

**МЕТОДИКА БЫСТРОГО ОБУЧЕНИЯ ПРОГРАММИРОВАНИЮ
НА ОСНОВЕ ИЗУЧЕНИЯ КЛАССОВ ЗАДАЧ (1-3)**

*Юрий Александрович Аляев, доцент,
доцент кафедры математики и естественнонаучных 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, система счисления.

Введение

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



Ю.А. Аляев

Этого удастся добиться за счет 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 удастся разбить на ряд классов и типизировать методы решения задач каждого класса. Безусловно, для решения любой задачи можно предложить несколько алгоритмов, каждый из которых в свою очередь может быть записан разными программами. Эти решения могут существенно отличаться по времени и объему требуемой памяти. Далеко не всегда программа, получаемая на основе предлагаемого подхода, будет самой эффективной. Однако, используя изложенные методические приемы, программа может быть написана в приемлемое для начинающего программиста время. Затем можно (и нужно) экспериментировать с написанной программой, добиваясь ее совершенства, что, конечно, лучше, чем так и не получить никакого решения, затратив массу времени.

В качестве алгоритмического языка выбран язык Паскаль. Такой выбор объясняется, прежде всего тем, что в нем отражены фундаментальные и наиболее важные концепции алгоритмов в очевидной, естественной и легко воспринимаемой форме. Объем языка не велик, все конструкции хорошо продуманы, и позволяют:

- применять регулярные процедуры проектирования программ, например, структурное, модульное или инструментальное, объектно-ориентированное программирование;
- формировать простые и эффективные алгоритмы, так как он имеет хороший выбор структур данных;
- разрабатывать надежные программы, которые легко отлаживать и сопровождать. Это достигается многими средствами, например, требованием описывать все переменные. Несмотря на то, что в результате возникает избыточность программ, можно

обнаруживать ошибки в употреблении переменных в программе на этапе ее трансляции, а не во время исполнения.

Предлагаемая методика быстрого обучения программированию на основе изучения классов задач, появилась благодаря усилиям В.П. Гладкова [2], и применяется на протяжении многих лет в процессе обучения программированию студентов пермских вузов.

В статье рассматриваются методики решения задач по трем из девятнадцати выделенных классов.

1 Линейные и разветвляющиеся алгоритмы

Анализ алгоритмов заключается в исследовании процессов, происходящих в ходе решения задачи на ЭВМ, и является одним из способов поиска и устранения ошибок в алгоритме.

Основным инструментом анализа алгоритмов служит таблица значений, заголовок которой показан на рисунке 1.

Память	Условия	Вывод

Рисунок 1 – Заголовок таблицы значений

Заполнение таблицы производится последовательно, по мере имитации выполнения алгоритма в компьютере:

1) при появлении в алгоритме имени переменной оно записывается в столбец «Память», имитируя выделение в ЭВМ соответствующей ячейки памяти с таким же именем;

2) первое значение, которое получит переменная, записывается в столбец «Память» в выделенную для нее строку. Например, если переменная А получает значение 5, то в строке А следует записать: А=5;

3) каждое новое значение переменной записывается в строке с ее именем через запятую. Например, если переменная А получает новое значение 7, то в строке А запись будет иметь вид: А=5,7. Следует помнить, что в памяти компьютера новое значение переменной будет записано на место старого;

4) столбцы «Условие» и «Вывод» заполняются по мере появления соответствующих предписаний в алгоритме.

Методику анализа алгоритмов рассмотрим на конкретных задачах.

Задача 1. Определить, что будет выведено на печать в результате выполнения следующего алгоритма, если при вводе переменные получили значения: А=2.5; В=0.5.

- 1 Начало;
- 2 Ввод(А,В);
- 3 $X:=A-B$;
- 4 $Y:=A+B$;
- 5 $Z:=Y^X$;
- 6 $Z:=\frac{\sqrt{10 \cdot A - Z}}{Z + 1}$;
- 7 Вывод(А,В,Х,У,З);
- 8 Конец.

Алгоритм решения задачи представлен в виде псевдокодов и содержит 8 строк. В вычислительном процессе используются пять переменных: A, B, X, Y, Z. Команда Начало означает, что процессор ЭВМ приступил к решению данной задачи. Оператор Ввод(A, B) позволяет записать в ячейку памяти A число 2.5, а в ячейку B – число 0.5. В третьей строке алгоритма записана операция присваивания $X := A - B$. В ЭВМ она реализуется следующим образом: содержимое ячеек A и B направляется в процессор, где выполняется операция вычитания. Результат выполнения этой операции (2) записывается в ячейку X. Оператор присваивания $Y := A + B$ реализуется аналогично: содержимое ячеек A и B снова направляется в процессор, результат операции сложения (3) записывается в ячейку Y. После выполнения пятой операции в ячейку Z будет записано число 9. При выполнении шестой операции в процессор направляются числа из ячеек A и Z (2.5 и 9). Затем в соответствии с правилами приоритета выполняются операции, стоящие в правой части оператора присваивания: умножение, вычитание, вычисление квадратного корня, сложение, деление. Полученный результат (0.4) записывается в ячейку Z вместо старого значения (9).

Оператор Вывод(X, Y, Z) позволяет вывести значения переменных X, Y и Z на экран дисплея или принтер.

Команда Конец означает, что процессор ЭВМ завершил решение задачи.

Таблица значений, заполненная в ходе анализа алгоритма задачи 1, будет иметь вид, показанный в таблице 1.

Таблица 1

Таблица значений

Память	Условия	Вывод
A=2.5 B=0.5 X=2 Y=3 Z=9,0.4		2.5 0.5 2 3 0.4

Задача 2. Определить значение переменной X, которое будет получено в результате выполнения фрагмента алгоритма:

.....
Если $A \geq B$ То
X:=2;
Иначе
X:=3;
Конец-Если;

.....
при условии, что переменные A и B имеют следующие значения: A=2, B=2.

Для анализа фрагмента алгоритма используем таблицу значений (таблица 2).

При реализации оператора ветвления проверяется условие $A \geq B$. Условие выполняется, значит в ячейку X записывается число 2.

Построение алгоритма является одним из этапов решения задачи с использованием ЭВМ. Этому этапу предшествует анализ задачи на содержательном уровне и составление математической модели решаемой задачи.

Задача 3. Разработать алгоритм для вычисления и печати значения функции Y , заданной формулой:

$$Y = e^{X^2+1} + \cos(X^2 + 1) + \ln(X^2 + 1) .$$

Таблица 2

Таблица значений

Память	Условия	Вывод
A=2 B=2 X=2	$2 \geq 2$ да	

В алгоритме предусмотреть ввод аргумента X и вывод на печать введенной информации и результатов решения.

При анализе данной задачи на содержательном уровне определяется ее сущность, возможность решения на ЭВМ. При необходимости на задачу могут быть наложены определенные допущения и ограничения.

Составление математической модели целесообразно начать с определения переменных и констант, которые должны быть введены в машину в качестве исходных данных. Затем пользователь должен определить, какие переменные и в каком виде необходимо получить в качестве результатов решения. Далее необходимо подобрать математические зависимости, которые позволяют преобразовать исходные данные в конечные результаты.

В рассматриваемой задаче математическую модель можно записать следующим образом:

Ввод: X .

Расчет: $Y = e^{X^2+1} + \cos(X^2 + 1) + \ln(X^2 + 1) .$

Вывод: X, Y .

Следует обратить внимание на то, что математическое выражение для расчета переменной Y содержит три повторяющихся фрагмента X^2+1 . В результате выделения этого фрагмента в промежуточную переменную не только сокращается запись соответствующего оператора присваивания, но и уменьшается расчет примерно на 100 коротких машинных операций.

Алгоритм должен содержать оператор Список данных. Этот оператор обеспечивает резервирование необходимого количества ячеек памяти машины. В нем указываются имена переменных и их атрибуты. После оператора Ввод целесообразно поставить оператор вывода всех вводимых переменных и констант. Это позволяет избежать ошибок ввода, связанных с ошибками набора и преобразования информации в машинные коды.

Алгоритм решения задачи в виде псевдокодов будет иметь следующий вид:

- 1 Начало;
- 2 Список данных: X, Y, Z – вещ.;
- 3 Ввод(X);
- 4 Вывод(X);
- 5 $Z := X^2 + 1$;
- 6 $Z := e^Z + \cos(Z) + \ln(Z)$;
- 7 Вывод(Y);
- 8 Конец.

Задача 4. Разработать алгоритм, обеспечивающий присваивание переменной A значения 1 при выполнении следующего условия:

переменная X принадлежит промежутку $]0;1[$.

Данная задача математически определена не полностью. Из условия задачи не ясно, какое значение получит A , если переменная X не попадает в указанный промежуток.

В этом случае пользователь должен сам принять решение. Например, пусть переменная A получит значение 0, если переменная X не удовлетворяет указанному в задаче условию.

Математическую модель решаемой задачи можно записать так:

Ввод: X .

Расчет: $A = \begin{cases} 1, & \text{если } 0 < X < 1; \\ 0, & \text{если } 1 \leq X \leq 0. \end{cases}$

Вывод: X, A .

Алгоритм решения задачи, записанный в виде псевдокодов, имеет следующий вид:

- 1 Начало;
- 2 Список данных: X – вещ.; A – цел.;
- 3 Ввод(X);
- 4 Вывод(X);
- 5 $A := 0$;
- 6 Если ($X > 0$ и $X < 1$) То
- 7 $A := 1$;
- 8 Конец-Если 6;
- 9 Вывод(A);
- 10 Конец.

2 Алгоритмы, содержащие циклические структуры.

Алгоритмы обработки массивов

Независимо от способа задания алгоритма (псевдокоды, развернутая схема, компактная схема), анализ алгоритма целесообразно проводить по развернутой схеме, т. к. только в ней содержится полный перечень выполняемых операций и указана последовательность их выполнения. Основным инструментом анализа остается таблица значений.

Задача 1. Определить значение переменной M , которое будет получено в результате выполнения фрагмента алгоритма:

```
.....  
M:=A;  
Цикл по T от A до B шаг C;  
    M:=M+1;  
Конец-Цикла;  
M:=M+B;  
.....
```

при условии, что переменные имеют следующие значения: $A=1$; $B=5$; $C=1$.

Данный фрагмент алгоритма содержит цикл с заданным числом повторений. Параметр T изменяется от 1 до 5 с шагом 1. Сначала в таблицу состояния записывается содержимое ячеек памяти перед выполнением цикла: $A=1$; $B=5$; $C=1$; $M=1$ (таблица 3).

Таблица значений

Память	Условия	Вывод
A=1	1>5 нет	
B=5	2>5 нет	
C=1	3>5 нет	
M=1,2,3,4,5,6,11	4>5 нет	
T=1,2,3,4,5,6	5>5 нет	

В соответствии с развернутой схемой цикла с заданным числом повторений, параметру цикла присваивается начальное значение: T=1. Затем проверяется условие: «Значение параметра больше конечного значения?»; в нашем случае: 1>5? Данное условие не выполняется, поэтому далее следуют операторы тела цикла. В рассматриваемом примере в теле цикла стоит один оператор присваивания: M:=M+1. В ячейку M записывается число 2. Затем к параметру добавляется шаг, и T принимает значение, равное 2. После проверки условия 2>5 снова выполняется оператор, стоящий в теле цикла, происходит увеличение параметра на величину шага, и цикл повторяется в очередной раз. Последний раз тело цикла будет выполняться, когда параметр равен конечному значению, т. е. T=5.

По окончании цикла: M=6, T=6. Следующим за циклом стоит оператор присваивания M:=M+B, поэтому после выполнения всего фрагмента алгоритма M=11 (таблица 3).

Разработка алгоритма начинается с анализа задачи на содержательном уровне и с составления математической модели.

Задача 2. Разработать алгоритм, обеспечивающий вычисление суммы элементов массива:

$$C = \sum_{I=1}^N X_I .$$

Данная задача может быть решена с помощью ЭВМ следующим образом. В памяти машины выделяется ячейка C, в которую первоначально записывается 0. Затем, используя цикл с заданным числом повторений, к содержимому ячейки C прибавляют значение очередного элемента массива X. Математическая модель решаемой задачи имеет вид:

Ввод: N – число элементов массива;
X(1:N) – значения элементов массива X.

Расчет:

$$C = \sum_{I=1}^N X_I .$$

Вывод: X,C.

Схема алгоритма решения задачи приведена на рисунке 2.

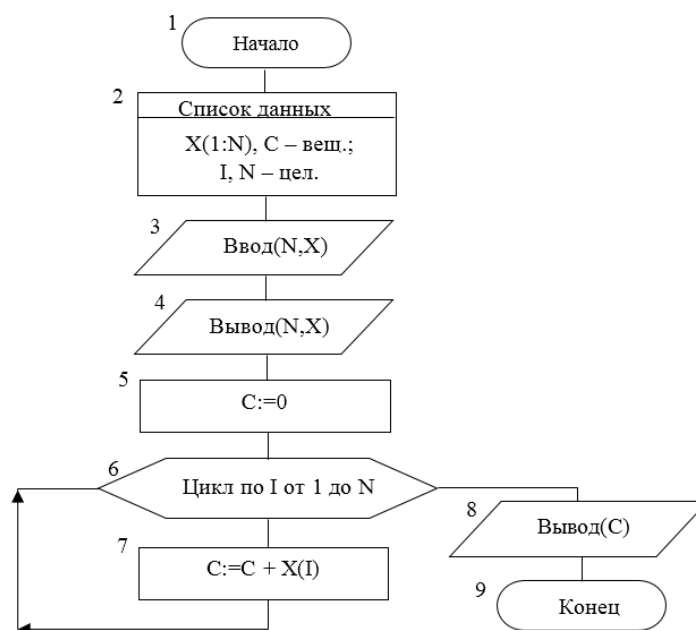


Рисунок 2 – Схема алгоритма

Практические задачи часто требуют обработки многомерных массивов.

Задача 3. Разработать алгоритм, позволяющий найти и вывести на печать произведение элементов двухмерного массива X, содержащего N строк и M столбцов.

Для решения этой задачи необходимо использовать два цикла с параметром, причем циклы должны располагаться один внутри другого («цикл в цикле»).

Математическая модель решаемой задачи имеет вид:

Ввод: N,M,X(1:N,1:M).

Расчет:

$$C = \prod_{I=1}^N \prod_{J=1}^M X(I,J).$$

Вывод: X,C.

Алгоритм решения задачи в виде псевдокодов содержит следующие операторы:

- 1 Начало;
- 2 Список данных: X(1:N,1:M), C – вещ.;
N,M,I,J – цел.;
- 3 Ввод(N,M,X);
- 4 Вывод(N,M,X);
- 5 C:=1;
- 6 Цикл по I от 1 до N;
- 7 Цикл по J от 1 до M;
- 8 C:=C*X(I,J);
- 9 Конец-Цикла 7;
- 10 Конец-Цикла 6;
- 11 Вывод(C);
- 12 Конец.

Задача поиска экстремума является одной из распространенных практических задач.

Задача 4. Разработать алгоритм, позволяющий найти максимальный элемент в массиве X(1:N).

Решение данной задачи можно начать с предположения, что максимальным элементом является первый элемент массива, и записать его значение в специально выделенную ячейку памяти XMAX. Затем сравнить значение очередного элемента с содержимым ячейки XMAX. Если очередной элемент больше XMAX, то максимальным становится этот элемент, и его значение заносится в ячейку XMAX.

Алгоритм решения задачи в виде псевдокодов содержит следующие операторы:

- 1 Начало;
- 2 Список данных: X(1:N), XMAX – вещ.;
N, I – цел.;
- 3 Ввод(N, X);
- 4 Вывод(N, X);
- 5 XMAX:=X(1);
- 6 Цикл по I от 2 до N;
- 7 Если X(I)>XMAX То
- 8 XMAX:=X(I);
- 9 Конец-Если 7;
- 10 Конец-Цикла 6;
- 11 Вывод(XMAX);
- 12 Конец.

3 Системы счисления

Задача 1. Перевести целое число из десятичной системы счисления в систему счисления с основанием p.

Решение. Для перевода целого числа нужно его делить на p до получения целого частного. Полученный остаток, в том числе и 0, будет младшим разрядом числа в новой системе счисления. Полученное частное нужно опять разделить на p до получения целого частного и продолжать этот процесс до тех пор, пока частное не станет нулем. Полученные остатки будут разрядами числа в новой системе счисления. Число в системе счисления с основанием p получается выписыванием в обратном порядке полученных остатков. Все действия выполняются по правилам десятичной системы счисления.

Пример. Перевести $39_{10} \rightarrow x_2$.

$$\begin{array}{r} 39 \overline{) 2} \\ \underline{38} \quad 19 \overline{) 2} \\ 1 \quad 18 \quad 9 \overline{) 2} \\ \quad 1 \quad 8 \quad 4 \overline{) 2} \\ \qquad 1 \quad 4 \quad 2 \overline{) 2} \\ \qquad \qquad 0 \quad 2 \quad 1 \overline{) 2} \\ \qquad \qquad \qquad 0 \quad 0 \\ \qquad \qquad \qquad \qquad 1 \end{array}$$

Ответ: $39_{10} \rightarrow 100111_2$.

Задача 2. Написать программу для перевода целого числа из десятичной системы счисления в систему счисления с основанием q ($q \leq 36$).

Решение.

```
const alf='0123456789abcdefghijklmnopqrstuvwxy';  
    {Возможные цифры в новой системе счисления}  
var s:string; {строка с результатом перевода}  
    n,        {исходное число}  
    q:longint; {основание новой системы счисления}begin
```



```

write('Введите число в десятичной системе счисления ');readln(n);
write('Введите основание новой системы счисления ');readln(q);
s:='';
while n>0 do
  begin s:=copy(alf,n mod q+1,1)+s;
        n:=n div q
  end;
write(s);
end.

```

Задача 3. Перевести число, заданное в виде правильной дроби, из десятичной системы счисления в систему счисления с основанием q .

Решение. Дробная часть переводится путем умножения ее на основание новой системы счисления. Полученная при этом целая часть является очередным разрядом числа в новой системе счисления. Процесс выполняется столько раз, сколько требуется получить знаков в дробной части числа новой системы счисления.

Пример. Перевести $0,97_{10} \rightarrow x_2 \rightarrow x_8$.

0,	97	0,	97
	×2		×8
1,	94	7,	76
	×2		×8
1,	88	6,	08
	×2		×8
1,	76	0,	64
	×2		×8
1,		5,	12

Ответ: $0,97_{10} \rightarrow 0,1111_2 \rightarrow 0,7605_8$.

В общем случае, когда задана s -разрядная дробь в системе счисления с основанием q , можно считать, что все ее цифры точные и погрешность этой дроби не превышает половины последнего разряда $\frac{1}{2}q^{-s}$. Если результат перевода изобразить t -разрядной дробью в системе счисления с основанием p , то погрешность этого приближения не превосходит $\frac{1}{2}p^{-t}$. Исходя из требования равноточности обоих представлений, получа-

ем приближенное равенство $p^{-t} \approx q^{-s}$ или $t \approx \frac{s \cdot \log q}{\log p}$ (логарифмы можно брать при любом основании). Для того чтобы обеспечить минимальный запас точности у результата перевода, в качестве t следует принять наименьшее целое число, большее или равное $s \cdot \log_p q$.

Применив формулу к рассмотренному примеру, имеем при переводе в двоичную систему счисления семь цифр в дробной части $2 \cdot \ln 10 / \ln 2 \approx 7$, т.е. нужно было получить $0,97_{10} = 0,1111100_2$ (в примере получено мало цифр). При переводе в восьмеричную систему счисления имеем три цифры в дробной части $2 \cdot \ln 10 / \ln 8 \approx 3$, т.е. нужно было получить $0,97_{10} = 0,760_8$ (в примере получены лишние цифры).

Задача 4. Написать программу для перевода числа, заданного в виде правильной дроби, из десятичной системы счисления в систему счисления с основанием q .

Решение.

```
const alf='0123456789abcdefghijklmnopqrstuvwxyz';
```

```

{Возможные цифры в новой системе счисления}
var s:string; {исходная строка, содержащая цифры дробной части}
n:real; {числовое представление переводимого числа}
i, {счетчик цикла}
q, {основание новой системы счисления}
k, {количество цифр в дробной части результата}
k1:integer; {код успешности преобразования строка - число}begin
write('Введите основание СС q ');readln(q); write('Введите правильную дробь
в 10 СС 0. ');
readln(s);
k:=round(length(s)*ln(10)/ln(q)+0.5);
s:='0.'+s;
val(s,n,k1);
s:='0.';n:=n*q;
for i:=1 to k do
begin s:=s+copy(alf,trunc(n)+1,1);
n:=frac(n)*q;
end;
write(s);
end.

```

Задача 5. Перевести целое число из системы счисления с основанием p в десятичную систему счисления.

Решение. Для перевода из системы счисления с основанием p в десятичную систему счисления нужно представить число в виде многочлена и вычислить значение этого многочлена по правилам арифметики десятичной системы счисления.

Пример. Перевести число 100111 из двоичной в десятичную систему счисления.

$$100111_2 \rightarrow 1 \cdot 2^5 + 0 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = 32 + 4 + 2 + 1 = 39_{10}.$$

Удобно вычислять значение многочлена, используя схему Горнера:

$$a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x^1 + a_0 = (\dots (a_n x + a_{n-1}) x + \dots + a_1) x + a_0.$$

Для рассмотренного примера:

$$100111_2 \rightarrow (((((1 \cdot 2 + 0) \cdot 2 + 0) \cdot 2 + 1) \cdot 2 + 1) \cdot 2 + 1).$$

Задача 6. Написать программу перевода целого числа из системы счисления с основанием q в десятичную систему счисления.

Решение.

```

const alf='0123456789abcdefghijklmnopqrstuvwxyz'; {Возможные цифры в но-
вой системе счисления}
var s:string; {исходное число в системе счисления с основанием q}
n, {результат}
i, {счетчик цикла}
q:longint; {основание старой системы}begin
write('Введите основание СС q ');readln(q); write('Введите число в СС q
');readln(s); n:=pos(s[1],alf)-1; for i:=2 to length(s) do n:=n*q+pos(s[i],alf)-1;
write(n);
end.

```

Задача 7. Написать программу для перевода правильной дроби из системы счисления с основанием q в десятичную систему счисления.

Решение. Для такого перевода также целесообразно воспользоваться схемой Горнера, учитывая, что требуемое значение является многочленом от q^{-1} .

```

const alf='0123456789abcdefghijklmnopqrstuvwxyz'; {Возможные цифры в но-
вой системе счисления}

```

```

var s:string; {исходная дробь}
    n:real; {результат перевода} i, {счетчик цикла} q, {основание
старой системы счисления} k:integer; {требуемое количество цифр}begin
write('Введите основание СС q ');readln(q);
write('Введите правильную дробь в СС q 0, ');readln(s);
k:=round(length(s)*ln(q)/ln(10)+0.5);
n:=0;
for i:=length(s) downto 1 do
    n:=(n+pos(s[i],alf)-1)/q;
write(n:0:k);
end.
    
```

Задача 8. Перевести смешанное число из системы счисления с основанием q в десятичную систему счисления.

Решение. Пусть нужно число 11.0010010_2 из двоичной системы счисления перевести в десятичную. Подсчитаем, сколько цифр должно быть в дробной части числа в десятичной системе счисления: $7 \cdot \ln 2 / \ln 10 \approx 3$.

Переводим число 110010010_2 в десятичную систему: $(((((1 \cdot 2 + 1) \cdot 2 + 0) \cdot 2 + 0) \cdot 2 + 1) \cdot 2 + 0) \cdot 2 + 0) \cdot 2 + 1) \cdot 2 + 0 + 402$.

Делим на $2^7 = 128$ с тремя знаками после точки: $402/128 = 3.141$.

Задача 9. Перевести число из двоичной системы счисления в четверичную, восьмеричную, шестнадцатеричную системы счисления и обратно.

Решение. Для перевода числа из двоичной системы счисления в четверичную (восьмеричную или шестнадцатеричную) необходимо переводимое число разбить по две (три или четыре) цифры влево и вправо от запятой и каждую группу цифр заменить соответствующим четверичным (восьмеричным или шестнадцатеричным) эквивалентом. Для этого необходимо знать таблицу соответствия:

«10»	«16»	«8»	«4»	«2»	«10»	«16»	«8»	«4»	«2»
0	0	0	0	0	9	9	11	21	1001
1	1	1	1	1	10	A	12	22	1010
2	2	2	2	10	11	B	13	23	1011
3	3	3	3	11	12	C	14	30	1100
4	4	4	10	100	13	D	15	31	1101
5	5	5	11	101	14	E	16	32	1110
6	6	6	12	110	15	F	17	33	1111
7	7	7	13	111	16	10	20	100	10000
8	8	10	20	1000	17	11	21	101	10001

Пример.

$110001011100100_2 \rightarrow \underline{11} \mid \underline{10} \mid \underline{00} \mid \underline{10} \mid \underline{11} \mid \underline{10} \mid \underline{01} \mid \underline{00}_2 \rightarrow 32023210_4$

$110001011100100_2 \rightarrow \underline{1} \mid \underline{110} \mid \underline{001} \mid \underline{011} \mid \underline{100} \mid \underline{100}_2 \rightarrow 161344_8$

$1110001011100100_2 \rightarrow \underline{1110} \mid \underline{0010} \mid \underline{1110} \mid \underline{0100}_2 \rightarrow E2E4_{16}$

Таким образом, рассмотрены методики решения задач по трем из девятнадцати выделенных классов:

1. Линейные и разветвляющиеся алгоритмы.

2 Алгоритмы, содержащие циклические структуры. Алгоритмы обработки массивов.

3 Системы счисления.

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

4 Основы языка Паскаль.

5 Линейные 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.

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, system of the numeration.

УДК 004.9

КОНЦЕПТУАЛЬНЫЕ ОСОБЕННОСТИ ОБУЧЕНИЯ СТУДЕНТОВ ВУЗА ИНФОРМАЦИОННЫМ ТЕХНОЛОГИЯМ В УСЛОВИЯХ КРЕДИТНОЙ ТЕХНОЛОГИИ

Мелис Казыкеевич Асаналиев, д-р пед. наук, проф., академик МАНПО РФ,

E-mail: melis.kazykeevich@mail.ru,

Казахский Национальный Педагогический Университет имени Абая,

http://www.kaznpu.kz

Куляш Нарзилдаевна Байсалбаева, канд. пед. наук, доц.,

E-mail: melis.kazykeevich@mail.ru,

Алматинская академия экономики и статистики,

http://www.aesa.kz,

Руслан Мелисович Асаналиев, канд. социол. наук,

старший преподаватель,

E-mail: melis.kazykeevich@mail.ru,

Академия дизайна и технологии «Сымбат», УО,

http://professionali.ru,

Жанар Жумагалиевна Жумагалиева, канд. техн. наук, старший преподаватель,

E-mail: melis.kazykeevich@mail.ru,

Казахский Национальный Педагогический Университет имени Абая,

http://www.kaznpu.kz