



**МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ
ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
ГРАЖДАНСКОЙ АВИАЦИИ**

В.Н. Агеев

ИНФОРМАТИКА И ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

**Учебно-методическое пособие
по проведению практических занятий**

*для студентов I курса
направления 25.03.01
очной формы обучения*

**Москва
2017**

ФЕДЕРАЛЬНОЕ АГЕНТСТВО ВОЗДУШНОГО ТРАНСПОРТА

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ
БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ

**«МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ ГРАЖДАНСКОЙ АВИАЦИИ» (МГТУ ГА)**

**Кафедра прикладной математики
В.Н. Агеев**

**ИНФОРМАТИКА
И ИНФОРМАЦИОННЫЕ
ТЕХНОЛОГИИ**

**Учебно-методическое пособие
по проведению практических занятий**

*для студентов I курса
направления 25.03.01
очной формы обучения*

Москва-2017

ББК 6ф6.5
А23

Рецензент д-р техн. наук, проф. А.А. Егорова

Агеев В.Н.

А23 Информатика и информационные технологии: учебно-методическое пособие по проведению практических занятий. – М.: МГТУ ГА, 2017. – 32 с.

Данное учебно-методическое пособие издается в соответствии с рабочей программой учебной дисциплины «Информатика и информационные технологии» по учебному плану для студентов I курса направления 25.03.01 очной формы обучения.

Рассмотрено и одобрено на заседаниях кафедры 15.11.2016 г. и методического совета 20.12.2016 г.

Подписано в печать 03.02.2017 г.

Печать офсетная
1,8 усл.печ.л.

Формат 60x84/16
Заказ № 1725/139

1,13 уч.-изд. л.
Тираж 80 экз.

Московский государственный технический университет ГА
125993 Москва, Кронштадтский бульвар, д.20
ООО «ИПП «ИНСОФТ»
107140, г. Москва, 3-й Красносельский д.21, стр.1

Содержание

	Введение.....	4
Часть I.	Арифметические и логические основы ЭВМ	5
Тема 1.	Системы счисления.....	5
Тема 2.	Арифметические операции с двоичными числами.....	6
Тема 3.	Формы представления чисел. Обратный и дополнительный коды.....	8
Тема 4.	Законы алгебры логики.....	9
Тема 5.	Логические функции. Способы задания логических функций.....	11
Тема 6.	Карты Карно.....	13
Тема 7.	Комбинационные схемы.....	14
Тема 8.	Понятие алгоритма. Блок-схемы алгоритмов.....	15
Часть II.	Программирование на языке Visual Basic.....	19
Тема 1.	Язык Visual Basic. Типы переменных. Преобразование типов.....	19
Тема 2.	Программирование разветвляющихся вычислительных процессов.....	20
Тема 3.	Операторы ввода и вывода.....	22
Тема 4.	Операторы цикла.....	23
Тема 5.	Действия с символьными строками.....	24
Тема 6.	Графические возможности языка Visual Basic.....	26
Тема 7.	Работа с файлами последовательного доступа.....	28
Тема 8.	Файлы прямого доступа.....	29
	Литература.....	32

Введение

Данные методические указания предназначены для студентов 1-го курса дневной формы обучения по направлению 25.03.01 и содержат описания практических работ в 1-м и 2-м семестрах. Методические указания могут быть использованы в качестве учебно-методического материала по аналогичным дисциплинам других направлений подготовки.

Первая часть пособия содержит материал, изучаемый в 1-м семестре и включает следующие темы:

- системы счисления;
- арифметические основы ЭВМ;
- логические основы ЭВМ;
- алгоритмы и блок-схемы.

Вторая часть содержит материал, изучаемый во 2-м семестре и включает следующие темы:

- основы языка Visual Basic;
- программирование линейных вычислительных процессов;
- программирования разветвляющихся и циклических процессов;
- графические возможности языка Visual Basic;
- работа с файлами прямого и последовательного доступа.

В каждом разделе приведены краткие сведения по методам решения задач и приведены соответствующие примеры.

Во второй части рассматриваются вопросы разработки программ на языке Visual Basic. Особое внимание уделено вопросам организации ввода и вывода данных, программированию разветвляющихся вычислительных процессов, работе с числовыми массивами, построению графических изображений. Ко всем изучаемым темам прилагаются задания для самостоятельной работы. Выполнение практических работ предполагается после изучения лекционного материала по соответствующим темам.

Целью выполнения практических работ является:

- закрепление пройденного материала;
- выполнения комплексного индивидуального задания студентам;
- приобретения практических навыков программирования различных типов задач.

Часть I. Арифметические и логические основы ЭВМ

Тема 1. Системы счисления.

Позиционная система счисления – способ записи чисел, в котором учитывается порядок следования числовых знаков (цифр). Количество цифр, используемых при записи чисел, определяет основание системы. В десятичной системе используются 10 цифр: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9. В двоичной системе – две: 0 и 1. В восьмеричной – восемь: 0, 1, 2, 3, 4, 5, 6, 7. В шестнадцатеричной – шестнадцать: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F.

В позиционной системе каждой цифре приписывается вес, равный основанию системы в степени, величина которой определяется номером соответствующего разряда. Для целых чисел разряды нумеруются справа налево, начиная с нуля.

Примеры:

$$457_{10} = 4 \cdot 10^2 + 5 \cdot 10^1 + 7 \cdot 10^0; \quad 273_8 = 2 \cdot 8^2 + 7 \cdot 8^1 + 3 \cdot 8^0; \quad 101_2 = 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0$$

Для дробной части номера разрядов отрицательные: -1, -2, -3 и т.д.:

$$0,293_{10} = 2 \cdot 10^{-1} + 9 \cdot 10^{-2} + 3 \cdot 10^{-3}; \quad 0,45_8 = 4 \cdot 8^{-1} + 5 \cdot 8^{-2}; \quad 0,101_2 = 1 \cdot 2^{-1} + 0 \cdot 2^{-2} + 1 \cdot 2^{-3}$$

Такая запись чисел называется развернутой.

Развернутая запись используется для перевода чисел из любой системы счисления в десятичную систему.

Для перехода из десятичной системы в другую с основанием q используются правила «деление уголком» (для целой части) и «умножение столбиком» (для дробной).

Примеры:

$$\begin{array}{r} 75 \overline{) 2} \\ \underline{1 \ 37} \ 2 \\ \underline{1 \ 18} \ 2 \\ \underline{0 \ 9} \ 2 \\ \underline{1 \ 4} \ 2 \\ \underline{0 \ 2} \ 2 \\ \ 0 \ 1 \end{array}$$

$$\begin{array}{r} 75 \overline{) 8} \\ \underline{3 \ 9} \ 8 \\ \underline{1 \ 1} \end{array}$$

$$\begin{array}{r} 75 \overline{) 16} \\ \underline{11} \ 4 \end{array}$$

$$75_{10} = 1001011_2 = 113_8 = 4B_{16}.$$

$$\begin{array}{r} 0,35 \times 2 \\ \hline 0,70 \times 2 \\ \hline 1,40 \times 2 \\ \hline 0,80 \times 2 \\ \hline 1,60 \times 2 \\ \hline 1,20 \end{array}$$

$$\begin{array}{r} 0,35 \times 8 \\ \hline 2,80 \times 8 \\ \hline 6,40 \times 8 \\ \hline 3,20 \times 8 \\ \hline 1,60 \times 8 \\ \hline 4,80 \end{array}$$

$$\begin{array}{r} 0,35 \times 16 \\ \hline 5,60 \times 16 \\ \hline 9,60 \times 16 \\ \hline 5,60 \end{array}$$

$$0,35 = 0,01011_2 = 0,26314_8 = 0,599_{16}.$$

Гораздо проще выполняется переходы между системами, основаниями которых являются степени двойки:

$$\begin{array}{ccc} \underbrace{1001011}_2 = 113_8 & \underbrace{1001011}_2 = 4B_{16} & \underbrace{0,010110}_2 = 0,26_8 \\ \begin{array}{ccc} 1 & 1 & 3 \end{array} & \begin{array}{cc} 4 & 11 \end{array} & \begin{array}{cc} 2 & 6 \end{array} \end{array}$$

Из двоичной в восьмеричную: число разбивается на триады (тройки) справа налево и каждая триада заменяется соответствующей десятичной цифрой. Для дробной части так же по три разряда, но справа налево. Из двоичной в 16-ричную правило то же, но разряды разбиваются на тетрады (группы по четыре разряда в каждой).

Задание 1.

1. Перевести число А из десятичной системы в 16-ричную, 8-ричную и двоичную.

Вариант	А	Вариант	А	Вариант	А
1	134,23	9	324,22	17	49,556
2	84,445	10	89,98	18	194,09
3	201,15	11	224,09	19	200,11
4	119,22	12	79,192	20	77,901
5	99,88	13	206,46	21	84,009
6	221,11	14	199,86	22	184,15
7	87,116	15	207,98	23	211,08
8	125,41	16	22,909	24	66,101

2. Перевести двоичное число А в 8-ричную, 16-ричную и десятичную систему.

Вариант	А	Вариант	А	Вариант	А
1	10100111010	9	11000110001	17	10011111011
2	11101000101	10	10110110110	18	10001111101
3	11000110111	11	10111001101	19	10101011110
4	10010100110	12	11111100011	20	11111000111
5	11011010011	13	11000011011	21	11110000110
6	10101010101	14	11100011111	22	10000111101
7	11001100110	15	11001111011	23	10001111001
8	10110110101	16	10001111111	24	11111111011

Тема 2. Арифметические операции с двоичными числами.

Во всех системах счисления сложение двух чисел выполняется поразрядно, начиная с младшего. Если результат сложения в текущем разряде больше или равен основанию системы, то из него нужно вычесть значение основания и единицу перенести в следующий разряд:

$$\begin{array}{r}
 1 \\
 + 357 \\
 \hline
 129 \\
 \hline
 486
 \end{array}
 \quad
 \begin{array}{r}
 11 \\
 + 01 \\
 \hline
 11 \\
 \hline
 1100
 \end{array}$$

← единицы переноса →

При выполнении умножения двоичных чисел используется то свойство, результатом умножения любого числа на 2^n будет то же число, сдвинутое на n разрядов влево:

$$1011 \times 1000 = 1011000; \quad 1011 \times 100 = 101100; \quad 1011 \times 10 = 10110.$$

Поэтому умножение двух двоичных целых чисел можно выполнять «столбиком», как и в десятичной системе:

$$\begin{array}{r}
 1011001 \\
 1011 \\
 \hline
 1011001 \\
 + 1011001 \\
 \hline
 1011001 \\
 \hline
 1111010011
 \end{array}$$

Деление в любой позиционной системе счисления производится по тем же правилам, как и деление в десятичной системе. В двоичной системе деление выполняется особенно просто, ведь очередная цифра частного может быть только нулем или единицей.

Пример.

Разделить число 30 на число 6.

Десятичная

$$\begin{array}{r}
 - 30 \overline{) 6} \\
 \underline{30} \\
 0
 \end{array}$$

Двоичная

$$\begin{array}{r}
 - 11110 \overline{) 110} \\
 \underline{110} \\
 110 \\
 \underline{110} \\
 0
 \end{array}$$

Восьмеричная

$$\begin{array}{r}
 - 36 \overline{) 6} \\
 \underline{36} \\
 0
 \end{array}$$

Задание 2.

1. Записать заданные числа А и В в двоичной и восьмеричной системах и выполнить сложение. Результат проверить, сложив заданные десятичные числа.

Вариант	А	В	Вариант	А	В	Вариант	А	В
1	216	129	9	159	932	17	168	601
2	431	510	10	358	458	18	990	122
3	670	111	11	723	610	19	136	882
4	146	981	12	831	109	20	358	109
5	361	230	13	199	334	21	490	333
6	542	101	14	201	333	22	862	199
7	589	459	15	459	259	23	113	982

Вариант	А	В	Вариант	А	В	Вариант	А	В
8	195	117	16	677	123	24	832	444

2. Выполнить умножение двоичных чисел А и В. Результат проверить в десятичной системе.

Вариант	А	В	Вариант	А	В	Вариант	А	В
1	1011	1010	9	1111	1101	17	1110	1101
2	1101	1100	10	1010	1101	18	1110	1001
3	1001	1110	11	1101	1011	19	1001	1011
4	1110	1010	12	1001	1001	20	1111	1000
5	1111	1001	13	1011	1011	21	1000	1100
6	1001	1011	14	1010	1111	22	1101	1101
7	1010	1010	15	1011	1100	23	1111	1111
8	1111	1110	16	1001	1111	24	1000	1110

Тема 3. Прямой, обратный и дополнительный коды.

Для записи двоичных чисел со знаком используются специальные коды, в которых старший разряд является знаковым. Туда записывается 0, если число положительное и 1, если отрицательное. В остальных разрядах записывается модуль числа. Это называется прямым кодом.

Кроме прямого, существуют также обратный и дополнительный коды числа со знаком. Для положительных чисел все три кода совпадают. Для отрицательных чисел обратный код получается из прямого инвертированием всех двоичных цифр модуля. Дополнительный код получается прибавлением 1 в младший разряд обратного кода.

Пример.

	Десятичное число	Двоичное число со знаком	Прямой код	Обратный код	Дополнительный код
А	253	11111101	011111101	011111101	011111101
В	-197	-11000101	111000101	100111010	100111011

Обратный и дополнительный коды используются для того, чтобы операцию вычитания заменить операцией поразрядного сложения: $253 - 197 = 253 + (-197) = A + B$.

Для этого поразрядно складываются обратные или дополнительные коды двоичных чисел. Старший (знаковый) разряд также учитывается при сложении. Если при сложении обратных кодов возникает единица переноса в старшем разряде, она прибавляется к полученному результату. При сложении дополнительных кодов такая единица переноса не прибавляется.

Результаты сложения обратных и дополнительных кодов показаны в следующей таблице. В строке А+В в сумму обратных кодов добавлена единица переноса из знакового разряда.

	Десятичное число	Прямой код	Обратный код	Дополнительный код
A	253	011111101	011111101	011111101
B	-197	111000101	100111010	100111011
A+ B			000111000	000111000

Проверка: $A+B = 253 - 197 = 56_{10} = 000111000_2$.

Задание 3.

1. Выполнить сложение чисел A и B, представив их двоичными обратными кодами. Результат сложения перевести в десятичное число и проверить результат, выполнив операцию сложения с десятичными числами.

Вариант	A	B	Вариант	A	B	Вариант	A	B
1	221	-118	9	313	-520	17	-222	599
2	119	-303	10	455	-227	18	-331	487
3	-199	218	11	521	-333	19	554	-239
4	201	-222	12	-259	399	20	298	-444
5	-234	454	13	-299	-101	21	301	-419
6	-334	491	14	333	-555	22	-292	-111
7	229	-355	15	291	-444	23	-332	-109
8	360	-579	16	-238	-322	24	313	-298

2. Выполнить сложение чисел A и B, представив их двоичными дополнительными кодами. Результат сложения перевести в десятичное число и проверить результат, найдя сумму заданных десятичных чисел.

Вариант	A	B	Вариант	A	B	Вариант	A	B
1	-121	-221	9	-213	-120	17	202	-390
2	-199	-333	10	-155	-203	18	-231	-187
3	-210	319	11	321	-433	19	-154	-339
4	-201	423	12	210	-309	20	191	-434
5	-334	155	13	211	-401	21	-301	-401
6	-300	499	14	-133	-505	22	-390	411
7	-220	-315	15	-201	-111	23	-312	319
8	-160	-279	16	-301	-332	24	-333	-208

Тема 4. Законы алгебры логики.

Логической переменной называют переменную, принимающую только одно из двух возможных значений: единица и ноль («истина» или «ложь»). Ло-

гической функцией называют функцию, принимающую на любом наборе определяющих ее переменных только одно из двух возможных значений – ноль или единица.

Логические операции				
инверсия	отрицание	НЕ (NOT)	$\neg A$	\bar{x}
конъюнкция	умножение	И (AND)	$x \wedge y$	$x \cdot y$
дизъюнкция	сложение	ИЛИ (OR)	$x \vee y$	$x + y$

Результаты применения логических операций задаются следующими таблицами:

НЕ	
x	\bar{x}
0	1
1	0

И		
x \ y	0	1
0	0	0
1	0	1

ИЛИ		
x \ y	0	1
0	0	1
1	1	1

Логические операции подчиняются законам алгебры логики (алгебры Дж. Буля).

Закон	Операция ИЛИ	Операция И
Переместительный	$x \vee y = y \vee x$	$x \cdot y = y \cdot x$
Сочетательный	$x \vee (y \vee z) = (x \vee y) \vee z$	$x \cdot (y \cdot z) = (x \cdot y) \cdot z$
Распределительный	$x \cdot (y \vee z) = x \cdot y \vee x \cdot z$	$x \vee (y \cdot z) = (x \vee y) \cdot (x \vee z)$
Правила де Моргана	$\overline{x \vee y} = \bar{x} \cdot \bar{y}$	$\overline{x \cdot y} = \bar{x} \vee \bar{y}$
Поглощения	$x \vee (x \cdot y) = x$	$x \cdot (x \vee y) = x$
Склеивания	$(x \cdot y) \vee (\bar{x} \cdot y) = y$	$(x \vee y) \cdot (\bar{x} \vee y) = y$
Операция переменной с ее инверсией	$x \vee \bar{x} = 1$	$x \cdot \bar{x} = 0$
Операция с константами	$x \vee 0 = x; x \vee 1 = 1$	$x \cdot 1 = x; x \cdot 0 = 0$
Двойного отрицания	$\overline{\bar{x}} = x$	

Примеры.

1. Подсчитать значение логического выражения $x \cdot (y + \bar{x} \cdot z) + z$ при $x = 1$, $y = 1$, $z = 0$.

Решение: При вычислениях учитывается порядок действий: вначале выполняются действия в скобках, самая старшая операция – отрицание, затем логическое умножение и логическое сложение.

Ответ: $1 \cdot (1 + 0 \cdot 0) + 0 = 1$.

2. Упростить выражение: $x + (y \cdot \bar{x} + x \cdot y + z)$

Решение:

$$x + (y \cdot \bar{x} + x \cdot y + z) = x \cdot y \cdot \bar{x} + x \cdot x \cdot y + x \cdot z = \bar{x} \cdot x \cdot y + x \cdot x \cdot y + x \cdot z = x \cdot (y+z)$$

Здесь использованы свойства конъюнкции (операции И): $x \cdot \bar{x} = 0$ и $x \cdot x = x$.

Задание 4.

Подсчитать значение логического выражения.

№	Выражение	Аргументы	№	Выражение	Аргументы
1	$x \cdot \bar{y} + \bar{y} \cdot (x \cdot \bar{z} + y \cdot z \cdot (x+y))$	$x=1, y=0, z=1$	12	$(x+\bar{y}) \cdot (x+y \cdot \bar{z} \cdot (\bar{x}+y))$	$x=0, y=1, z=1$
2	$(x+y) \cdot (x+y \cdot \bar{z} \cdot (x+\bar{y} \cdot x))$	$x=0, y=1, z=0$	13	$\bar{x}+y \cdot x \cdot (\bar{z} + \bar{y} \cdot z \cdot (x+y))$	$x=1, y=0, z=1$
3	$\bar{x}+(y+(y \cdot (x+\bar{y} \cdot z \cdot (x+y))))$	$x=1, y=1, z=0$	14	$(x+y) \cdot (\bar{x} + \bar{y} \cdot z \cdot (x+y \cdot x))$	$x=1, y=1, z=0$
4	$x \cdot \bar{y}+(y \cdot x \cdot (z+y \cdot \bar{z} \cdot (x+y)))$	$x=1, y=0, z=1$	15	$x \cdot \bar{y} + \bar{y} \cdot x \cdot \bar{z} + y \cdot z \cdot (x+y)$	$x=0, y=1, z=1$
5	$x+y \cdot \bar{x} \cdot (\bar{z}+y \cdot z \cdot (x+y))$	$x=0, y=1, z=0$	16	$\bar{x} + y \cdot (\bar{x} \cdot z + \bar{y} \cdot z \cdot (x+y))$	$x=0, y=1, z=0$
6	$\bar{x} \cdot (\bar{y}+y \cdot (x \cdot \bar{z}+y \cdot z \cdot (x+y)))$	$x=0, y=1, z=1$	17	$\bar{x} \cdot \bar{y} + (y \cdot x \cdot (z + y \cdot (x+y)))$	$x=1, y=0, z=1$
7	$x \cdot y+\bar{y} \cdot (x \cdot z+y \cdot z \cdot (x+\bar{y}))$	$x=1, y=1, z=0$	18	$(x+y) \cdot (\bar{x}+y \cdot z \cdot (x+\bar{y}))$	$x=0, y=1, z=1$
8	$(x+\bar{y}) \cdot (x+y \cdot \bar{z} \cdot (x+\bar{y} \cdot x))$	$x=1, y=0, z=1$	19	$\bar{x}+y \cdot x \cdot (\bar{z} + y \cdot z \cdot (x+y))$	$x=1, y=1, z=0$
9	$x+(y+(y \cdot (\bar{x}+y \cdot z \cdot (x+\bar{y}))))$	$x=0, y=1, z=0$	20	$y + \bar{y} \cdot (x \cdot \bar{z} + y \cdot z \cdot (x+y))$	$x=1, y=0, z=1$
10	$x+y \cdot \bar{x} \cdot (z+\bar{y} \cdot z \cdot (x+y))$	$x=0, y=1, z=1$	21	$(\bar{x}+y) \cdot (x+y \cdot \bar{z} \cdot (x+y))$	$x=0, y=1, z=0$
11	$\bar{x} \cdot y+y \cdot (\bar{x} \cdot z+\bar{y} \cdot z \cdot (x+y))$	$x=1, y=0, z=1$	22	$x \cdot \bar{y} + y \cdot (\bar{x} \cdot z + y \cdot \bar{z})$	$x=1, y=1, z=0$

Тема 5. Логические функции.

Чтобы задать логическую функцию, нужно указать, какие значения она принимает для всех комбинаций значений ее аргументов. Это можно сделать двумя способами: аналитическим, с помощью формулы и табличным (с помощью так называемой таблицы истинности). Например, логическую функцию четырех переменных можно задать формулой $F = x_2 \cdot x_1 + x_1 \cdot x_0$.

Эту же функцию можно описать с помощью таблицы истинности.

n	x_2	x_1	x_0	F
0	0	0	0	0
1	0	0	1	0
2	0	1	0	0

n	x_2	x_1	x_0	F
3	0	1	1	1
4	1	0	0	1
5	1	0	1	1
6	1	1	0	0
7	1	1	1	1

Для более компактной записи табличную форму представления можно заменить выражением типа $F(3,4,5,7)$, в котором указываются номера строк таблицы истинности, в которых функция F имеет значения 1.

Функцию, заданную в табличной форме, можно представить в аналитическом виде. Для этого есть несколько способов, одним из них является представление функции в *совершенной нормальной дизъюнктивной форме* (СНДФ). Для получения СНДФ нужно выписать из таблицы наборы переменных, для которых значения функции равны единице, при этом переменные, которые в наборе имеют значение 0, инвертируются. Затем все эти наборы объединяются с помощью операций сложения (дизъюнкцией). Для приведенного выше примера СНДФ имеет вид

$$F = \bar{x}_2 \cdot x_1 \cdot x_0 + x_2 \cdot \bar{x}_1 \cdot \bar{x}_0 + x_2 \cdot \bar{x}_1 \cdot x_0 + x_2 \cdot x_1 \cdot x_0$$

После упрощения функция принимает вид $x_2 \cdot x_1 \bar{x}_0 + x_1 \cdot x_0$.

Задание 5.

Записать логическую функцию в СНДФ и минимизировать ее.

№	$F(x_3, x_2, x_1, x_0)$	№	$F(x_3, x_2, x_1, x_0)$
1	0, 2, 7, 8, 10, 13, 15	15	0, 1, 2, 3, 4, 7, 11
2	2, 3, 5, 7, 10, 11, 15	16	3, 8, 9, 10, 11, 12, 15
3	2, 4, 6, 8, 10, 12, 14	17	1, 9, 10, 11, 12, 14, 15
4	4, 6, 7, 9, 11, 13, 15	18	2, 4, 6, 10, 13, 14, 15
5	6, 7, 8, 10, 12, 13, 14	19	0, 1, 4, 6, 8, 9, 14
6	5, 7, 8, 10, 13, 14, 15	20	3, 4, 6, 10, 11, 14, 15
7	1, 3, 5, 7, 10, 11, 14	21	0, 4, 6, 7, 10, 12, 14
8	0, 1, 4, 8, 12, 13, 14	22	1, 3, 4, 6, 9, 11, 12
9	1, 3, 7, 9, 11, 13, 15	23	4, 5, 6, 7, 9, 11, 15
10	0, 1, 4, 5, 11, 14, 15	24	4, 5, 7, 9, 11, 12, 13
11	2, 3, 6, 7, 8, 10, 15	25	0, 1, 4, 5, 12, 14, 15
12	2, 3, 6, 7, 12, 14, 15	26	3, 6, 7, 8, 10, 14, 15
13	3, 5, 7, 8, 9, 12, 13	27	1, 6, 9, 11, 13, 14, 15
14	4, 5, 7, 8, 10, 12, 14	28	4, 8, 11, 12, 13, 14, 15

Тема 6. Карты Карно

Другим способом минимизации логических функций является способ минимизации с использованием карты Карно, которая представляет собой квадрат или прямоугольник, разбитый на квадраты по числу возможных комбинаций значений переменных. Для функции трех переменных прямоугольник разбивается на восемь, а для четырех – на 16 частей. Карты Карно для случаев трех и четырех переменных показаны на рисунке.

	x_1	\bar{x}_1	
x_0	3	7	5
\bar{x}_0	2	6	4
	\bar{x}_2	x_2	\bar{x}_2

	\bar{x}_0	x_0	
\bar{x}_2	0	2	3
x_2	8	10	11
	\bar{x}_1	x_1	\bar{x}_1

Цифры внутри ячеек соответствуют номерам строк таблицы истинности. Например, в приведенной выше таблице истинности функции трех переменных в 7-ой строке значения всех трех переменных равны единице. В СНДФ ей соответствует слагаемое, в котором все три переменные перемножаются прямо (без инверсии). Соответствующая ячейка карты Карно с цифрой 7 внутри отражает это свойство.

Карта Карно позволяет быстрее перейти от табличного способа задания логической функции к аналитическому. Если заполнить карту Карно данными из приведенной выше таблицы истинности для функции трех переменных, она примет следующий вид:

	x_1	\bar{x}_1	
x_0	3	7	5
\bar{x}_0	2	6	4
	\bar{x}_2	x_2	\bar{x}_2

Единицы проставляются в ячейки, номера которых совпадают с номерами строк таблицы, в которых значение заданной функции равно 1.

Смежные клетки могут образовывать группы по 2, 4, 8 клеток. Для каждой такой группы в СНДФ

будет одно слагаемое, в котором в качестве множителей входят те переменные (или их инверсии), которые являются общими для группы.

Для приведенного здесь случая СНДФ имеет вид $F = \bar{x}_2 \cdot x_1 + x_1 \cdot x_0$ (результат, который был получен выше путем минимизации аналитического выражения для СНДФ, но гораздо более простым путем).

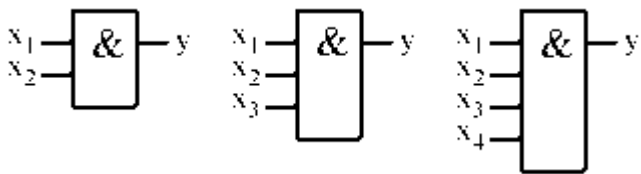
Задание 6.

Используя исходные данные из предыдущей темы написать выражение для заданной логической функции с помощью карты Карно.

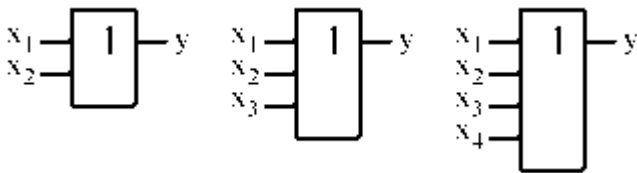
Тема 7. Комбинационные схемы.

Логическим элементом называется электрическая схема, выполняющая логические операции над входными данными, заданными в виде уровней напряжения, и возвращающая результат операции в виде выходного уровня напряжения. Высокий уровень напряжения (напряжение логической единицы) символизирует истинное значение операнда, а низкий (напряжение логического нуля) – ложное.

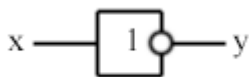
К числу логических операций, выполняемых логическими элементами относятся конъюнкция (логическое умножение, И), дизъюнкция (логическое сложение, ИЛИ), отрицание (НЕ) и сложение по модулю 2 (исключающее ИЛИ).



Графическое изображение элементов И с двумя, тремя и четырьмя входами.

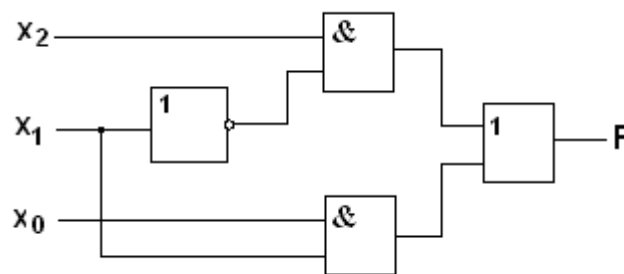


Графическое изображение элементов ИЛИ с двумя, тремя и четырьмя входами.



Графическое изображение инвертора (элемента НЕ) с одним входом и одним выходом.

Пример комбинационной схемы (реализация логической функции трех переменных, рассмотренной в качестве примера в Теме 5).



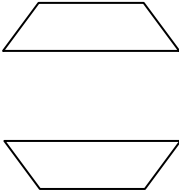

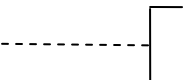
Задание 7.

Используя исходные данные из предыдущей темы составить функциональную схему, реализующую заданную логическую функцию.

Тема 8. Блок-схемы алгоритмов.

Основные элементы блок-схем алгоритмов.

Наименование	Обозначение	Функция
Блок начало-конец		Начало и конец алгоритма. Внутри фигуры записывается соответствующее действие
Блок вычислений		Выполнение одной или нескольких операций, обработка данных любого вида
Проверка условия		Отображает функцию переключательного типа с одним входом и двумя или более альтернативными выходами, из которых только один может быть выбран после вычисления условий, определенных внутри этого элемента.
Предопределенный процесс		Процесс, состоящий из одной или нескольких операций, определенных в другом месте программы (в подпрограмме, модуле).
Ввод/вывод данных		Ввод или вывод данных, имена переменных и вид действия указываются внутри фигуры
Цикл типа пере-счета		Цикл с заданным числом повторений. Внутри фигуры записывается параметр цикла, для которого указываются начальное и конечное значения, а также шаг изменения.

Наименование	Обозначение	Функция
Цикл с условием		Символ состоит из двух частей – начало и конец цикла – операции, выполняемые внутри цикла, размещаются между ними. Условия цикла и приращения записываются внутри символа начала или конца цикла – в зависимости от типа организации цикла.
Соединитель		Используется для обрыва линии и продолжения ее в другом месте
Комментарий		Используется для более подробного описания шага, процесса или группы процессов. Описание помещается со стороны квадратной скобки.

Задание 8.

В каждом варианте требуется разработать алгоритм и нарисовать его блок-схему.

1. В заданном массиве подсчитать количество четных и нечетных чисел:

234	122	213	245	279	220	334	303	217	160
-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

В зависимости от полученного результата вывести сообщение «четных больше», «нечетных больше» или «одинаковое количество».

2. Для заданного массива значений случайной величины вычислить среднее арифметическое и сформировать новый массив из квадратов разностей элементов заданного массива и средним арифметическим

28	22	21	15	27	28	33	19	21	22
----	----	----	----	----	----	----	----	----	----

3. В заданном массиве отыскать наибольший элемент, вывести его номер и значение

-23	12	21	15	-29	22	33	13	-2	16
-----	----	----	----	-----	----	----	----	----	----

4. Вычислить длины векторов А и В:

3	0	-1	5	9
---	---	----	---	---

1	4	4	-2	2
---	---	---	----	---

В зависимости от полученного результата вывести сообщение «длина А больше», «длина В больше» или «длины одинаковы».

5. Упорядочить заданный массив по возрастанию

18	22	12	13	27	28	33	19	20	12
----	----	----	----	----	----	----	----	----	----

6. Даны три вектора:

A	1	3	5
---	---	---	---

B	1	2	4
---	---	---	---

C	2	3	3
---	---	---	---

Определить, какой из векторов, A или B, ближе по направлению к вектору C.

7. В заданном массиве подсчитать количество чисел, делящихся без остатка на 3:

28	22	21	15	27	28	33	19	21	22
----	----	----	----	----	----	----	----	----	----

8. Вводится целое число N. Вычислить и вывести значение $Z=N!=1\cdot 2\cdot \dots \cdot N$. Учесть, что по определению $0!=1$. Предварительно сделать проверку знака N и если $N<0$ вычислений не проводить, а вывести сообщение " $N<0$ ". Провести расчеты для $N=4$.

9. Упорядочить заданный массив по убыванию

28	12	24	13	27	29	31	21	20	14
----	----	----	----	----	----	----	----	----	----

10. Вводятся три числа A, B и C. Определить действительные корни X_1 и X_2 квадратного уравнения $Ax^2+Bx+C=0$. Если действительных корней нет, вывести соответствующее сообщение. Для расчетов взять $A=1$, $B=-1$, $C=-6$.

11. В заданном массиве отыскать наименьший элемент, вывести его номер и значение.

21	11	21	15	22	15	28	13	23	16
----	----	----	----	----	----	----	----	----	----

12. В заданном массиве поменять местами наибольший и наименьший элементы

22	28	12	13	27	18	33	19	21	15
----	----	----	----	----	----	----	----	----	----

13. Найти наибольшее значение функции $y=x^2-A\cdot x$ на отрезке $[0, A]$, изменяя аргумент x от 0 до A с шагом h . Вывести значения x_{\max} и y_{\max} . Для расчетов взять $A=2$, $h=0.2$

14. В заданном массиве подсчитать количество чисел, кратных 5:

234	122	213	245	279	220	334	303	217	160
-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

15. Вводятся 6 чисел – координаты вершин треугольника: $x_1, y_1, x_2, y_2, x_3, y_3$. Вычислить его площадь. Для расчетов взять числа $(0,0,1,1,2,0)$.

16. Разделить заданный массив на два, один из которых содержит только отрицательные числа исходного массива, а другой – нулевые и положительные:

-23	22	21	45	-27	-20	-34	30	27	-16
-----	----	----	----	-----	-----	-----	----	----	-----

17. Для заданного массива вычислить среднее геометрическое Z и сформировать новый массив, вычитая из заданных значений величину Z :

28	22	21	15	27	28	33	19	21	22
----	----	----	----	----	----	----	----	----	----

18. Нормализовать заданный вектор, то есть сделать его длину равной единице. Для этого надо вычислить длину вектора и разделить на нее все элементы заданного вектора:

1	1	2	3	1
---	---	---	---	---

19. Даны три вектора:

A

2	3	2
---	---	---

B

1	2	5
---	---	---

C

2	3	4
---	---	---

Определить, какой из векторов, A или B, дальше по направлению отстоит от C.

20. Разложить заданный массив на два, первый состоит из четных значений исходного массива, второй – из нечетных:

234	122	213	245	279	220	334	303	217	160
-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

Часть II. Программирование на языке Visual Basic.

Тема 1. Язык Visual Basic. Типы переменных.

Переменные задаются *именами*, которые определяют области памяти, в которых хранятся их *значения*. Значениями переменных могут быть *данные* различных типов (целые или вещественные числа, символьные строки и т.д.).

Тип переменных определяется типом данных, которые могут быть значениями переменных. Значениями переменных числовых типов (Byte, Integer, Long, Single, Double) являются числа, логических (Boolean) — True или False, строковых (String) — последовательности символов.

Типы переменных:

Тип данных	Возможные значения	Объем занимаемой памяти	Приставка к имени
Byte	целые числа от 0 до 255	1 байт	byt
Integer	целые числа от -32768 до 32767	2 байта	int
Long	целые числа двойной длины	4 байта	lng
Single	Десятичные числа от 1.401298E-35 до 3.4022823E38	4 байта	sng
Double	Десятичные числа двойной точности от $1.9 \cdot 10^{-324}$ до $1.8 \cdot 10^{308}$	8 байт	dbl
Boolean	Логическое значение True или False	2 байта	bin
String	Строка символов	1 байт на символ	str
Currency	Число в денежном формате	8 байт	cur
Date	Дата от 1 января 1 00 г. до 31 декабря 9999г.	8 байт	dtm
Object	Ссылки на любой объект	4 байта	obj
Variant	Любые значения	> 16 байт	vnt

Математические функции в языке Visual Basic:

Функция	Возвращаемое значение
Sin(A)	синус числа A
Cos(A)	косинус числа A
Tan(A)	тангенс числа A
Atn(A)	арктангенс числа A
Sqr(A)	квадратный корень из числа A
Log(A)	логарифм числа A
Exp(A)	показательная функция числа A
Int(A)	наибольшее целое число, не превышающее число A
CInt(A)	целое число, ближайшее к числу A
Fix(A)	целое число, равное числу A без дробной части
Abs(A)	абсолютное значение числа A
Rnd	случайное число в интервале (0, 1)

Задание 1.

Написать операторы, результатом выполнения которых будет получение переменной y значения заданной функции для заданного значения аргумента x .

Указание: ввести вспомогательные переменные для вычисления формулы по частям.

Варианты задания.

№	функция	аргумент
1	$(x^3 - 8) \cdot \cos(2 \cdot \pi \cdot x - 3)$	0, 25
2	$3 \cdot (x^2 - 4) \cdot (x^2 - 1)$	-2,53
3	$x \cdot (x^3 - 2 \cdot x^2 - x + 2)$	0,5
4	$8 \cdot e^{2x-1} \cdot \sin(2 \cdot \pi \cdot (x+2))$	-1,1
5	$2 \cdot e^{2(x+1)} \cdot \cos(3 \cdot \pi \cdot (x+1))$	0, 25
6	$10 \cdot e^{2x} \cdot \sin(2 \cdot \pi \cdot (x+0.5))$	-0,4
7	$x \cdot (x^2 + x - 2)$	1,2
8	$5 \cdot e^{3x} \cdot \cos(3 \cdot \pi \cdot x + 5)$	0,25
9	$4 \cdot \cos(x^2) \cdot \sin(2 \cdot \pi \cdot x)$	0,28
10	$3 \cdot x^3 \cdot \cos(4 \cdot \pi \cdot x - 2)$	-0,9
11	$4 \cdot (2 \cdot x^2 - 5x + 3) \cdot (2 \cdot x^2 - 9 \cdot x + 10)$	0,3
12	$2 \cdot e^{-3x} \cdot \cos(2 \cdot \pi \cdot x + 1)$	0,37
13	$4 \cdot x^3 \cdot \sin(2 \cdot \pi \cdot x - 3)$	-1,25
14	$x^4 - 4 \cdot x^3 + x^2 - 6x$	-2,15
15	$3 \cdot x^2 \cdot \cos(3 \cdot \pi \cdot x - 1)$	0,6
16	$5 \cdot x^3 \cdot \cos(4 \cdot \pi \cdot x - 3)$	1,2
17	$5 \cdot x^3 \cdot \sin(\pi \cdot x - 3)$	-0,55
18	$4 \cdot e^{2x+1} \cdot \sin(3 \cdot \pi \cdot (x+1.5))$	-1,34
19	$3 \cdot x^2 \cdot \sin(3 \cdot (\pi \cdot x - 1))$	-0,95
20	$(x^2 - 3 \cdot x + 2) \cdot (x^2 - 7 \cdot x + 12)$	0,45

Тема 2. Программирование разветвляющихся вычислительных процессов.

Изменить последовательность выполнения операторов можно с помощью операторов ветвления. Одним из них является условный оператор **If ... Then**. Он имеет два формата записи.

Однострочный формат:

If условие Then Оператор1 [Else Оператор2]

(в квадратные скобки взята необязательная часть). Если *условие* истинно, *Оператор1* выполняется, в противном случае он пропускается и выполняется *Оператор2*, если он указан.

Блочный формат:

```
If условие Then
    Операторы1
[ Else
    Операторы2 ]
End If
```

Эта форма применяется, когда нужно при выполнении или не выполнении заданного условия выполнить не один, а несколько операторов.

Еще одним оператором ветвления является **Select Case**.

```
Select Case Выражение
Case Список_значений1
    Операторы1
Case Список_значений2
    Операторы2
    .....
[Case Else Список_значений3
    Операторы3 ]
End Select
```

Выражение может быть как именем переменной, так и арифметическим выражением, а параметры *Список_значений1*, *Список_значений2* и т.д. могут быть как одиночными числовыми значениями, так и списками – несколькими числами, разделенными запятыми.

Задание 2.

Написать операторы, результат выполнения которых определяется заданными условиями.

Варианты задания.

1. По заданным коэффициентам квадратного уравнения **A,B,C** определить, есть ли у этого уравнения действительные корни и вывести соответствующие сообщения.

2. Для заданного **X** вычислить значение функции **Y**, значение которой равно 0, если $1 < X \leq 2$ и 1 в противном случае.

3. Значениями символьных переменных **S1** и **S2** являются фамилии студентов. Вывести их на печать в алфавитном порядке.

4. Заданы координаты трех точек **A, B, C** на плоскости. Определить, образуют ли отрезки **AB** и **AC** прямой угол.

5. Определить, попадает ли точка **A** с координатами (X_A, Y_A) в круг радиусом **R** и центром в точке с координатами (X_C, Y_C).

6. Пенсии назначаются мужчинам в возрасте 60 лет, а женщинам с 55. Числовая переменная **AGE** содержит возраст человека, а символьная переменная **S** – значения «**M**» и «**J**». Написать оператор, выводящий сообщение «**Возраст пенсионный**» или «**Возраст не пенсионный**»

7. Функция **F(X)** определена формулой:

$$F(X) = \begin{cases} -1, & \text{для } X \leq -1, \\ X, & \text{для } -1 < X \leq 1, \\ 1, & \text{для } 1 < X. \end{cases}$$

Написать оператор **Select Case**, вычисляющий значение этой функции для заданного **X**.

8. Вводятся числовые коэффициенты квадратного уравнения $Ax^2+Bx+C=0$. Написать оператор **Select Case**, выводящий сообщения «**Корней нет**», «**Корень один**», «**Корней два**» в зависимости от значения дискриминанта.

9. Задана скорость спутника **V** км/сек. Написать оператор **Select Case**, выводящий сообщения «**Упадет на Землю**», «**Будет двигаться по орбите**», «**Улетит в космос**» в зависимости от значения скорости.

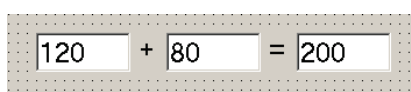
Тема 3. Операторы ввода и вывода.

В системе программирования Visual Basic имеется несколько способов организовать ввод и вывод данных в процессе выполнения программы.

1. Стандартный элемент управления **TextBox** можно использовать как для ввода числовых данных и текста, так и для вывода.

Пример:

На экранной форме размещены три элемента **Text1**, **Text2** и **Text3**. Первые два для ввода числовых данных с помощью клавиатуры, третье – для вывода результата, суммы вводимых чисел.



Фрагмент программного кода:

```
Text3.Text = Val(Text1.Text) + Val(Text2.Text)
```

2. Диалоговое окно **InputBox** позволяет ввести нужные данные в текстовое поле окна.

Пример:

St = InputBox (“Введите фамилию пользователя:”)

В результате выполнения этой команды появится диалоговое окно, куда можно ввести данные, которые после нажатия кнопки «ОК» будут присвоены переменной **St**.

При нажатии кнопки «Cancel» переменная **St** не изменится.

3. Диалоговое окно **MsgBox** позволяет отобразить в диалоговом окне краткое сообщение, или значение некоторого выражения. Обычно используется в процессе отладки программы для вывода промежуточных результатов. Имеется несколько видов окна.

4. Оператор печати **Print**. С помощью этого оператора можно выводить значения переменных и текстовые сообщения на экранную форму.

Чтобы вывести сообщение в нужное место экранной формы, перед применением оператором **Print** необходимо задать координаты точки вывода:

Me.CurrentX = отступ от левого края в пикселах
Me.CurrentY = отступ от верхнего края в пикселах

Можно также задать параметры шрифта, название, размер, цвет:

Me.Font.Size = 18 ‘ размер шрифта в пунктах
Me.Font.Name = “Arial” ‘ имя набора, например, **Times**
Me.ForeColor = vbRed ‘ или **vbBlue, vbBlack, vbGreen**

Тема 4. Оператор цикла типа пересчета.

Для многократного выполнения одного или нескольких операторов применяются операторы цикла. Оператор цикла типа пересчета:

For *Счетчик* = *Начальное_значение* **To** *Конечное_значение* [**Step** *Шаг*]

Операторы

Next [*Счетчик*]

Здесь *Счетчик* – параметр цикла, переменная целого или вещественного типа, *Начальное_значение* и *Конечное_значение* – числа, задающие границы интервала изменения параметра цикла, *Шаг* – шаг изменения параметра цикла, если он не указан, то по умолчанию он принимается равным 1. *Операторы* – один или несколько операторов, которые повторяются до тех пор, пока *Счетчик* не достигнет конечного значения.

Примеры:

1. Найти сумму ста элементов массива **A**.

```
S = 0
For i = 1 To 100
    S = S + A(i)
Next
Print S
```

2. Подсчитать количество элементов массива **A**, значение которых кратно 5.

```
N = 0
For i = 1 To 100
    if A(i) mod 5 = 0 then N=N+1
Next
Print N
```

Задание 4.

Написать программы, реализующие алгоритмы, рассмотренные в части I (Тема 8).

Тема 5. Действия с символьными строками.

В таблице приведены примеры использования некоторых функций для работы с символьными строками.

Выражение	Комментарий
$N = \text{Len}(S)$	Длина строки S присваивается числовой переменной N
$S1 = \text{Left}(S, n)$	Строковая переменная $S1$ получает n левых символов S
$S1 = \text{Right}(S, n)$	Строковая переменная $S1$ получает n правых символов S
$S1 = \text{Mid}(S, n, m)$	Строковая переменная $S1$ получает n символов S , начиная с m -го
$N = \text{Val}(S)$	Числовая переменная N получает значение числа из строки S
$S = \text{Str}(N)$	Строковая переменная S получает символьное представление числа N
$N = \text{Asc}(C)$	N получает значение ASCII-кода символа C
$C = \text{Chr}(N)$	C получает значение символа с кодом ASCII
$S1 = \text{Ucase}(S)$	Преобразование всех букв строки S в прописные
$S1 = \text{Lcase}(S)$	Преобразование всех букв строки S в строчные

Следует помнить, что коды заглавных и прописных букв разные. Коды заглавных букв от «А» до «Я» (без буквы «Ё») идут подряд от 192 до 223, а прописных – от 224 до 255 (т.е. для код каждой прописной буквы на 32 больше кода заглавной). Исключение – буква «Ё» (код 168) и «ё» (код 184).

Коды с 0 до 31 относятся не к символам, отображаемым на экране, а служат для обозначения управляющих команд. Так, например код 13 означает «возврат каретки», то есть переход к началу строки, а код 10 – переход на следующую строку.

Задание 5

Написать программы, реализующие алгоритмы, рассмотренные в части I (Тема 8).

Варианты задания.

1. Символьная строка содержит три числа, разделенных пробелами. Найти сумму этих чисел.
2. Дана символьная строка из цифр. Подсчитать количество цифр во введенной строке и вычислить их сумму.
3. Символьная строка содержит русские буквы. Подсчитать, сколько раз в этой строке появляются буквы «а» и «е» (как строчных, так и прописных).
4. Даны две символьные строки S1 и S2. Сформировать третью строку S3 из символов, которые есть как в S1, так и в S2.
5. Символьная строка S1 короче строки S2. Определить, входит ли S1 в S2 как подстрока.
6. Символьная строка состоит из букв и цифр. Сформировать новую строку из символов первой, оставив только буквы.
7. Строка содержит набор русских строчных букв. Сформировать новую строку, в которой те же буквы, но упорядоченные по алфавиту.
8. Символьная строка содержит три слов, разделенных пробелами. Переставить местами первое и третье слова.
9. Подсчитать, сколько раз в строке, содержащей буквы русского алфавита, встречается буква «а».

Тема 6. Графические возможности языка Visual Basic

На экранной форме или в графическом поле можно рисовать различные графические примитивы с использованием графических методов. Ниже приведены примеры использования этих методов. В качестве объекта *object*, куда выводятся графические примитивы, может служить сама форма (в этом случае имя объекта *Form* можно не указывать) или графическое окно *PictureBox*.

Наименование	Синтаксис и комментарии
Точка	object.Pset (X,Y), C X, Y – координаты точки, C – цвет.
Окружность	object.Circle (X, Y), R, C X, Y – координаты центра в выбранной системе координат, R – радиус, C – цвет.
Дуга окружности	object.Circle (X, Y), R, C, A, B X, Y – координаты центра, R – радиус, C – цвет. A, B – углы дуги в радианах. дуга строится против часовой стрелки от A к B.
Овал	object.Circle (X, Y), R, C,, K K – коэффициент сжатия овала. При $0 < K < 1$ сжатие по горизонтали, при $K > 1$ – по вертикали.
Отрезок линии	object.Line (X1,Y1) –(X2,Y2), C X1, Y1 – координаты точки начала отрезка, X2, Y2 – его конца, C – цвет.
Прямоугольник	object.Line(X1,Y1) –(X2,Y2), C, B X1, Y1 — координаты левой верхней вершины прямоугольника, X2, Y2 — координаты правой нижней вершины, C – цвет.
Прямоугольник закрашенный	object.Line (X1, Y1) –(X2, Y2), C, BF X1, Y1 — координаты левой верхней вершины прямоугольника, X2, Y2 — координаты правой нижней вершины, C – цвет.
Вывод строки символов	object.Print [output] Вывод осуществляется от последней построенной точки изображения. Для указания точки вывода можно использовать метод Pset(X,Y).
Масштабирование окна вывода	object.Scale (X1, Y1) – (X2, Y2) (X1, Y1) и (X2, Y2) – «мировые» координаты выводимого изображения, левой верхней и правой нижней вершины окна соответственно

Пример. Построить график функции $y = e^x \cdot \sin(2\pi x)$ на интервале $[-1, 1]$ в графическом окне Picture1. Запускается процедура по нажатию кнопки Command1.

```
Private Sub Command1.Click ( )
Const PI = 3.141529
```

```

Picture1.Scale (-1, 2) - (1, -2) ' Задание масштаба
Picture1.Line (-1, 0) - (1, 0) ' Построение координатных осей
Picture1.Line (0, -2) - (0, 2)
For I = -1 To 1 Step 0.25 ' Вывод надписей на оси X
    Picture1.PSet (I, 0)
    Picture1.Print I
Next I
For I = -2 To 2 ' Вывод надписей на оси Y
    Picture1.PSet (0, I)
    Picture1.Print I
Next I
For X = -1 To 1 Step 0.002 ' Вывод точек графика функции
    Y = exp(X) * sin(2 * PI * X)
    Picture1.PSet (X, Y)
Next X

```

Задание 6

Построить в графическом окне график функции на заданном интервале.

Варианты задания.

№	функция	интервал
1	$3 \cdot x^2 \cdot \cos(3 \cdot \pi \cdot x - 1)$	[0, 1]
2	$10 \cdot e^{2x} \cdot \sin(2 \cdot \pi \cdot (x + 0.5))$	[-1, 1]
3	$x \cdot (x^2 + x - 2)$	[-2, 2]
4	$5 \cdot e^{3x} \cdot \cos(3 \cdot \pi \cdot x + 5)$	[0, 2]
5	$4 \cdot \cos(x^2) \cdot \sin(2 \cdot \pi \cdot x)$	[-1, 2]
6	$3 \cdot x^3 \cdot \cos(4 \cdot \pi \cdot x - 2)$	[-1, 1]
7	$x \cdot (x^3 - 2 \cdot x^2 - x + 2)$	[-1, 2]
8	$2 \cdot e^{-3x} \cdot \cos(2 \cdot \pi \cdot x + 1)$	[0, 3]
9	$4 \cdot x^3 \cdot \sin(2 \cdot \pi \cdot x - 3)$	[-2, 2]
10	$x^4 - 4 \cdot x^3 + x^2 - 6x$	[-3, 1]
11	$8 \cdot e^{2x-1} \cdot \sin(2 \cdot \pi \cdot (x + 2))$	[-2, 1]
12	$5 \cdot x^3 \cdot \cos(4 \cdot \pi \cdot x - 3)$	[-2, 2]
13	$3 \cdot (x^2 - 4) \cdot (x^2 - 1)$	[-3, 3]
14	$4 \cdot e^{2x+1} \cdot \sin(3 \cdot \pi \cdot (x + 1.5))$	[-2, 3]
15	$3 \cdot x^2 \cdot \sin(3 \cdot (\pi \cdot x - 1))$	[-1, 1]
16	$(x^2 - 3 \cdot x + 2) \cdot (x^2 - 7 \cdot x + 12)$	[0, 4]
17	$7 \cdot e^{3x} \cdot \sin(2 \cdot \pi \cdot (x - 2.5))$	[-2, 2]
18	$2 \cdot x^2 \cdot \cos(3 \cdot \pi \cdot x + 2)$	[-1, 2]

Тема 7. Работа с файлами последовательного доступа.

Для открытия текстового файла используется оператор

Open *имя_файла* **For** *тип_доступа* **As** # *номер*

Здесь *имя_файла* – имя открываемого файла с указанием пути доступа к нему, *тип_доступа* – тип доступа (**Input** – для чтения, **Output** – для вывода, **Append** – для добавления), *номер* – порядковый номер от 1 до 255. В дальнейшем при работе с открытым файлом указывается не имя, а его порядковый номер.

Чтобы закрыть файл используется оператор **Close** (*номер*).

Для считывания одной строки из открытого для чтения текстового файла используется оператор **Input**:

Input # *номер*, *имя_переменной*

где *имя_переменной* – имя переменной типа **String**.

При открытии файла в его начало автоматически устанавливается так называемый «файловый указатель». После команды **Input** он перемещается к началу следующей строки. После считывания всех строк он устанавливается в конец файла. При попытке выполнить еще одну команду **Input** в этом случае будет выдано сообщение об ошибке (код 62).

Для того, чтобы избежать аварийного завершения программы можно использовать логическую функцию **EOF**(*номер*), где *номер* – номер открытого файла. Например, для считывания данных из текстового файла А.TXT в строковый массив **MAS**(*i*) можно использовать цикл типа **Do Until ... Loop**:

```
i=0
Do Until EOF (1)
    i=i+1
    Input # 1, MAS(i)
Loop
```

Запись данных в файл, открытый для записи, осуществляется командой

Print #*номер*, *список_вывода*

Здесь *список_вывода* – одна или несколько переменных (или строки символов, взятые в кавычки), разделенных либо запятыми (,), либо точками с запятой (;). В первом случае выводимые значения будут записаны в виде одной строки и разделены пробелами, во втором – выведены без разделителя, слитно. Если поставить точку с запятой в конец списка вывода, то следующая порция данных, выводимая оператором, будет дописана в ту же строку.

Варианты задания.

1. Текстовый файл содержит несколько строк с фамилиями студентов. Создать новый файл, в котором будут те же фамилии, но расположенные в алфавитном порядке.

2. Текстовый файл содержит одну строку, в которой записаны фамилии, разделенные пробелами. Создать новый файл, в котором будут те же фамилии, но расположенные в разных строках.
3. Текстовый файл содержит несколько строк, в каждой из которых записано целое число. Разделить этот файл на два, записав в один только четные числа, а в другой – нечетные.
4. Текстовый файл А содержит две строки с текстом, а файл В – одну строку. Создать файл С, в котором будут эти же три строки, но строка из файла В должна оказаться между строками из файла А.
5. Текстовый файл А содержит текст из заглавных (прописных) русских букв. Создать файл В, в котором будет тот же текст, но из строчных букв.
6. Текстовый файл А содержит текст с русскими и латинскими символами. Создать новый файл В, в котором из этого текста будут удалены латинские символы.
7. Текстовый файл содержит три строки символов. Создать новый файл, в котором эти строки будут записаны в обратном порядке.
8. В текстовых файлах А и В записаны два массива по N чисел в каждом, каждое число в отдельной строке. Создать файл из N строк, в каждой из которых записана сумма соответствующих чисел из файлов А и В.
9. В файле А построчно записаны фамилии студентов. В файле В – две фамилии из тех, которые приведены в А. Создать файл, в который записать все фамилии из файла А кроме тех, что указаны в файле В.
10. Текстовый файл А содержит несколько строк с фамилиями студентов, а В – строки с их именами. Создать новый файл, в котором строки будут содержать фамилии тех же студентов вместе с их именами.

Тема 8. Файлы прямого доступа.

Прямой или произвольный (*Random*) доступ для ввода или вывода применяется в случае, когда записи в файле имеют фиксированную длину. Открывается файл для произвольного доступа с помощью оператора

Open имя_файла For Random [Access доступ] [Lock] As # номер [Len = длина_записи]

В квадратных скобках указаны необязательные параметры. Параметр **Access** позволяет задать тип доступа к открываемому файлу. Если он не ука-

зан, то по умолчанию используется значение **ReadWrite** (возможно как чтение, так и запись).

Параметр **Lock** позволяет задать права доступа к открытому файлу и указывается, если один и тот же файл может использоваться несколькими пользователями или приложениями. Он может принимать следующие значения:

1) **Shared** (общий) – файл может использоваться всеми процессами для считывания и для записи;

2) **LockRead** (закрит для чтения) – никакой другой процесс не может считывать данные из файла;

3) **LockWrite** (закрит для записи) – никакой другой процесс не может записывать данные в файл;

4) **LockReadWrite** (закрит для чтения и записи) – никакой другой процесс не может ни считывать, ни записывать данные в файл;

Параметр **Len** задает длину одной записи в байтах

Для чтения данных используется оператор **Get**:

Get # номер , номер_записи, переменная

Для записи – оператор **Put**:

Put # номер , номер_записи, переменная

В качестве примера рассмотрим следующую задачу. Необходимо создать файл с именем *Spisok.dat*, в котором каждая запись состоит из двух полей. Первое поле для записи фамилии (типа **String** длиной 20 символов), вторая для записи личного номера (типа **Ineger**).

Для того, чтобы в одной записи сохранить значения данных различных типов используем в программе пользовательский тип данных, которому присвоим наименование *Person*. Кроме того, опишем массив *Mass* из 100 элементов типа *Person*.

```
Type Person
    Fam As String * 20
    Nom As Integer
End Type
Dim Mass (1 to 100) As Person
```

В процедуре, которая открывает файл для записи сформированного массива с фамилиями и номерами будут следующие операторы

```
Private Sub Command1_Click()
Open "Spisok.dat" For Random As #1 Len = 22
For i =1 to 100
    Put #1, i , Mass(i)
Next i
Close #1
End Sub
```

Предположим, что в другой процедуре нужно открыть файл Spisok.dat, прочитать записи и вывести значения полей каждой записи на печать.

```

Private Sub Command2_Click()
Dim zap As Person
Dim nomer As Integer
Open "Spisok.dat" For Random As #1 Len = 22
nomer = 1
Do until EOF(1)
    Get #1, nomer, zap
    Print zap.Fam, zap.Nom
    nomer = nomer + 1
Loop
Close #1
End Sub

```

Варианты задания.

1. Написать программу, которая позволяет ввести с клавиатуры результаты экзаменов по трем дисциплинам и сохранить результаты в файле прямого доступа. Каждая запись этого файла имеет поля с фамилией студента, и тремя оценками.

2. Каждая запись файла прямого доступа **AVTO.DAT** имеет два поля. Первое типа **String*20** содержит название марки автомобиля, второе – типа **Integer** – его цену. Написать программу, в результате работы которой данные из файла считываются и выводятся на экранную форму.

3. Файлы прямого доступа **A** и **B** имеют записи с полями **String*10** и **Integer**. В первом файле – результаты экзамена по информатике. Во втором – результаты для тех же студентов по математике. Написать программу, в результате работы которой по данным из этих файлов создается третий файл с именем **C**, в котором приводятся результаты двух экзаменов.

4. Файл прямого доступа **A.DAT** содержит список студентов и оценки, полученные на экзамене по информатике. Записи с полями **String*12** и **Integer**. Написать программу, в результате которой создается новый файл **B.DAT**, в котором содержится список тех же студентов, но имеющих оценки выше средней по данным из **A.DAT**.

5. Написать программу для ввода сведений об участниках соревнований и занятых местах. Записать введенные данные в файл прямого доступа.

Указание: использовать пользовательский тип данных типа запись с полями: фамилия, имя, дата рождения, занятое место.

Литература

1. Симонович С.В. Информатика: базовый курс. Учебное пособие.–М., 2010
2. Гуда Н. и др. Информатика. Общий курс: Учебник.-4-е изд.– М., 2011
3. Аверьянов Г.П., Дмитриева В.В. Современная информатика: Учебное пособие.– М.: МИФИ, 2011.
4. Андреева Т.И., Пичугин А.А. (№1124) Информатика. Информатика и ИТ: пособие по выполнению практических занятий и комплексных домашних заданий для студ. 1 курса дневного обучения.– 2012.
5. Андреева Т.И., Кишенский С.Ж., Пичугин А.А. Информатика: пособие по изучению дисциплины, выполнению контрольных работ и варианты заданий. Эл. версия НТБ.– 2011.
6. Агеев В.Н. Программирование и основы алгоритмизации. Конспект лекций.– М., 2012.
7. Культин Н.Б. Visual Basic. Освой на примерах.– М., 2012.