

## МЕТОДИКА БЫСТРОГО ОБУЧЕНИЯ ПРОГРАММИРОВАНИЮ НА ОСНОВЕ ИЗУЧЕНИЯ КЛАССОВ ЗАДАЧ (4-5)

*Юрий Александрович Аляев, доц., доц. кафедры  
математики и естественно-научных дисциплин,  
E-mail: alyr1@yandex.ru,*

*Российская академия народного хозяйства и государственной  
службы при Президенте РФ (Пермский филиал),  
<http://perm.ranepa.ru>*

*Предлагается методика быстрого обучения программированию на основе изучения классов задач, разработанная и применяющаяся на практике в процессе обучения программированию студентов вузов.*

*Ключевые слова: алгоритм, программа, язык программирования Паскаль, массив, константы, переменные, операции, функции, линейные Паскаль-программы.*

### Введение



**Ю.А. Аляев**

Разделы курса «Информатика» – алгоритмизация и программирование – остаются наиболее важными для формирования алгоритмического мышления. Поскольку в школах данные разделы преподаются в недостаточном объёме, в вузе возникает необходимость начинать обучение с нуля и достичь хорошего уровня программирования при ограниченном количестве часов преподавания.

Этого удаётся добиться за счёт применения рациональных методов обучения, прежде всего, последовательно проводя идеи обучения на основе выделения элементарных операций деятельности по построению алгоритмов и программ; выявления структуры алгоритма и форм ее записи на алгоритмическом языке; одинаковой формы алгоритма для решения задач с одинаковой структурой исходных данных [1–2]. Благодаря этим идеям, задачи по программированию удаётся разбить на ряд классов и типизировать методы решения задач каждого класса.

Предлагаемая методика быстрого обучения программированию на основе изучения классов задач, появилась и применяется на протяжении многих лет в процессе обучения программированию студентов пермских вузов благодаря В.П. Гладкову [2]. В статье рассматриваются методики решения по двум (4–5) из девятнадцати выделенных классов задач (1–3 классы задач рассмотрены в [3]).

### 4 Основы языка Паскаль

#### 4.1 Типы данных (константы, переменные, операции, функции)

Задача 1. Подсчитать количество имён, имеющих длину 4 символа, и которые на языке Паскаль можно записать, используя только следующие символы: 4, 8, 9, А, В.

*Решение.* Среди заданных символов только А и В могут стоять в начале имени, а на каждом из следующих трёх мест могут стоять любые из пяти заданных символов. В соответствии с основным принципом комбинаторики получаем  $2 \cdot 5 \cdot 5 \cdot 5 = 250$  имён.

Задача 2. Пусть имя построено в соответствии с правилами, описанными в предыдущей задаче. Написать функцию, проверяющую правильность построения имени, заданного параметром. Если имя построено в соответствии с правилами, то функция должна возвращать значение TRUE. В противном случае – значение FALSE.

*Решение.* Вначале проверяем символы со второго по четвёртый. Каждый из них должен быть одним из символов: 4, 8, 9, А, В. Формируя логическую переменную, ис-

пользуем операцию конъюнкции, чтобы учесть значения предыдущих символов. Затем проверяем длину имени (она должна равняться 4) и значение первого символа.

На любом естественном языке одну и ту же мысль можно высказать разными способами. Не является исключением и язык Паскаль.

Ниже приводятся три наиболее характерных, на наш взгляд, записи предложенного выше решения на языке Паскаль.

```
{ВАРИАНТ 1}
function prov_imja(s:string):boolean;
var i:integer;
    f:boolean;
begin f:=true;
    for i:=2 to 4 do f:=f and (s[i] in ['4','8','9','A','B']);
prov_imja:=(length(s)=4) and (s[1] in ['A','B']) and f;
end.
```

```
{ВАРИАНТ 2}
function prov_imja(s:string):boolean;
var i:integer;
    f:boolean;
begin f:=true;
    for i:=2 to 4 do f:=f and (pos(s[i],'489AB')>0);
prov_imja:=(length(s)=4) and (pos(s[1],'AB')>0) and f;
end.
```

```
{ВАРИАНТ 3}
function prov_imja(s:string):boolean;
var i:integer;
    f:boolean;
begin f:=true;
    for i:=2 to 4 do f:=f and ((s[i]='A') or (s[i]='B') or (s[i]='4') or (s[i]='8') or
(s[i]='9'));
prov_imja:=(length(s)=4) and ((s[1]='A') or (s[1]='B')) and f;
end.
```

Задача 3. Имена файлов строятся из символа 'Т', порядкового номера файла, с учетом ведущих нулей, символов 'a' или 'b' и расширения 'dat'. Написать программу, перечисляющую в алфавитном порядке имена всех файлов до 100 включительно.

*Решение.* В цикле 100 раз преобразуем счётчик цикла в строковое представление числа, формируем недостающие нули, а затем формируем требуемое имя файла.

```
var i:integer;
    s:string;
begin for i:=1 to 100 do
    begin str(i,s);
        if length(s)=1
        then s:='00'+s
        else if length(s)=2
        then s:='0'+s;
        write('T'+s+'a.dat T'+s+'b.dat');
    end;
end.
```

Задача 4. Описать переменные, содержащие следующую информацию о жильцах многоквартирного дома:

- 1) количество квартир на каждой площадке;

- 2) среднее количество жильцов в каждой квартире;
- 3) наличие в квартире собаки;
- 4) количество жильцов в каждой квартире.

*Решение*

- 1) var kv:integer; {количество квартир – целое число}
- 2) var sr:real; {разделив общее количество жильцов дома на количество квартир, можно получить дробное число}
- 3) var hund:boolean; {собака может быть (ДА, true) или отсутствовать (НЕТ, false), поэтому можно воспользоваться логическими данными}
- 4) var kk:array[1..nn] of integer; {количество жильцов в каждой квартире число четное (integer), квартир несколько (nn), все квартиры удобно представить в виде массива}

Задача 5. Выделить последнюю цифру целого десятичного числа k.

*Решение.* Цифрами десятичной системы счисления являются: 0,1,2,3,4,5,6,7,8,9. Им соответствуют остатки от деления целого числа на 10. Отсюда возможное решение –  $k \bmod 10$ .

#### **4.2 Арифметические выражения**

Задача 1. Записать выражение по правилам языка Паскаль:

$$\frac{\sqrt{\sin x + \left| \frac{\operatorname{tg} y}{a+b} \right|} - \alpha^3}{e^{14} + 24,5 \cdot 10^{-6}}.$$

*Решение.*  $(\operatorname{sqrt}(\sin(x)+\operatorname{abs}(\sin(y)/\cos(y)/(a+b)))-\alpha*\operatorname{sqr}(\alpha))/(\exp(14)+24.5e-6)$

Задача 2. Выражение записано на языке Паскаль:

$(a+b)/(a-b)*11e3+\operatorname{sqr}(\operatorname{sqr}(x))/(\operatorname{abs}(a-2*b)-1/1000)$ .

Записать его в общепринятой форме.

*Решение.*

$$\frac{11 \cdot 10^3 \cdot (a+b)}{a-b} + \frac{x^4}{|a-2 \cdot b| - 0,001}.$$

#### **4.3 Логический тип данных. Логические выражения**

Задача 1. Проверить, является ли число a четным.

*Решение.* Необходимо записать: условие истинно, если число a является четным, и ложно в противном случае. Согласно определению четное число – есть число, делящееся на 2 без остатка. Отсюда следует, что если при делении числа a на 2 остаток равен нулю, то число является четным. Условие на языке Паскаль можно записать одним из предложенных ниже вариантов.

- a mod 2=0;
- a-(a div 2)\*2=0;
- a-trunc(a/2)\*2=0.

Задача 2. Проверить, является ли число a кратным числу b.

*Решение.* Согласно определению: число a кратно числу b, если a делится на b нацело (без остатка). Отсюда:

- a mod b=0;
- a-(a div b)\*b=0
- или
- a-trunc(a/b)\*b.

Задача 3. Проверить, имеет ли действительные корни квадратное уравнение  $ax^2+bx+c=0$  ( $a \neq 0$ ).

*Решение.* Согласно определению, квадратное уравнение имеет действительные корни, если его дискриминант неотрицательный. Это определение можно записать на языке Паскаль так:  $\text{sqrt}(b*b-4*a*c) \geq 0$  или  $\text{sqrt}(\text{sqrt}(b)-4*a*c) \geq 0$ .

**Задача 4.** Проверить, лежит ли точка с координатами  $(x,y)$  вне круга радиуса  $r$  с центром в начале координат (точка  $0,0$ ).

*Решение.* Согласно определению точки, лежащие на окружности радиуса  $r$  с центром в точке  $(x_c, y_c)$ , удовлетворяют уравнению  $(x-x_c)^2+(y-y_c)^2=r^2$ . Поскольку нас интересуют точки, лежащие вне круга, следовательно, их расстояние до центра должно быть больше радиуса. С учетом того, что центр круга лежит в точке  $(0,0)$ , искомое условие запишется так:  $\text{sqrt}(x)+\text{sqrt}(y) > \text{sqrt}(r)$ .

**Задача 5.** Проверить, является ли натуральное число  $n$  полным квадратом.

*Решение.* Согласно определению, число является полным квадратом, если квадратный корень из него является числом целым. У целого числа отсутствует дробная часть, поэтому если извлечь квадратный корень из заданного числа, отбросить у него дробную часть и возвести результат в квадрат, то полученное значение может совпасть с исходным только тогда, когда исходное значение было целым. В результате получаем условие  $\text{sqrt}(\text{trunc}(\text{sqrt}(n)))=n$ .

**Задача 6.** Проверить, принадлежит ли действительное число  $x$  отрезку  $[-1;1]$ .

*Решение.* Согласно определению число принадлежит отрезку, если выполняется неравенство  $-1 \leq x \leq 1$  или  $|x| \leq 1$ . Соответствующее условие на Паскале запишется так:  $\text{abs}(x) \leq 1$  или  $(-1 \leq x) \text{ and } (x \leq 1)$ .

**Задача 7.** Проверить, принадлежит ли действительное число  $x$  отрезку  $[0;1]$ .

*Решение.* Согласно определению, число принадлежит отрезку, если выполняется неравенство  $0 \leq x \leq 1$ . Для того чтобы записать его так же, как в предыдущем примере, найдем середину интервала  $(1+0)/2=0,5$ . Затем из всех частей неравенства вычтем эту середину:  $-0,5 \leq x-0,5 \leq 0,5$  или  $|x-0,5| \leq 0,5$ . Условие на языке Паскаль запишется так:  $\text{abs}(x-0,5) \leq 0,5$  или  $(0 \leq x) \text{ and } (x \leq 1)$ .

**Задача 8.** Проверить, принадлежит ли действительное число  $x$  отрезку  $[a;b]$ .

*Решение.* Для истинности этого условия необходимо, чтобы выполнялось неравенство  $a \leq x \leq b$ . Найдем координату середины отрезка:  $a+(b-a)/2=(2 \cdot a+b-a)/2=(a+b)/2$ . Вычтем полученное значение из всех частей неравенства  $a \leq x \leq b$ , получим  $a-(a+b)/2 \leq x-(a+b)/2 \leq b-(a+b)/2$  или  $-(b-a)/2 \leq x-(a+b)/2 \leq (b-a)/2$ .

Отсюда  $|x-(a+b)/2| \leq (b-a)/2$ .

Запись на языке Паскаль:  $\text{abs}(x-(a+b)/2) \leq (b-a)/2$  или  $(a \leq x) \text{ and } (x \leq b)$ .

**Задача 9.** Записать на языке Паскаль выражение «Возраст человека заключен между 18 и 35 годами». Найти его отрицание и записать словами.

*Решение.* Пусть переменная  $x$  обозначает возраст человека. Тогда заданное высказывание может быть записано так:  $(18 \leq x) \text{ and } (x \leq 35)$ .

Для нахождения отрицания воспользуемся законами де Моргана.

$$\begin{aligned} \text{not}[(18 \leq x) \text{ and } (x \leq 35)] &= \text{not}(18 \leq x) \text{ or} \\ &\text{not}(x \leq 35) = (18 > x) \text{ or } (x > 35). \end{aligned}$$

Отрицание: «Неверно, что возраст человека  $x$  заключен между 18 и 35 годами». «Возраст человека меньше 18 или больше 35 лет»;  $(18 > x) \text{ or } (x > 35)$ .

**Задача 10.** Записать на языке Паскаль высказывание «Неверно, что треугольник ABC прямоугольный и равнобедренный».

*Решение.* Обозначим:  $P$  – треугольник ABC прямоугольный;  $R$  – треугольник ABC равнобедренный. Тогда высказывание запишется так:  $\text{not}(P \text{ and } R)$ .

Применяя закон де Моргана, получаем  $\text{not } P \text{ or } \text{not } R$ . Соответствующее высказывание: «Треугольник ABC не прямоугольный или неравнобедренный».

**Задача 11.** Записать на языке Паскаль высказывание «Неверно, что число 9 четное или простое».

*Решение.* Обозначим С – число 9 четное, Р – число 9 простое. Тогда исходное высказывание запишется на языке Паскаль так:  $\text{not}(C \text{ or } P)$ . Применяя закон де Моргана, получаем  $\text{not } C \text{ and } \text{not } P$ , что соответствует высказыванию: «Число 9 нечетное и непростое».

**Задача 12.** Записать на языке Паскаль высказывание «Неверно, что каждое из чисел  $m$  и  $n$  четно».

*Решение.* Если число  $m$  четное, то выполняется условие:  $m \bmod 2=0$ . Точно также, если число  $n$  четное, то выполняется условие:  $n \bmod 2=0$ . Учитывая эти факты, заданное высказывание можно записать так:  $\text{not}((m \bmod 2=0) \text{ and } (n \bmod 2=0))$ . Применяя закон де Моргана, получаем  $\text{not}(m \bmod 2=0) \text{ or } \text{not}(n \bmod 2=0)$  или  $(m \bmod 2 \neq 0) \text{ or } (n \bmod 2 \neq 0)$ . Последнее условие соответствует высказыванию: «Числа  $m$  или  $n$  нечетны».

**Задача 13.** Записать на языке Паскаль высказывание «Неверно, что, хотя бы одно из чисел  $r$  и  $s$  простое».

*Решение.* Обозначим  $r$  – простое число,  $s$  – простое число. Тогда заданное высказывание запишется на языке Паскаль так:  $\text{not}(r \text{ or } s)=\text{not } r \text{ and } \text{not } s$ . Последнее условие соответствует высказыванию: «Неверно, что  $r$  и  $s$  – простые числа».

**Задача 14.** Записать условие истинности, если  $x$  принадлежит отрезку  $[2;5]$  или  $[-1;1]$ . Затем найти его отрицание.

*Решение.* Это условие на языке Паскаль может быть записано так:  $((2 \leq x) \text{ and } (x \leq 5)) \text{ or } \text{abs}(x) \leq 1$ . Отрицание этого высказывания можно найти используя законы де Моргана:  $(x < -1) \text{ or } (1 < x) \text{ and } (x < 2) \text{ or } (x > 5)$ .

**Задача 15.** Записать на языке Паскаль высказывание «Каждый из трех друзей: Ваня, Коля или Петя, имеет карманные деньги».

*Решение.* Обозначим высказывание «Ваня имеет  $b$  рублей карманных денег» –  $b$ . «Коля имеет  $k$  рублей карманных денег» –  $k$ . «Петя имеет  $p$  рублей карманных денег» –  $p$ . Тогда заданное высказывание запишется так:  $(b > 0) \text{ and } (k > 0) \text{ and } (p > 0)$ .

**Задача 16.** Записать на языке Паскаль высказывание «Хотя бы один из трех друзей имеет карманные деньги».

*Решение.* В обозначениях предыдущего примера заданное условие будет иметь вид  $(b > 0) \text{ or } (k > 0) \text{ or } (p > 0)$ .

**Задача 17.** Записать на языке Паскаль высказывание «Ни один из друзей не имеет карманных денег».

*Решение.* В обозначениях 3.3.2.15 заданное условие запишется так:  $(b \leq 0) \text{ and } (k \leq 0) \text{ and } (p \leq 0)$ .

**Задача 18.** Записать на языке Паскаль высказывание «Хотя бы один из трех друзей не имеет карманных денег».

*Решение.* В обозначениях 3.3.2.15 заданное условие запишется так:  $(b \leq 0) \text{ or } (k \leq 0) \text{ or } (p \leq 0)$ .

**Задача 19.** Записать на языке Паскаль высказывание «Только один из друзей имеет карманные деньги».

*Решение.* В обозначениях 3.3.2.15 заданное условие запишется так:  $((b > 0) \text{ and } (k \leq 0) \text{ and } (p \leq 0)) \text{ or } ((b \leq 0) \text{ and } (k > 0) \text{ and } (p \leq 0)) \text{ or } ((b \leq 0) \text{ and } (k \leq 0) \text{ and } (p > 0))$ .

**Задача 20.** Записать на языке Паскаль высказывание «Целые  $n$  и  $k$  имеют одинаковую четность».

*Решение.* Числа имеют одинаковую четность, если их остатки от деления совпадают, поэтому  $n \bmod 2 = k \bmod 2$ .

Можно воспользоваться стандартной функцией odd(i): odd(n)=odd(k).

**Задача 21.** Записать на языке Паскаль высказывание «Числа  $x$ ,  $y$ ,  $z$  равны между собой».

*Решение.* Заданное высказывание можно записать так:  $(x=y) \text{ and } (x=z)$ .

**Задача 22.** Записать на языке Паскаль высказывание «Из чисел  $x$ ,  $y$ ,  $z$  только два равны между собой».

*Решение.* Вариант записи:  $(x=y) \text{ and } (x \neq z) \text{ or } (x=z) \text{ and } (x \neq y) \text{ or } (y=z) \text{ and } (x \neq y)$ .

**Задача 23.** Записать на языке Паскаль высказывание «Цифра три входит в десятичную запись трехзначного целого числа».

*Решение.* Вариант записи:  $(k \text{ div } 100 = 3) \text{ or } (k \text{ div } 10 \text{ mod } 10 = 3) \text{ or } (k \text{ mod } 10 = 3)$ .

#### 4.4 Строковый тип данных

**Задача 1.** Пусть описана переменная var a: string [3]. Нарисовать содержимое памяти компьютера после выполнения указанных операторов присваивания.

*Решение*

a:='';	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
a:='У';	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
a:='УА';	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
a:='УРА';	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
a:='УРАЛ';	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

**Задача 2.** Определить результат сравнения строк: 'школа' и 'школьник'.

*Решение.* Строки сравниваются посимвольно слева направо до первого результата или до исчерпания символов строки. Символы сравниваются как данные типа char.

'школа' < 'школьник'. Результат сравнения «истина» (true), так как 'ш'='ш', 'к'='к', 'о'='о', 'л'='л', 'а' < 'ь' (символ 'а' расположен в кодовой таблице раньше символа 'ь').

**Задача 3.** Привести примеры использования процедур и функций над строками.

*Пример 1.* СЦЕПЛЕНИЕ – CONCAT(строка1,строка2,...). Аналогична операции сцепления.

Исходные данные: a='код', b='ил'.

Оператор: s:=concat('кро',a,b).

Результат: S='крокодил'.

*Пример 2.* КОПИРОВАТЬ – COPY(строка,число1,число2). Из указанной строки выделяется подстрока, начиная с позиции, заданной числом 1, длиной, заданной числом 2.

Исходные данные: S='крокодил'.

Оператор: b:=copy(S,2,3).

Результат: b='рок'.

*Пример 3.* ПОЗИЦИЯ – POS(строка1,строка2). Отыскивает первое вхождение строки 1 в строке 2 и возвращает номер начальной позиции вхождения или нуль, если строка 1 не входит в строку 2.

Исходные данные: S='крокодил'.

Оператор: i:=pos('око',S).

Результат: i=3.

Оператор: i:=pos('я', 'крокодил').

Результат:  $i=0$ .

*Пример 4.* ДЛИНА – LENGTH(строка). Возвращает длину строки – аргумента.

Исходные данные:  $S='крокодил'$ .

Оператор:  $j:=length(S)$ .

Результат:  $j=8$ .

*Пример 5.* ВСТАВИТЬ – INSERT(строка1,строка2,число). Вставляет строку 1 в строку 2, начиная с позиции, заданной числом. Если в результате получается строка больше максимальной длины, то она усекается справа.

Исходные данные:  $S='крокодил'$ .

Оператор:  $d:=copy(S,3,3)$ .

Результат:  $d='око'$ .

Оператор:  $insert('H', d, 3)$ .

Результат:  $d='окно'$ .

*Пример 6.* УДАЛИТЬ – DELETE(строка,число1,число2). Удаляет из строки подстроку, начиная с позиции, заданной числом 1, длиной, заданной числом 2. Если число 1 больше размера строки, то подстрока не удаляется. Если число 2 больше имевшегося количества, то удаляются символы до конца строки.

Исходные данные:  $S='крокодил'$ .

Оператор:  $delete(S,4,3)$ .

Результат:  $S='кроил'$ .

Оператор:  $delete(S,1,1)$ .

Результат:  $S='роил'$ .

*Пример 7.* ПРЕОБРАЗОВАТЬ число в строку – STR(число[:M[:N]],строка). Преобразует число в строку. M задает общее количество символов, получаемых в строке, N – для вещественных чисел (типа real) задает количество цифр в дробной части.

$str(123,S)$ .

Результат:  $S='123'$ .

*Пример 8.* ПРЕОБРАЗОВАТЬ строку в число – VAL(строка,число,код). Преобразует строку символов во внутреннее представление числа. Код указывает номер неправильного символа или равен 0 в случае успешного преобразования.

$val('+12.3',v,k)$ .

Результат:  $v=12.3, k=0$  {преобразование прошло успешно}.

$val('23+5',v,k)$ .

Результат:  $v$  – неопределенно.  $k=3$  {ошибка при попытке преобразовать третий символ}.

Задача 4. Выделить часть строки после первого пробела.

*Решение.*  $copy(s, pos(' ', s) + 1, length(s) - pos(' ', s))$ .

Задача 5. Удалить последний символ строки S.

*Решение.*  $delete(S, length(S), 1)$ .

Задача 6. Записать оператор, позволяющий получить из слова «моряк» слово «морячок».

*Решение.* Пусть  $s='моряк'$ , тогда  $insert('чо', s, 5)$ .

Задача 7. Выделите из строки S подстроку между i-й и j-й позициями, включая эти позиции.

*Решение.*  $copy(s, i, j - i + 1)$ .

## **5 Линейные Паскаль-программы**

Задача 1. В алгоритмическом языке «МИНУС АЛЬФА» используются арифметические константы, которые имеют вид чисел с фиксированной точкой, и переменные,

имена которых строятся из строчных букв латинского алфавита, тип всех переменных – вещественный. Перед переменной может стоять знак минус, что приводит к изменению значения переменной на противоположное. Операторами языка являются операторы: ввод – `get <переменная>` – прочитать значение переменной с клавиатуры; `print <переменная>` – напечатать значение переменной; `<переменная>:=<переменная или константа>` `<операция>` `<переменная или константа>` – поместить результат операции в переменную, указанную справа. В качестве операций используются арифметические операции: сложение, вычитание, умножение, деление.

Написать программу на языке МИНУС АЛЬФА для вычисления значения выражения:

$$z = \frac{(a+3)^2 - 2/(a+3)}{(a+3)^3 + 5b}.$$

*Решение.* Для того чтобы вычислить значение заданного выражения, необходимы значения переменных  $a$  и  $b$ . Поэтому первыми действиями программы нужно организовать ввод нужных значений. Несложно заметить, что в исходной формуле имеется повторяющаяся часть:  $a+3$ . Её следует вычислить в первую очередь. Остальные части формулы можно вычислять в порядке старшинства операций. В заключение программы должен быть произведен вывод вычисленного значения. Соответствующая программа приводится ниже.

```
get a    {прочитали значение переменной a}
get b    {прочитали значение переменной b}
c:=a+3   {c равно a+3}
c1:=c*c  {c1 равно (a+3)2}
c2:=2/c  {c2 равно 2/(a+3)}
c:=c*c1  {c равно (a+3)3}
c1:=c1-c2 {в c1 хранится значение числителя}
c2:=5*b  {c2 равно 2b}
c2:=c+c2 {в c2 хранится значение знаменателя}
z:=c1/c2
print z
```

Задача 2. Написать на языке МИНУС АЛЬФА программу для вычисления значения выражения:  $10x^4 + 8x^3 + 6x^2 + 4x + 2$ .

*Решение.* Предложенная ниже программа вычисляет степени, перемножает их с заданными коэффициентами, а затем складывает. Общее количество действий равно 14.

```
get x
x2:=x*x
x3:=x2*x
x4:=x3*x
x4:=10*x4      {10x4}
x2:=x*x
x3:=x2*x
x3:=8*x3      {8x3}
x2:=x*x
x2:=6*x2      {6x2}
x:=4*x        {4x}
z:=x4+x3
z:=z+x2
z:=z+x
z:=z+2
print z
```



В предложенном решении повторяются части. Чтобы избавиться от повторов, используют представление заданного многочлена в виде схемы Горнера:  $((10x+8)x+6)x+4)x+2$ . Для вычисления выражения по этой форме потребуется только 8 действий.

```
get x
z:=10*x
z:=z+8
z:=z*x
z:=z+6
z:=z*x
z:=z+4
z:=z*x
z:=z+2
print z
```

**Задача 3.** Заданы два числа, которые хранятся в переменных a и b. Необходимо поменять местами значения этих переменных.

**Решение.** Заданы две переменные, которые хранят числовую информацию. Числовая информация в алгоритмическом языке может быть представлена как целая или вещественная (приближенная). Поскольку в условии задачи нет явного указания на тип числовой информации, будем использовать более общий тип – вещественный. Исходными данными могут быть, например, такие числа:

Исходные данные		Результат	
a	b	a	b
5	3	3	5
3.14	2.81	2.81	3.14
-40	+15	+15	-40

Решим задачу с помощью оператора присваивания  $a:=b$ . После его выполнения значения переменных a и b будут одинаковыми, но будет утеряно первоначальное значение переменной a. Для того чтобы его сохранить, поместим его в рабочую переменную r. Тогда последовательность операторов, решающая задачу, будет такой:

```
r:=a; { запомнили значение переменной a }
a:=b; { переменной a присвоили значение переменной b }
b:=r; { переменной b присвоили сохранённое в r старое значение a }
```

После выполнения этих операторов значения переменных поменялись местами. Решение получено, но так как существует несколько алгоритмов для решения задачи, продолжим поиски. Найдём алгоритм, который не требует введения вспомогательной переменной. Заметим, что исходные данные представляют собой числа, над которыми можно выполнять арифметические действия. Пусть  $a:=a+b$ . Известна сумма заданных значений и исходное значение переменной b. С помощью вычитания находим значение a и помещаем в b, т. е.  $b:=a-b$ . Теперь известна сумма заданных значений и исходное значение переменной a, которое хранится в переменной b. Для того чтобы решить задачу, осталось найти  $a:=a-b$ .

Проведём трассировку найденных операторов.

```
a:=a+b;
b:=a-b;
a:=a-b;
```

Пусть первоначально  $a=5$ ,  $b=3$ . После выполнения оператора  $a:=a+b$  получим  $a=8$ ,  $b=3$ . После исполнения оператора  $b:=a-b$  значения переменных изменятся:  $a=8$ ,  $b=5$ . После выполнения оператора  $a:=a-b$  окончательно получим  $a=3$ ,  $b=5$ .

Можно получить ещё одно решение, если воспользоваться двоичным представлением чисел. Над двоичными числами можно выполнять операцию исключающего ИЛИ (xor), которая задаётся следующей таблицей.

x	y	x xor y
0	0	0
0	1	1
1	0	1
1	1	0

С использованием двоичного представления чисел и операции xor поставленную задачу решает приведённая ниже последовательность операторов.

Пусть  $a=5$  в двоичном представлении 0101,  $b=11$  в двоичном представлении 1011.

a:= a xor b; xor

0101
1011
1110

Результат помещаем в a.

b:= a xor b; xor

1110
1011
0101

Результат помещаем в b.

a:= a xor b; xor

1110
0101
1011

Результат помещаем в a.

Окончательно получаем  $a=1011$ ,  $b=0101$ .

Из трёх рассмотренных алгоритмов второй и третий не используют дополнительной переменной, причём в третьем только одна операция. Однако этот алгоритм можно применить к двоичным данным (любим, хранящимся в виде двоичных последовательностей), а второй – только к числовым. Первый алгоритм имеет универсальный характер: с его помощью можно поменять местами не только числа, но и различные предметы: мел и тряпку, записи на магнитофонных кассетах и т. д. К тому же этот алгоритм представляется наиболее наглядным. Приведём запись этого алгоритма на языке Паскаль:

```

program обмен;
{ поменять местами значения двух переменных }
var a,b:real; { исходные данные и результат }
    r:real; { вспомогательная переменная }
begin write('Введите два числа ');
      readln(a,b);
      r:=a; { 1 }
      a:=b; { 2 }
      b:=r; { 3 }
      writeln('результат ',a,' ',b);
end.
    
```

**Задача 4.** Пусть построен робот «Кибик», умеет выполнять следующий набор команд и проверок, заданных в виде функций:

сум(a,b) – возвращает сумму аргументов а и b;  
раз(a,b) – возвращает разность аргументов а и b;  
кв(a) – возвращает квадрат аргумента а;  
пол(a) – возвращает половину аргумента а;  
?() – вводит число, заданное пользователем;  
!() – выводит результат;

знак(a) – проверяет аргумент и возвращает  $\begin{cases} +1, & \text{если } x > 0, \\ 0, & \text{если } x = 0, \\ -1, & \text{если } x < 0. \end{cases}$

Все используемые роботом числа – вещественные. Программа для робота строится в виде суперпозиции функций, т. е. вложения их друг в друга.

Пусть необходимо вычислить сумму двух заданных пользователем чисел.

*Решение.* Программа для Кибика запишется так: !(сум(?(),?())). Если полученную программу назвать сум2=!(сум(?(),?())) и записать в библиотеку программ робота, то ее можно будет использовать в других программах наравне с командами робота.

Задача 5. Составить для Кибика программу вычисления выражения: 352+498-703.

*Решение.* Здесь исходными данными являются три числа. Они могут быть любыми, поэтому в общем случае искомая программа должна находить значение выражения a+b-c.

Расставим скобки в программируемом выражении так, чтобы отразить порядок его выполнения: ((a+b)-c). Заменяем внутренние скобки командой робота, получим: (сум(a,b)-c),

а внешние скобки – соответствующей командой робота: раз(сум(a,b),c). Значения переменных должны быть введены, поэтому заменим их соответствующей командой. Получим: раз(сум(?(),?()),?()). Наконец, для вывода результата добавим соответствующую команду и законченная программа будет иметь вид: !(раз(сум(?(),?()),?())).

Другой вариант программы получим следующим образом. Расставляем скобки: (a+(b-c)). Операции в полученном выражении последовательно заменяем командами робота:

(a+раз(b,c))  
сум(a,раз(b,c))  
сум(?(),раз(?(),?()))  
!(сум(?(),раз(?(),?()))).

Задача 6. Составить для Кибика программу вычисления выражения  $\frac{a+b}{2} - c$ .

*Решение.* Здесь исходными данными являются числа а, b и с. Расставим в заданном выражении скобки, отражающие порядок его вычисления (((a+b)/2)-c). Заменяем каждую пару скобок, начиная с внутренних, соответствующими командами робота. Последовательно получим программу для робота. Заменяем сумму – ((сум(a,b)/2)-c), заменим деление на 2 – (пол(сум(a,b))-c), заменим разность – раз(пол(сум(a,b)),c).

Эту же задачу, можно было решить с помощью формулы: a/2+b/2-c. Тогда последовательно получим (((a/2)+(b/2))-c). Заменяем a/2–((пол(a)+(b/2))-c), заменим b/2–((пол(a)+пол(b))-c), заменим сумму – (сум(пол(a),пол(b))-c), заменим разность – раз(сум(пол(a),пол(b)),c).

Задача 7. Необходимо научить робота умножать, т. е. написать программу, которая под именем умн(a,b) будет включена в библиотеку программ робота, причем умн(a,b)=a·b.

*Решение.* Для построения алгоритма проанализируем систему команд робота. В ней нет команды умножения, но есть команды нахождения суммы, разности, возведения в квадрат. Найдем квадрат суммы или разности. Используя формулы сокращенного умножения, получим  $(a+b)^2 = a^2 + 2 \cdot a \cdot b + b^2$ . Отсюда

$$a \cdot b = ((a+b)^2 - a^2 - b^2) / 2. \quad (1)$$

Последнюю формулу можно переписать в виде:

$$a \cdot b = ((a+b)^2 - (a^2 + b^2)) / 2. \quad (2)$$

Используя другую формулу сокращенного умножения, получаем

$$a \cdot b = (a^2 + b^2 - (a-b)^2) / 2. \quad (3)$$

Построим программу для робота по (1). Расставим сначала скобки, отражающие порядок выполнения операций, а затем заменим операции в скобках соответствующими командами робота. Начнем замену с внутренних скобок (заменяемые части подчеркнуты). Последовательно получаем:

$a \cdot b = (((a+b)^2 - a^2) - b^2) / 2;$   
 $\text{умн}(a,b) = (((\text{сум}(a,b))^2 - a^2) - b^2) / 2;$   
 $\text{умн}(a,b) = (((\text{кв}(\text{сум}(a,b)) - a^2) - b^2) / 2);$   
 $\text{умн}(a,b) = (((\text{кв}(\text{сум}(a,b)) - \text{кв}(a)) - b^2) / 2);$   
 $\text{умн}(a,b) = ((\text{раз}(\text{кв}(\text{сум}(a,b)), \text{кв}(a)) - b^2) / 2);$   
 $\text{умн}(a,b) = ((\text{раз}(\text{кв}(\text{сум}(a,b)), \text{кв}(a)) - \text{кв}(b)) / 2);$   
 $\text{умн}(a,b) = (\text{раз}(\text{раз}(\text{кв}(\text{сум}(a,b)), \text{кв}(a)), \text{кв}(b)) / 2);$   
 $\text{умн}(a,b) = \text{пол}(\text{раз}(\text{раз}(\text{кв}(\text{сум}(a,b)), \text{кв}(a)), \text{кв}(b)) / 2).$

Программы по (2) и (3) строятся аналогично.

(2)  $\text{умн}(a,b) = \text{пол}(\text{раз}(\text{кв}(\text{сум}(a,b)), \text{сум}(\text{кв}(a), \text{кв}(b))))).$

(3)  $\text{умн}(a,b) = \text{пол}(\text{раз}(\text{сум}(\text{кв}(a), \text{кв}(b)), \text{кв}(\text{раз}(a,b))))).$

**Задача 8.** Написать программу для робота Кибика, вычисляющую  $y = |x|$ .

*Решение.* По определению  $y = \begin{cases} x, & \text{если } x > 0, \\ 0, & \text{если } x = 0, \\ -x, & \text{если } x < 0. \end{cases}$

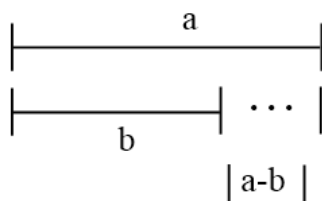
Сравним запись определения модуля с командой  $\text{знак}(a)$ , имеющейся в системе команд робота:

$$\text{знак}(a) = \begin{cases} +1, & \text{если } x > 0, \\ 0, & \text{если } x = 0, \\ -1, & \text{если } x < 0. \end{cases}$$

Сравнив записи, можно заметить, что первая может быть получена умножением команды  $\text{знак}(x)$  на  $x$ . Так можно получить новую команду:  $\text{мод}(x) = \text{умн}(\text{знак}(x), x)$ .

**Задача 9.** Написать программу для робота, указывающую максимальное из двух заданных чисел:  $\text{max}(a,b)$ .

*Решение.* Для нахождения максимального значения необходимо заданные числа сравнить между собой. Однако в системе команд робота нет операций сравнения, кроме операции  $\text{знак}(a)$ . Узнать, на сколько одно число отличается от другого, можно с помощью операции «разность». Для того чтобы такое сравнение было наглядным, представим числа  $a$  и  $b$  в виде отрезков, длины которых равны  $a$  и  $b$ , и нарисуем отрезки друг под другом (рисунок 1).



**Рисунок 1 – Отрезки**

Примем для определенности  $a > b$ . Из рисунка 1 видно, что длина большего отрезка больше длины меньшего отрезка на  $|a-b|$ . Если найти сумму длин отрезков  $a$ ,  $b$ ,  $|a-b|$ , то полученное число будет равно удвоенному максимальному. Отсюда  $a+b+|a-b|=2 \cdot \max$ . Тогда программа для работа будет иметь вид:

$$\max(a,b)=\text{пол}(\text{сум}(\text{сум}(\phi,и),\text{мод}(\text{раз}(\phi,и))))).$$

**Задача 10.** Написать программу для работа, определяющую минимальное из двух заданных чисел:  $\min(a,b)$ .

*Решение.* Аналогично предыдущему, можно получить формулу, а затем в соответствии с ней написать программу:

$$\min(a,b)=\text{пол}(\text{раз}(\text{сум}(a,b),\text{мод}(\text{раз}(a,b))))).$$

**Задача 11.** Написать программу для работа, упорядочивающую три заданных числа.

*Решение.* Здесь исходные данные: три произвольных числа. Обозначим их:  $a$ ,  $b$ ,  $c$ . Результат: эти же числа, расположенные в порядке: минимальное, среднее, максимальное.

Предположим, что у нас есть готовые программы для нахождения нужных значений. Назовем их:  $\min(a,b,c)$  – минимальное из трех,  $\text{ср}(a,b,c)$  – среднее из трех,  $\max(a,b,c)$  – максимальное из трех. Тогда поставленную задачу можно решить с помощью, например, такой программы:  $!(\min(a,b,c),\text{ср}(a,b,c),\max(a,b,c))$ .

Получив это решение, мы свели сложную исходную задачу к трем более простым. Решим их последовательно.

Для нахождения минимального из трех можно использовать программу  $\min(a,b,c)=\min(\min(a,b),c)$ . Здесь дважды применялась написанная ранее программа нахождения минимального из двух.

Аналогично можно написать программу для нахождения максимума из трех:  $\max(a,b,c)=\max(\max(a,b),c)$ . Но можно написать эту программу, используя ранее написанную:

$$\max(a,b,c)=\text{раз}(0,\min(\text{раз}(0,a),\text{раз}(0,b),\text{раз}(0,c))).$$

Для нахождения среднего по величине числа поступим так:  $\text{среднее}=a+b+c-\max(a,b,c)-\min(a,b,c)$ . В этом случае программа для работа будет иметь вид:

$$\text{ср}(a,b,c)=\text{раз}(\text{раз}(\text{сум}(\text{сум}(a,b),c),\max(a,b,c)),\min(a,b,c)).$$

Нетрудно также заметить, что средним будет число, являющееся максимальным из минимумов всевозможных пар, которые можно образовать из трех чисел. Всего возможны три пары:  $ab$ ,  $ac$ ,  $bc$ . Отсюда получаем программу:

$$\text{ср}(a,b,c)=\max(\min(a,b),\min(a,c),\min(b,c)).$$

Найдем среднее как минимум из максимумов пар, тогда программа будет иметь вид  $\text{ср}(a,b,c)=\min(\max(a,b),\max(a,c),\max(b,c))$ .

**Задача 12.** Написать программу для работа, вычисляющую

$$y=5(3x^2+4)^4-3(3x^2+4)^3+7(3x^2+4)^2-1/4 \cdot (3x^2+4).$$

*Решение.* Здесь исходные данные: переменная  $x$ . Результат: значение заданного выражения в переменной  $y$ .

Представим заданное выражение формулой  $y=((f1(x)-f2(x))+f3(x))-f4(x)$ . Здесь  $f1(x)$ ,  $f2(x)$ ,  $f3(x)$ ,  $f4(x)$  – вспомогательные алгоритмы для вычисления выражений  $5(3x^2+4)^4$ ,  $3(3x^2+4)^3$ ,  $7(3x^2+4)^2$ ,  $1/4 \cdot (3x^2+4)$ , соответственно.

Программа для работа будет иметь вид:

$$\text{выр}(x)=\text{раз}(\text{сум}(\text{раз}(f1(x), f2(x)), f3(x)), f4(x)).$$

Написав, например, заглушки:  $f1(x)=\text{сум}(1,1)$ ,  $f2(x)=\text{сум}(1,1)$ ,  $f3(x)=\text{сум}(1,1)$ ,  $f4(x)=\text{сум}(1,1)$ , можно проверить работоспособность программы.

Напишем программу для  $f1(x)=5(3x^2+4)^4$ . Заметим, что выражение в скобках повторяется во всех вспомогательных алгоритмах, поэтому разработаем для него отдельную программу.

$$ff(x)=3x^2+4=((3\cdot(x^2))+4)=\text{сум}(\text{умн}(3,\text{кв}(x)),4).$$

Четвертую степень можно найти как  $\text{кв}(\text{кв}(x))$ .

Тогда  $f1(x)=\text{умн}(5,\text{кв}(\text{кв}(ff(x))))$ . После этого заглушку  $f1(x)$  можно заменить написанной программой и проверить правильность работы всей программы.

Третью степень можно найти как  $\text{умн}(\text{кв}(x),x)$ .

Тогда  $f2(x)=\text{умн}(3,\text{умн}(\text{кв}(ff(x)),ff(x)))$ , а  $f3(x)=\text{умн}(7,\text{кв}(ff(x)))$ . Одну четвертую можно представить как  $\text{пол}(\text{пол}(x))$ , тогда  $f4(x)=\text{пол}(\text{пол}(ff(x)))$ .

Все вспомогательные алгоритмы разработаны. Заменяем заглушки и проверим работу всей программы. Можно поступить иначе, подставив написанные вспомогательные алгоритмы в основной. В результирующей записи необходимо проверить соответствие количества открывающих скобок количеству закрывающих.

Таким образом, представлены методики решения по двум из девятнадцати выделенных классов задач:

#### 4 Основы языка Паскаль.

4.1 Типы данных (константы, переменные, операции, функции).

4.2 Арифметические выражения.

4.3 Логический тип данных. Логические выражения.

4.4 Строковый тип данных.

#### 5 Линейные Паскаль-программы.

В следующей статье мы продолжим знакомство с методикой быстрого обучения программированию на основе изучения классов задач. Будут рассмотрены методики по следующим двум из девятнадцати выделенных классов задач:

#### 6 Ветвящиеся Паскаль-программы.

#### 7 Циклические Паскаль-программы.

### Литература

1. *Аляев Ю.А.* Алгоритмизация и языки программирования Pascal, C++, Visual Basic / Ю.А. Аляев, О.А. Козлов. М.: Финансы и статистика, 2002, 2004, 2007. 320 с.

2. *Аляев Ю.А.* Практикум по алгоритмизации и программированию на языке Паскаль / Ю.А. Аляев, В.П. Гладков, О.А. Козлов. М.: Финансы и статистика, 2004, 2007. 528 с.

3. *Аляев Ю.А.* Методика быстрого обучения программированию на основе изучения классов задач (1-3) // Образовательные ресурсы и технологии. 2015. № 1 (9). С. 3–14.

URL: [http://www.muiv.ru/ vestnik/pdf/pp/ot\\_2015\\_1\\_3-14.pdf](http://www.muiv.ru/ vestnik/pdf/pp/ot_2015_1_3-14.pdf)

### Methods of the quick education to programming on base of the study of the classes of the problems

*Yuri Alexandrovich Alyaev, assistant professor, assistant professor of the pulpit mathematicians and naturally-scientific discipline, The Russian Presidential Academy of National Economy and Public Administration (Permskiy branch).*

*Is offered methods of the quick education to programming on base of the study of the classes of the problems, designed and using in practice in process of the education to programming student high school.*

*The Keywords: algorithm, program, programming language Pascal, array, constants, variables, operations, functions, linear Pascal-programs.*