоритмы, стр 69-74

Задание : Прочитать параграф, ответить на вопросы в конце параграфа

ВВОД A ... ВЫВОД A

ЕК ЦОР: часть 2, глава 6, § 33. ЦОР № 2, 7.

$A := A * A$ 22

$A := A * A$

Вывод: A

www

§ 10

Линейные вычислительные алгоритмы

Основные темы параграфа:

- присваивание; свойства присваивания;
- обмен значениями двух переменных;
- описание линейного вычислительного алгоритма.

Присваивание. Свойства присваивания

Поскольку присваивание является важнейшей операцией в алгоритмах, работающих с величинами, то поговорим о ней более подробно.



Переменная величина получает значение в результате присваивания.

Присваивание производится компьютером при выполнении одной из двух команд из представленной выше системы команд: команды присваивания или команды ввода.

Рассмотрим последовательность выполнения четырех команд присваивания, в которых участвуют две переменные: a и b . В приведенной ниже таблице против каждой команды указываются значения переменных, которые устанавливаются после ее выполнения. Такая таблица называется **трассировочной таблицей**, а процесс ее заполнения называется **трассировкой** алгоритма.

| Команда | a | b |
|------------|-----|-----|
| $a:=1$ | 1 | — |
| $b:=2 * a$ | 1 | 2 |
| $a:=b$ | 2 | 2 |
| $b:=a + b$ | 2 | 4 |

Проверка в таблице означает неопределенное значение переменной. Конечные значения, которые получают переменные a и b , соответственно равны 2 и 4.

Этот пример иллюстрирует три основных свойства присваивания. Вот эти свойства:

- 1) пока переменной не присвоено значение, она остается неопределенной;
- 2) значение, присвоенное переменной, сохраняется вплоть до выполнения следующего присваивания этой переменной нового значения;
- 3) новое значение, присвоенное переменной, заменяет ее предыдущее значение.

Обмен значениями двух переменных

Рассмотрим еще один очень полезный алгоритм, с которым при программировании часто приходится встречаться. Даны две переменные X и Y . Требуется произвести между ними обмен значениями. Например, если первоначально было: $X = 1; Y = 2$, то после обмена должно стать: $X = 2; Y = 1$.

Хорошим аналогом для решения такой задачи является следующая: даны два стакана, в первом — молоко, во втором — вода; требуется произвести обмен их содержимым. Всякому ясно, что в этом случае нужен дополнительный, третий, пустой стакан. Последовательность действий будет следующей:

- 1) перелить из 1-го стакана в 3-й;
- 2) перелить из 2-го стакана в 1-й;
- 3) перелить из 3-го стакана во 2-й.

Цель достигнута!

По аналогии для обмена значениями двух переменных нужна третья дополнительная переменная. Назовем ее Z . Тогда задача решается последовательным выполнением трех операторов присваивания (пусть начальные значения 1 и 2 для переменных X и Y задаются вводом):

| Команда | X | Y | Z |
|--------------|-----|-----|-----|
| ввод X, Y | 1 | 2 | — |
| $Z:=X$ | 1 | 2 | 1 |
| $X:=Y$ | 2 | 2 | 1 |
| $Y:=Z$ | 2 | 1 | 1 |
| вывод X, Y | 2 | 1 | 1 |

X - не должен потерять
3, 1, 2

Действительно, в итоге переменные X и Y поменялись значениями. На экран будут выведены значения X и $Y: 2, 1$. В трассировочной таблице выводимые значения выделены жирным шрифтом.

Аналогия со стаканами не совсем точна в том смысле, что при переливании из одного стакана в другой первый становится пустым. В результате же присваивания ($X:=Y$) переменная, стоящая справа (Y), сохраняет свое значение.

Описание линейного вычислительного алгоритма

Алгоритмы, результатами выполнения которых являются числовые величины, будем называть вычислительными алгоритмами.

Рассмотрим пример решения следующей математической задачи: даны две простые дроби; получить дробь, являющуюся результатом деления одной на другую.

В школьном учебнике математики правила деления обыкновенных дробей описаны так:

1. Числитель первой дроби умножить на знаменатель второй.
2. Знаменатель первой дроби умножить на числитель второй.
3. Записать дробь, числителем которой является результат выполнения пункта 1, а знаменателем — результат выполнения пункта 2.

В алгебраической форме это выглядит следующим образом:

$$\frac{a}{b} : \frac{c}{d} = \frac{a \cdot d}{b \cdot c} = \frac{m}{n}$$

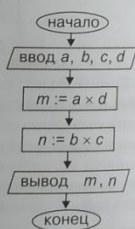
Теперь построим алгоритм деления дробей для компьютера. В этом алгоритме сохраним те же обозначения для переменных, которые использованы в записанной выше формуле. Исходными данными являются целочисленные переменные a, b, c, d . Результатом — также целые величины m и n .

Ниже алгоритм представлен в двух формах: в виде блок-схемы и на Алгоритмическом языке (АЯ).

Раньше прямоугольник в схемах алгоритмов управления мы называли блоком простой команды. Для вычислительных алгоритмов такой простой командой является команда присваивания. Прямоугольник будем называть блоком присваивания, или вычислительным блоком. В форме параллелограмма рисуется блок ввода/вывода. Полученный алгоритм имеет линейную структуру (рис. 2.1).

алг Деление дробей
цел a, b, c, d, m, n
нач
ввод a, b, c, d
 $m := a \times d$
 $n := b \times c$
вывод m, n
кон

опис переменной



$$\frac{1^a}{2^b} + \frac{1^c}{2^d} = \frac{2}{2} = 1$$

Рис. 2.1. Блок-схема алгоритма деления дробей

В алгоритме на АЯ строка, стоящая после заголовка алгоритма, называется **описанием переменных**. Служебное слово **цел** означает целый тип. Величины этого типа могут иметь только целочисленные значения.

Описание переменных имеет вид:

<тип переменных> <список переменных>

Список переменных включает все переменные величины данного типа, обрабатываемые в алгоритме.

В блок-схемах типы переменных не указываются, но подразумеваются. Запись алгоритма на АЯ ближе по форме к языкам программирования, чем блок-схемы.

Коротко о главном

Основные свойства присваивания:

- значение переменной не определено, если ей не присвоено никакого значения;
- новое значение, присваиваемое переменной, заменяет ее старое значение;
- присвоенное переменной значение сохраняется в ней вплоть до нового присваивания.

Обмен значениями двух переменных можно производить через третью дополнительную переменную.

Трассировочная таблица используется для «ручного» исполнения алгоритма с целью его проверки.

В алгоритмах на АЯ указываются типы всех переменных. Такое указание называется описанием переменных.

Числовые величины, принимающие только целочисленные значения, описываются с помощью служебного слова **цел** (целый).

Вопросы и задания

1. Из каких команд составляется линейный вычислительный алгоритм?
2. Что такое трассировка? Как она производится?
3. В каком случае значение переменной считается неопределенным?
4. Что происходит с предыдущим значением переменной после присваивания ей нового значения?
5. Как вы думаете, можно ли использовать в выражении оператора присваивания неопределенную переменную? К каким последствиям это может привести?
6. Напишите на АЯ алгоритм сложения двух простых дробей (без сокращения дроби).

Handwritten notes and symbols on the right margin of page 72, including a question mark and a checkmark.

АЯ опис
цел a, b, c, d, e, f
Чис

7. Напишите на АЯ алгоритм вычисления y по формуле

$$y = (1 - x^2 + 5x^4)^2,$$

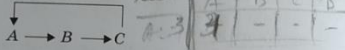
где x — заданное целое число. Учтите следующие ограничения: 1) в арифметических выражениях можно использовать только операции сложения, вычитания и умножения; 2) каждое выражение может содержать только одну арифметическую операцию. Выполните трассировку алгоритма при $x = 2$.

8. Пользуясь ограничениями предыдущей задачи, напишите наиболее короткие алгоритмы вычисления выражений:

$$y = x^8; y = x^{10}; y = x^{15}; y = x^{19}.$$

Постарайтесь использовать минимальное количество дополнительных переменных. Выполните трассировку алгоритмов.

9. Запишите алгоритм циклического обмена значениями трех переменных A, B, C . Схема циклического обмена:



Например, если до обмена было: $A = 1, B = 2, C = 3$, то после обмена должно стать: $A = 3, B = 1, C = 2$. Выполните трассировку.

www

ЕК ЦОР: часть 2, глава 6, § 34. ЦОР № 9, 10.

§ 11

Знакомство с языком Паскаль

Основные темы параграфа:

- возникновение и назначение Паскаля;
- структура программы на Паскале;
- операторы ввода, вывода, присваивания;
- правила записи арифметических выражений;
- пунктуация Паскаля.

Возникновение и назначение Паскаля

После того как построен алгоритм решения задачи, составляется программа на определенном языке программирования.

Среди современных языков программирования одним из самых популярных является язык Паскаль. Этот язык разработан в 1971 году и назван в честь Блеза Паскаля — французского ученого, изобретателя механической вычислительной машины. Автор языка Паскаль — швейцарский профессор Николаус Вирт.



Паскаль — это универсальный язык программирования, позволяющий решать самые разнообразные задачи обработки информации.

Команду алгоритма, записанную на языке программирования, принято называть **оператором**.

Программа на Паскале близка по своему виду к описанию алгоритма на АЯ. Сравните алгоритм решения уже знакомой вам задачи деления простых дробей с соответствующей программой на Паскале:

алг Деление дробей
цел a, b, c, d, m, n
нач
 ввод a, b, c, d
 $m := a \cdot d$
 $n := b \cdot c$
 вывод m, n
кон

```
Program Division;
var a, b, c, d, m, n: integer;
begin
  readln(a, b, c, d); {Ввод}
  m:= a*d; {Числитель}
  n:= b*c; {Знаменатель}
  write(m, n) {Вывод}
end.
```

Структура программы на Паскале

Даже не заглядывая в учебник по Паскалю, в этой программе можно все понять (особенно помогает знание английского языка).

Заголовок программы начинается со слова **Program** (программа), за которым следует произвольное имя, придуманное программистом:

```
Program <имя программы>;
```

Раздел описания переменных начинается со слова **Var** (variables — переменные), за которым идет список имен переменных через запятую. Тип указывается после двоеточия. В стандарте языка Паскаль существуют два типа числовых величин: **вещественный** и **целый**. Слово **integer** обозначает целый тип (является идентификатором целого типа). Вещественный тип обозначается словом **real**. Например, раздел описания переменных может быть таким:

```
var a, b: integer; c, d: real;
```