

С. Г. Никитенко

СВОБОДНОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ BASIC-256 для школы

Санкт-Петербург

«БХВ-Петербург»

2011

УДК 681.3.068(075.3)+800.92Basic-256
ББК 32.973.26–018.1я721
Н62

Никитенко С. Г.

Н62 Свободное программное обеспечение. BASIC-256
для школы. — СПб.: БХВ-Петербург, 2011. — 224 с.:
ил. — (ИиИКТ)

ISBN 978-5-9775-0740-0

Для свободно распространяемой кроссплатформенной среды BASIC-256 рассматриваются: интерфейс пользователя, основные операторы, правила написания программ, примеры решения типовых школьных задач из курса информатики и ИКТ, методика решения заданий ЕГЭ. Подробно рассмотрены часто встречающиеся на олимпиадах различного уровня темы: ввод/вывод числовых и строковых данных посредством текстовых файлов и построение графиков функций. Приведены задания для тренировки учащихся, проведения зачетных работ и выполнения проектов. Все зачетные работы и проекты ориентированы на выполнение по строго индивидуальным заданиям. Также книга содержит справочную информацию по BASIC-256.

Для образовательных учреждений

УДК 681.3.068(075.3)+800.92Basic-256
ББК 32.973.26–018.1я721

ISBN 978-5-9775-0740-0

© Никитенко С. Г., 2011
© Оформление, издательство "БХВ-Петербург", 2011

Оглавление

Введение. Свободное программное обеспечение в школе	7
Глава 1. Начальные сведения о BASIC-256	13
1.1. Что такое BASIC-256?	13
1.2. Интерфейс пользователя BASIC-256	14
1.3. Особенности BASIC-256 v.0.9.6p для Win32 и v.0.9.6.48 для ALT Linux 5.0.2 Школьный	17
Глава 2. Основные операторы BASIC-256	21
2.1. Операторы обработки числовых данных	21
2.2. Операторы обработки строковых данных	26
2.3. Управляющие операторы BASIC-256	30
2.3.1. Операторы формирования условий	30
2.3.2. Операторы анализа условий (операторы ветвления)	32
2.3.3. Операторы управления циклами	34
2.3.4. Операторы управления переходами	36
2.3.5. Операторы для обслуживания подпрограмм	38
2.4. Операторы для обработки дат и времени	40
Глава 3. Графика в BASIC-256	41
3.1. Операторы управления графическим экраном	41
3.2. Операторы формирования графических примитивов в графическом окне	44
Глава 4. Операторы для работы с файлами	47
Глава 5. Порядок разработки и отладки программ в среде BASIC-256	49
Вариант 1	51
Вариант 2	53

Вариант 3	53
Вариант 4	53

Глава 6. Примеры программ для иллюстрации типовых школьных задач57

6.1. Вывод в окно текста.....	57
6.2. Формирование подвижных объектов	58
Пример 6.2.1	59
Пример 6.2.2	60
6.3. Вычисление таблицы функции	62
Порядок выполнения зачетной работы	62
Пример 6.3.1	63
6.4. Поиск данных в массивах по заданным условиям	64
Пример 6.4.1	65
Пример 6.4.2	67
6.5. Задачи сортировки	69
Пример 6.5.1	71
Пример 6.5.2	74
Пример 6.5.3	76
6.6. Формирование подвижных изображений на текстовом экране	79
Пример 6.6.1	79
6.7. Ввод/вывод данных посредством текстовых файлов.....	81
Пример 6.7.1	82
Пример 6.7.2	83

Глава 7. Примеры решения графических задач в BASIC-25687

Пример 7.1	87
Пример 7.2	89
Пример 7.3	92
Пример 7.4	94
Пример 7.5	95
Пример 7.6	97
Пример 7.7	98
Пример 7.8	100
Пример 7.9	101

Глава 8. Обработка строковых данных111

Пример 8.1	111
Пример 8.2	114

Глава 9. Решение заданий ЕГЭ средствами BASIC-256.....	117
Задача C1-2011D	118
Задача C2-2011D	120
Задача C4-2011D	121
Глава 10. Задачник по программированию	125
10.1. Освоение операторов языка и типов данных BASIC-256.....	125
Задача 10.1.П-1.....	126
Задача 10.1.П-2.....	127
Задача 10.1.П-3.....	128
Задача 10.1.П-4.....	129
Задание 10.1.1.....	129
Задание 10.1.2.....	129
Задание 10.1.3.....	130
Задание 10.1.4.....	131
Задание 10.1.5.....	131
Задание 10.1.6.....	131
Задание 10.1.7.....	132
Задание 10.1.8.....	132
Задание 10.1.9.....	133
Задание 10.1.10.....	133
Задание 10.1.11.....	134
Задание 10.1.12.....	135
Задание 10.1.13.....	135
Задание 10.1.14.....	136
Задание 10.1.15.....	136
Задание 10.1.16.....	137
Задание 10.1.17.....	138
Задание 10.1.18.....	138
Задание 10.1.19.....	139
Задание 10.1.20.....	140
Задание 10.1.21.....	141
Задание 10.1.22.....	141
Задание 10.1.23.....	141
Задание 10.1.24.....	142
Задание 10.1.25.....	143
Задание 10.1.26.....	144
Задание 10.1.27.....	144
Задание 10.1.28.....	145
Задание 10.1.29.....	146
Задание 10.1.30.....	146

10.2. Циклы с параметром	147
10.3. Задачи поиска	150
10.4. Задачи сортировки	153
10.5. Моделирование математических, экономических и физических зависимостей	158
Заключение	165
Приложение 1. Справочник по BASIC-256.....	167
Приложение 2. Ответы на задачи раздела 10.5.....	179
Литература.....	221

ВВЕДЕНИЕ

Свободное программное обеспечение в школе

Впервые идея о необходимости разработки свободной отечественной операционной системы (ОС) для использования в учреждениях бюджетной сферы на государственном уровне была высказана в послании Президента Федеральному собранию РФ в 2006 г. В конце 2007 г. Правительство России приняло решение о выделении 100 миллионов долларов для Министерства образования РФ на закупку комплекта лицензионного программного обеспечения (ПО) и поставку его в 52 000 школ России с лицензией до 31.12.2010 г. и о разработке комплекта свободного ПО (СПО) для школ¹. Исполнение проекта было поручено по конкурсу ряду фирм: группе компаний "Армада", ООО "Альт Линукс", ОАО "Линукс Инк".

В 2008 г. в 1107 школах Татарстана, Пермского края и Томской области и еще в 770 школах из других регионов были установлены школьные версии ОС ALT Linux, проведено обучение педагогов и начата опытная эксплуатация комплекта СПО в учебном процессе и документообороте. В этом же году Министерством информационных технологий и связи РФ разработана "Концепция развития, разработки и использования свободного программного обеспечения в РФ". В марте 2009 г. Минкомсвязи опубликовало план перехода государственных органов на СПО. Летом 2009 г. во все школы России был поставлен комплект ПО "Первая Помощь 2.0" из 27 компакт-дисков и DVD, в состав которого входило 9 дисков с различными версиями школьных дистрибутивов ОС ALT Linux и с дидактическими материалами.

¹ Распоряжение Правительства РФ № 1147-р от 18.10.2007 г.

Экономическое положение разных регионов, возможности по обновлению парка компьютеров в школах очень разные. По данным автора (из бесед с учителями информатики и заместителями директоров по информационным технологиям московских школ) многие школы вообще не озабочиваются проблемой перехода на СПО: "Директор обещал, что найдет нужные 800 долларов на лицензионную Windows и прочие программы". Из бесед с директорами сельских школ Рязанской области: "Денег нет и не будет. Альтернативы переходу на Linux нет". Это крайние точки зрения. На самом деле есть школы — обладатели грантов приоритетного национального проекта "Образование" и других проектов, которые приобрели компьютерные классы с предустановленной лицензионной версией Windows 7/Vista. Есть школы, которые имеют по два и более компьютерных класса. Причем самые новые из них укомплектованы лицензионной Windows 7, а в остальных стоит Windows XP с истекшим 31.12.2010 г. сроком лицензии. Вдобавок практически во всех регионах есть школы с классами, оснащенными компьютерами фирмы Apple. На них в качестве базовой ОС установлена одна из версий лицензионной MAC OS X, в дополнение к которой обязательно установлена лицензионная Windows XP/Vista/7.

В упоминавшемся Распоряжении Правительства РФ рекомендуется дать школе возможность выбора. Возможные варианты представлены далее. Вариативность обеспечивается еще и тем, что наиболее популярные свободные программы (типа свободного офиса OpenOffice.org) имеют версии и для Windows, и для Linux, и для MAC OS X:

- лицензионные Windows XP/Vista/7, Office, графика, среды программирования;
- лицензионные Windows/MAC OS X, свободные Office, графика, среды программирования;
- полный переход на свободное ПО.

Возможны и промежуточные решения. Так, компьютеры бухгалтерии и администрации, использующие специальные программы, не имеющие аналогов в Linux (например, программы печати аттестатов, базы данных для отчетов по ЕГЭ), остаются с ОС

Windows, а компьютерные классы и учебные кабинеты, обеспечивающие учебный процесс, переводятся на Linux полностью или сохраняют лицензионную Windows или MAC OS X с необходимым набором свободного ПО. Возможен вариант, когда в современных компьютерных классах, обеспечивающих учебный процесс, устанавливаются и Windows, и Linux. Учащихся физико-математических, экономических, информационно-технологических классов есть прямой смысл знакомить и с Windows, и с Linux, и с соответствующими наборами приложений. Профессиональное образование в вузах и колледжах уже несколько лет постепенно переводится на двух- или даже трехплатформенное обучение.

Одна из главных проблем перехода на СПО в школе в сложившихся условиях — обеспечение методического единства в преподавании всех дисциплин, связанных с применением компьютеров в условиях использования разных операционных систем и комплектов приложений. Положение облегчается тем, что большинство популярных программ, включая среды программирования, имеют свободные аналоги для работы во всех перечисленных ОС: Windows, Linux, Mac OS X. Данные о соответствии сред программирования, использовавшихся в школах для изучения раздела "Основы алгоритмизации и элементы программирования" курса информатики и ИКТ, их свободным аналогам в составе школьных дистрибутивов ALT Linux и для ОС Windows приведены в табл. В1.

В этой книге мы предлагаем в качестве инструментального средства для ознакомления с элементарными основами программирования в курсе информатики и ИКТ использовать среду BASIC-256. Выбор обоснован следующими обстоятельствами. BASIC-256 уже поставлен во все школы России в составе школьных дистрибутивов ALT Linux. Последняя серия дистрибутивов ALT Linux 5.0.2 Школьный представлена в свободном доступе на сайте фирмы и содержит работоспособные версии приведенных в табл. В1 сред программирования. Существует версия BASIC-256 и для Windows. Поэтому есть возможность в школьных компьютерных классах с тремя упомянутыми ОС (Windows, ALT Linux, Mac OS X) обеспечить методическое единство в преподавании основ алгоритмизации и программирования за счет использова-

Таблица В1. Среды программирования, поставлявшиеся в школы, и их свободные аналоги

Назначение приложения	Среды, поставлявшиеся в школы в 1985–2009 гг.	Дистрибутивы Alt Linux 4.0–5.0.2 Школьный	Свободные или бесплатные приложения Windows
Школьный алгоритмический язык	Кумир	Кумир	
Исполнители для начального обучения основам алгоритмизации	Кенгуренок Пылесосик Муравей Машинист Кукарача (Таракан) в составе пакета Роботландия	Scratch	
LOGO, начальное обучение основам алгоритмизации и программирования	LogoWriter LogoMiry DLOGO	KTurtle	KTurtle, DLOGO
BASIC	QBasic Quick Basic Turbo Basic	BASIC-256	BASIC-256, Small BASIC
Pascal	Borland Pascal 7.0 Turbo Pascal 7.0	Free Pascal	Free Pascal Pascal ABC BlackBox Component Builder
Visual Basic	Visual Basic 2005 в составе Visual Studio 2005	Gambas	Gambas
Delphi (Object Pascal)	Borland Delphi 2006 в составе Borland Developer Studio 2006	Lazarus	Lazarus Borland Delphi 7 Lite
C/C++, Java	Borland Developer Studio 2006 Visual C++ 2005 в составе Visual Studio 2005	Eclipse Code::Blocks IDE	Eclipse Code::Blocks IDE

ния одного и того же продукта. Вдобавок, BASIC-256, пожалуй, ближе всех к популярной в прошлые годы среди учителей информатики среде программирования QBasic от Microsoft — ее использовали от 60 до 80 % учителей информатики вплоть до конца 2010 г.

Далее для BASIC-256 рассматриваются: интерфейс пользователя, основные операторы, правила написания программ, примеры решения типовых школьных задач из курса информатики и ИКТ, методика решения заданий ЕГЭ, приведены также задания для тренировки учащихся, проведения зачетных работ и проектов. Все зачетные работы и проекты ориентированы на выполнение их строго по индивидуальным заданиям — никаких традиционных среди многих учителей контрольных работ с 2–4 вариантами заданий автор, имеющий 10-летний опыт преподавания информатики в различных школах, ПТУ и колледжах г. Рязани, не признает: 30 учащихся — 30 пусть однотипных, но разных вариантов заданий по каждой зачетной теме.

ГЛАВА 1

Начальные сведения о BASIC-256

1.1. Что такое BASIC-256?

BASIC-256 — свободная версия популярного среди учителей информатики интерпретатора языка BASIC, используемая в соответствии с правилами GNU General Public License. Авторское право на эту программу принадлежит с 2006 г. группе The BASIC-256 Team. Первоначальное название языка — KidBasic.

Разработаны версии интерпретаторов языка для ряда платформ, в частности, для Linux и для 32-битных версий Windows. В состав серии дистрибутивов ALT Linux 5.0.2 Школьный (январь 2011 г.) включен BASIC-256 версии 0.9.6.58 со встроенным справочником в переводе В. Чёрного и С. Ирюпина. В июне 2010 г. для 32-битных версий Windows, включая соответствующие версии Windows Vista и Windows 7, разработан BASIC-256 версии 0.9.6р.

Сайт группы разработчиков (англоязычный) размещен по адресу [1]. Дистрибутив BASIC-256 версии 0.9.6р можно загрузить со страницы [2]. Руководство по основным структурам данных и операторам BASIC-256 версии 0.9.5 на английском языке доступно на странице [3], а на немецком языке — на странице [4].

В BASIC-256 версии 0.9.6р имеется встроенный англоязычный справочник RU-BASIC-256 Reference 0.9.5x, вызываемый по клавише <F1> [5]. Справочник [5] и руководство [7] в наибольшей степени соответствуют взятым за основу в пособии версиям

0.9.6p (2010-06-10) для Windows и 0.9.6.58 (2011-01-07) для ALT Linux 5.0.2 Школьный.

1.2. Интерфейс пользователя BASIC-256

После установки русской версии BASIC-256 (например, v.0.9.6p) запуск ее выполняется из меню Windows командой **Пуск | Программы | BASIC-256 | BASIC-256**. Открывшееся окно программы (рис. 1.1) имеет обычный для приложений Windows вид. Заголовок окна имеет стандартную структуру. Меню программы содер-

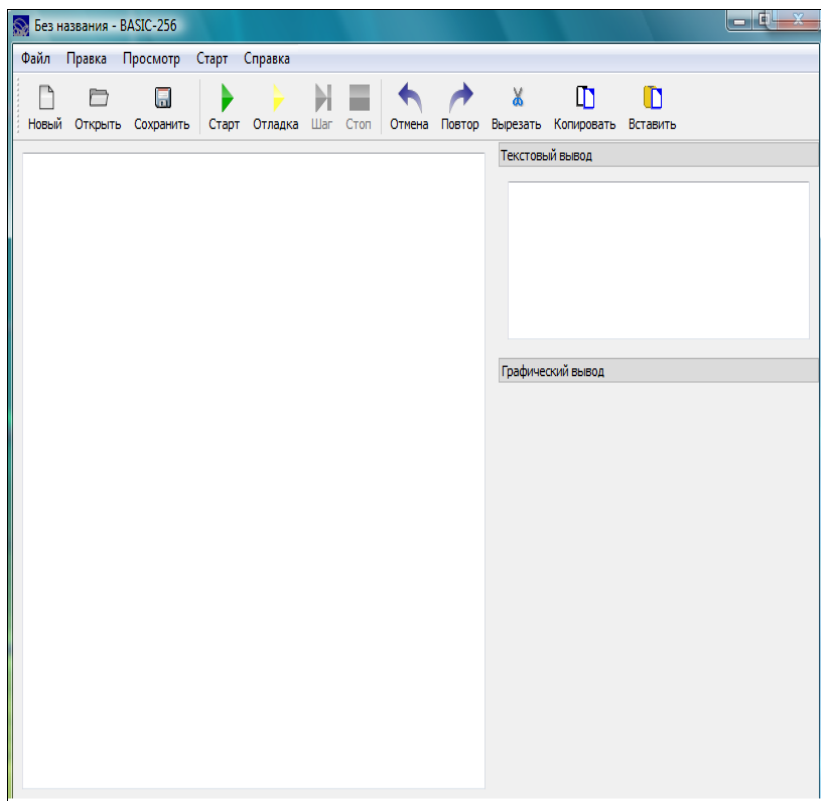


Рис. 1.1. Окно русских версий BASIC-256

жит пункты **Файл, Правка, Просмотр, Старт, Справка**. Смысл любого пункта интуитивно понятен при открытии его списка команд.

Расположенная под строкой меню панель инструментов содержит набор кнопок быстрого вызова наиболее употребительных команд: **Новый, Открыть, Сохранить, Старт, Отладка, Шаг, Стоп, Отмена, Повтор, Вырезать, Копировать, Вставить**. Кнопки формируют команды, позволяющие выполнять вызов программ из файлов, их сохранение, запуск, отладку и редактирование.

Рабочая область окна BASIC-256 выглядит несколько необычно по сравнению с более ранними версиями BASIC, такими как QBasic или Quick Basic. В формируемой по умолчанию рабочей области окна выделены три поля:

- слева находится основное поле — окно текста программы (окно редактора), управляемое редактором текста программы;
- справа в общем вертикальном столбце сформированы два поля (окна) исполнителя программы, управляемые отладчиком и средой исполнения:
 - верхнее поле называется **Текстовый вывод** и представляет собой экран исполнителя в текстовом режиме. На нем будут отображаться результаты вычислений и обработки данных, представляемые в виде текстовых (символьных) строк и выводимые чаще всего оператором `print`, а также сообщения об ошибках, формируемые самим интерпретатором на этапе синтаксического контроля программы пользователя;
 - нижнее поле называется **Графический вывод** и представляет собой графический экран исполнителя. На нем отображаются результаты выполнения графических операторов программы. По умолчанию графический экран имеет размеры 300×300 пикселей.

Характеристики текстового экрана (число строк, число символов в строке) зависят от типа и размера шрифта, используемого для вывода текстовой информации. Характеристики шрифта задаются специальными операторами BASIC-256 или командами меню.

Каждый из экранов очищается отдельной командой. Текстовый экран очищается обычной командой очистки текстового экрана `cls` (clear screen, очистить экран). Графический экран очищается специальной командой `clg` (clear graphic screen, очистить графический экран).

Окно текста программы присутствует на экране всегда. Экраны исполнителя (окно текста и окно графики) можно вызывать на экран или убирать с экрана командами меню **Просмотр | Окно текста** и **Просмотр | Окно графики**. По умолчанию эти функции включены — в соответствующем пункте меню стоит флажок ("галочка"). Одна подача команды включает функцию, повторная подача той же команды функцию отображения соответствующего окна на экране выключает. Отладчик позволяет при отладке программы вызвать на экран дополнительное окно для наблюдения за переменными программы — окно просмотра переменных. Оно вызывается на экран или убирается командой меню **Просмотр | Окно просмотра переменных**.

Размер шрифта, которым отображаются текст программы в окне редактора и результаты вычислений в окне текста, можно дискретно изменять командами меню **Просмотр | Размер шрифта | Мелкий / Средний / Большой / Огромный**. По умолчанию после загрузки BASIC-256 включен средний размер шрифта. Характеристики текстового экрана в режиме мелкого шрифта определяются экспериментально как 11 строк примерно по 40 символов \times . Большинство шрифтов Windows не моноширинные, т. е. различные символы имеют различную ширину. Поэтому ширину строки в числе символов можно определить только приблизительно для некоторого символа со средней шириной. Если число выводимых строк превышает 9, в окне текста появляется вертикальная полоса прокрутки. Если выводимая строка превысит ширину окна текста, появится горизонтальная полоса прокрутки.

Запуск программы, текст которой подготовлен в окне редактора, осуществляется следующими способами: командой меню **Старт | Старт**; кнопкой **Старт** панели инструментов; клавишей `<F5>`.

ПРИМЕЧАНИЕ

Клавиша `<F5>` используется для запуска приложений во многих реализациях BASIC.

1.3. Особенности BASIC-256 v.0.9.6p для Win32 и v.0.9.6.48 для ALT Linux 5.0.2 Школьный

Программа на языке BASIC-256 представляет собой последовательность строк кода. В каждой строке допускается один оператор или два оператора, разделенные двоеточием. Операторы в рассматриваемых версиях можно писать и строчными, и прописными латинскими буквами. То есть операторы `print x`, `PRINT x` и `Print x` будут выполняться одинаково. Функции `sin(x)`, `SIN(x)` и `Sin(x)` также будут вычисляться одинаково.

Имена переменных могут содержать только латинские буквы и цифры и должны начинаться с буквы. Причем, в именах переменных учитывается регистр: `x` и `X` — это разные переменные. Любые другие символы (символ подчеркивания, знаки препинания, знаки математических операций и другие специальные символы) в именах переменных недопустимы. Исключение составляет знак денежной единицы `$` — он (как во многих других версиях BASIC) ставится в конце имен переменных строкового типа.

В BASIC-256 используются только два типа данных: числа и символьные строки (строковые значения). Например, `z` — числовая переменная, а `z$` — строковая переменная (читается "z строковое"). Строковые значения и строковые константы пишутся в кавычках.

Оператор присваивания (так же, как и во всех версиях BASIC) — знак равенства `=`. Примеры записи операторов присваивания:

```
□ x = 5.47
```

```
□ y$ = "Привет, BASIC-256"
```

Оператор присваивания следует читать как "принять равным" или "присвоить значение". В правой части оператора присваивания можно записывать константу или выражение: `x = x + 5` (икс принять равным икс плюс пять).

Для чисел используется внутреннее представление в форме 32-битных двоичных чисел со знаком с плавающей запятой. В операторах и константах допускается запись чисел только в

естественной форме с десятичной точкой в качестве разделителя целой и дробной части: 2.345; -0.0076.

ВНИМАНИЕ!

Запись чисел в экспоненциальной (полулогарифмической) форме в рассматриваемых версиях BASIC-256 не поддерживается.

Максимально представимые числа по абсолютной величине: +/-1410065408. Точность представления результатов при выводе на текстовый экран оператором `print` чисел $N < 1$ составляет 7 десятичных цифр, включая ноль целых с округлением до 6-го знака после запятой. Так `print 0.0000005` выведет на экран **0**, а `print 0.00000051` выведет **0.000001**. Другими словами, число $|N| \leq 0.0000005$ считается машинным нулем.

Из сложных структур данных в BASIC-256 определены только одномерные и двумерные массивы: числовые и строковые. Массивы описываются оператором `DIM`:

- `DIM Z(20)` — числовой массив из 20 элементов $Z[0] — Z[19]$;
- `DIM V$(15)` — строковый массив из 15 элементов V[0] — V$[14]$.

В общем случае оператор `DIM R(N)` при заданном N описывает N числовых переменных $R[0] — R[N - 1]$, т. е. отводит для этих переменных необходимую область оперативной памяти.

Оператор `DIM U(N, M)` при заданных N и M описывает $N \times M$ числовых переменных $U[0,0] — U[N - 1, M - 1]$. Такой массив можно записать в виде прямоугольной таблицы из N строк по M чисел в каждой строке. Элемент $U[i,j]$ при этом будет располагаться в строке i и иметь номер j слева: $i = 0 — N - 1, j = 0 — M - 1$.

Если размерность массива неизвестна, ее можно определить следующим образом: для одномерного массива $X(N)$: $N = X[?]$; для двумерного массива $U(N, M)$: $N = U[?,]$, $M = U[,?]$. Каждый массив должен вводиться отдельным оператором `DIM`.

ВНИМАНИЕ!

В отличие от многих других версий, BASIC-256 не позволяет одним оператором `DIM` описать несколько массивов.

В BASIC-256 допустимы анонимные массивы — последовательности числовых или строковых констант в фигурных скобках, разделенных запятыми. Анонимные массивы можно использовать для присвоения значений обычным массивам:

```
dim Y(4): Y = {10, 20, 30, 40}
```

Приведенная последовательность операторов присвоит элементам массива $Y[0]$ — $Y[3]$ значения, перечисленные в списке анонимного массива: 10, 20, 30, 40. В общем случае, если в списке анонимного массива перечислены N значений, указанная последовательность операторов присвоит их элементам $Y[0]$ — $Y[N - 1]$.

Размерность числового или строкового массива можно переопределить в программе операторами вида `Redim x(n1)`, `Redim y$(n2)`, `Redim x1(n1, m1)`, `Redim y1$(n2, m2)`. При этом, если размерность массива увеличивается, добавленные элементы в числовом массиве инициализируются нулями, а в строковом массиве — пустыми строками. Если новая размерность массива меньше заданной ранее, "лишние" элементы теряются.

ГЛАВА 2

Основные операторы BASIC-256

2.1. Операторы обработки числовых данных

В отличие от многих других реализаций BASIC, оператор `print` в текущей версии BASIC-256 v.0.9.6p выводит на экран не список вывода, а результаты вычисления или обработки только одного выражения и имеет синтаксис `print expr`. Здесь `expr` — сокращение от `expression` — произвольное выражение, формирующее результат в числовой или строковой форме для вывода на текстовый экран (в текстовое окно) исполнителя. Следует заметить, что сам оператор `print` преобразует перед выводом на текстовый экран любой результат, в том числе числовой, в текстовую строку. Этот факт можно использовать для принудительного формирования списка вывода в составе `expr` в виде одной текстовой строки, как это делается в различных модификациях Visual Basic. Типовые приемы формирования списка вывода в составе `expr` в виде одной текстовой строки рассмотрены в *разд. 2.2*.

Оператор `input` обеспечивает диалоговый ввод данных в числовую или строковую переменную в следующих формах:

- `input expr, x1` — ввод числа с клавиатуры в переменную `x1`;
- `input expr, x2$` — ввод текстовой строки с клавиатуры в переменную `x2$`.

Здесь `expr` — необязательное строковое выражение, представляющее собой побуждающее сообщение для ввода. Например, первая строка приведенной ниже последовательности:

```
input "Введите целое x1 = ", x1
print x1
```

выведет на текстовый экран сообщение: **Введите целое x1 =**. Пользователь в той же строке вводит с клавиатуры нужное число и нажимает клавишу <Enter>. Переменной `x1` будет присвоено введенное число. Вторая строка выведет присвоенное `x1` значение в окно текста.

Над числовыми данными возможны следующие операции:

- сложение: $y_1 = x_1 + x_2$;
- вычитание: $y_2 = x_1 - x_2$;
- умножение: $y_3 = x_1 * x_2$;
- деление: $y_4 = x_1 / x_2$;
- возведение в степень: $y_5 = x_1 ^ x_2$;
- вычисление целого остатка y_6 от деления целых чисел x_3 и x_4 :
 $y_6 = x_3 \% x_4$;
- целочисленное деление $y_7 = x_5 \setminus x_6$ — при этом вычисляется целая часть частного y_7 от деления целых чисел x_5 и x_6 .

Например: команда `print 4.5^(-2.5)` выведет на экран результат вычисления: **0.026081**; команда `print 45 % 7` выведет остаток от деления 45 на 7, т. е. **3**; команда `print 45 \ 7` выведет целую часть частного от деления 45 на 7, т. е. **6**.

Для числовых аргументов могут вычисляться следующие функции:

- `abs(x)` — вычисляет абсолютное значение выражения x , например: `abs(-4.5) = 4.5`;
- `x1 = ceil(x)` — вычисляет наименьшее целое $x_1 \geq x$, например: `ceil(4.5) = 5`; `ceil(-4.5) = -4`;
- `x2 = floor(x)` — вычисляет наибольшее целое $x_2 \leq x$, например: `floor(4.5) = 4`; `floor(-4.5) = -5`;

- $x3 = \text{int}(x)$ — вычисляет целое число $x3$ путем отбрасывания дробной части числа x , например: $\text{int}(4.5) = 4$; $\text{int}(-4.5) = -4$;
- pi, PI — встроенная константа $\pi = 3.141593$;
- $\text{cos}(x)$ вычисляет $\cos x$;
- $\text{sin}(x)$ вычисляет $\sin x$;
- $\text{tan}(x)$ вычисляет $\text{tg } x$;

ВНИМАНИЕ!

Для всех тригонометрических функций аргумент x задается в радианах. Аргумент $x1$, заданный в градусах, необходимо преобразовать в радианы с помощью переводного множителя $1^\circ = \text{pi}/180$, например: $y1 = \text{sin}(x1 * \text{pi}/180)$. Можно также воспользоваться для преобразования аргумента $x1$, заданного в градусах, в радианы специальной функцией $\text{radians}(x1)$, например:

$y1 = \text{sin}(\text{radians}(x1))$.

- функция $y = \text{atan}(x)$ вычисляет $\text{arctg } x$, причем y получится в радианах.

Если необходимо получить результат $y2$ в градусах, следует использовать выражение вида $y2 = \text{atan}(x) * 180 / \text{pi}$ или воспользоваться для преобразования угла из радианной меры в градусную функцией degrees : $y2 = \text{degrees}(\text{atan}(x))$.

Аналогично для вычисления математических функций $\text{arcsin } x$, $\text{arccos } x$ можно в BASIC-256 пользоваться функциями $\text{asin}(x)$, $\text{acos}(x)$;

- вычисление корней произвольной степени выполняется через операцию возведения в степень:
 - квадратный корень вычисляется так: $x1 = x^{(1/2)}$;
 - кубический корень: $x2 = x^{(1/3)}$;
 - корень произвольной степени n : $x3 = x^{(1/n)}$;
- функция $y1 = \text{log}(x)$ вычисляет натуральный (Неперов) логарифм $\ln x$ по основанию $e = 2.73$: $\text{log}(10) = 2.302585$;
- функция $y2 = \text{log10}(x)$ вычисляет десятичный логарифм $\lg x$: $\text{log10}(1000) = 3$;

- логарифм числа x по произвольному основанию a вида $y_3 = \log_a x$ можно вычислить, используя стандартное математическое соотношение $\log_a x = \ln x / \ln a$. В BASIC-256 для этого придется записать одно из выражений: $y_3 = \log(x) / \log(a)$ или $y_3 = \log_{10}(x) / \log_{10}(a)$.

ПРИМЕЧАНИЕ

Следует заметить, что имеющаяся во всех реализациях BASIC функция $\exp(x) = e^x$, обратная натуральному логарифму, в BASIC-256 отсутствует. При необходимости ее использования в программе можно ввести константу $e = 2.718282$ и использовать функцию вида e^x вместо $\exp(x)$.

Оператор `rand` формирует псевдослучайное число x в диапазоне $0 \leq x < 1$, распределенное по закону равной плотности. В отличие от других реализаций BASIC инициализация генератора случайных чисел (функции `rand`) в BASIC-256 выполняется не аргументом, а автоматически при каждой очередной загрузке. После каждой новой загрузки BASIC-256 будет формироваться новая последовательность случайных чисел от 0 до 0.999999. Если необходимо сформировать случайную последовательность чисел в другом диапазоне для моделирования некоторой последовательности событий, можно воспользоваться правилом: любое выражение, содержащее функцию формирования случайной последовательности `rand`, дает в результате случайную последовательность. Формирование выражений можно пояснить следующими примерами:

- бросание монеты — орлу ставится в соответствие 1, решке — 0. Случайная последовательность равновероятных 1 и 0 формируется с помощью выражения: $x = \text{int}(2 * \text{rand})$;
- бросание кубика — формирование случайной последовательности целых чисел x_1 в диапазоне от 1 до 6 реализуется выражением:

$$x_1 = 1 + \text{int}(6 * \text{rand}); \quad (2.1)$$

- вытаскивание карты из хорошо перемешанной колоды из 36 карт — если поставить в соответствие картам от валета до туза числа от 11 до 14, то без учета масти карт процесс моде-

лируется формированием последовательности случайных целых чисел x_2 в диапазоне от 6 до 14 согласно выражению:

$$x_2 = 6 + \text{int}(9 * \text{rand}). \quad (2.2)$$

В общем случае если некоторая последовательность событий моделируется последовательностью случайных целых чисел x_3 в диапазоне от N_1 до N_2 , то можно использовать выражение вида:

$$x_3 = N_1 + \text{int}((N_2 - N_1 + 1) * \text{rand}). \quad (2.3)$$

Оператор `pause x` — задержка на x секунд. Здесь x — десятичное число. Так, команда `pause 0.15` задерживает исполнение программы на 0.15 секунды. Этот оператор полезен при формировании подвижных изображений в графическом или текстовом окнах или при отображении результата работы программы автоматически сменяющейся с определенной задержкой последовательностью кадров, т. е. в виде слайд-фильма.

Оператор `Sound` — воспроизводит звук или последовательность звуков заданной частоты F и длительности T через системный динамик. Частота задается в герцах (Гц), длительность в миллисекундах (мс): 1 мс = 0.001 с. Команда `Sound` имеет следующие форматы:

- `Sound F1, T1` или `Sound(F1, T1)` — воспроизводит звук частоты F_1 Гц длительностью T_1 мс, например, команда `Sound(1000, 5000)` воспроизведет звук частоты 1000 Гц длительностью 5 сек. Паузу можно задать частотой 0 и нужной длительностью;
- `Sound Y`, `Sound(Y)`, `Sound {F1, T1, F2, T2, ... , Fn, Tn}` — аргументом команды может быть массив Y из n пар чисел F_i, T_i , в том числе анонимный массив.

ПРИМЕЧАНИЕ

При некотором навыке в этой форме можно записывать мелодии (табл. 2.1). Слышимость звуков зависит от полосы пропускания системного динамика. Звуки очень низких и очень высоких звуковых частот (ниже 50 Гц, выше 7–8 кГц) могут не воспроизводиться системным динамиком и будут не слышны.

В табл. 2.1 приведены латинские и русские обозначения нот и соответствующие им частоты в Гц для двух октав. Ноты проме-

Таблица 2.1. Таблица соответствия частот и нот [7]

	A _{Sharp}	B	C	C _{Sharp}	D	D _{Sharp}	E	F	F _{Sharp}	G	G _{Sharp}
	B _{Flat}			D _{Flat}		E _{Flat}			G _{Flat}		A _{Flat}
ЛЯ		СИ	ДО		РЕ		МИ	ФА		СОЛЬ	
440	466	493	523	554	587	622	659	698	740	784	
220	233	247	262	277	294	311	330	349	370	392	415
								175	185	196	208

жуточных частот обозначены следующим образом: C_{Sharp} это ДО-диез; D_{Flat} это РЕ-бемоль. Длительность звучания нот определяется темпом исполнения — числом ударов метронома в минуту. Каждый удар — четверть ноты. Темп 120 — это 120 ударов метронома в минуту. Целая нота — 2 сек. Половинная нота — 1 сек. Четверть ноты — 0,5 сек.

Оператор Volume X или Volume(X) — задает громкость звука, воспроизводимого оператором Sound. Уровень громкости задается целым числом x от 0 до 10. По умолчанию x = 5.

Оператор System z\$ или System(z\$) — позволяет из программы в среде BASIC-256 выполнить системную команду (команду операционной системы), заданную текстовой строкой z\$, в окне терминала. Это аналог команды Windows Пуск | Выполнить. Применение такой команды требует большой осторожности и высокой квалификации пользователя.

2.2. Операторы обработки строковых данных

Оператор Asc(x\$) — код первого символа строки x\$ или символа x\$: Asc(A) = 65; asc(blue) = 98 — код символа b.

Оператор Chr(N) — формирует символ по коду N: chr(90) = Z; chr(123) = {.

Символы стандарта ASCII с кодами от 32 до 126 приведены в табл. 2.2.

Таблица 2.2. Кодирование символов стандарта ASCII (коды от 32 до 126)

32	Пробел	44	,	56	8	68	D	80	P	92	\	104	h	116	t
33	!	45	-	57	9	69	E	81	Q	93]	105	i	117	u
34	"	46	.	58	:	70	F	82	R	94	^	106	j	118	v
35	#	47	/	59	;	71	G	83	S	95	_	107	k	119	w
36	\$	48	0	60	<	72	H	84	T	96	`	108	l	120	x
37	%	49	1	61	=	73	I	85	U	97	a	109	m	121	y
38	&	50	2	62	>	74	J	86	V	98	b	110	n	122	z
39	'	51	3	63	?	75	K	87	W	99	c	111	o	123	{
40)	52	4	64	@	76	L	88	X	100	d	112	p	124	
41	(53	5	65	A	77	M	89	Y	101	e	113	q	125	}
42	*	54	6	66	B	78	N	90	Z	102	f	114	r	126	~
43	+	55	7	67	C	79	O	91	[103	g	115	s	127	

Следует обратить внимание на два похожих при печати символа:

- `chr(39) = ' — апостроф;`
- `chr(96) = ` — символ слабого ударения.`

Символ ПУСТО (пустая строка) отображается командой `chr(0)`.

Коды от 1 до 31 таблицы символов ASCII отведены для неотображаемых управляющих символов. Например, команды `chr(10)`, `chr(13)` при выводе в текстовое окно оператором `print` обеспечивают переход в начало следующей строки.

Знак `+` для строковых переменных и значений выполняется как операция конкатенации (объединения) текстовых строк. Строка программы вида:

$$x\$ = x1\$ + x2\$ + x3\$ \quad (2.4)$$

сформирует объединенную текстовую строку `x$`, состоящую из трех составляющих текстовых строк `x1$`, `x2$`, `x3$`, записанных последовательно в порядке их следования в выражении. В выражении вида (2.4) могут использоваться и числовые переменные. Они автоматически будут преобразованы в строковое представление. Выражение вида:

$$y\$ = "x1 = " + x1 + " " + "x2 = " + x2, \quad (2.5)$$

где обрабатываются одновременно числовые и строковые значения, сформирует результирующую текстовую строку `y$`, в которую числовые значения `x1` и `x2` войдут также в форме текстовых строк. Выражение (2.5) позволяет в операторе `print` выводить в одной строке значения нескольких переменных, если объединить их в общую текстовую строку, добавив в нее необходимые разделительные пробелы и знаки препинания. Для примера — последовательность строк программы (листинг 2.1)

Листинг 2.1

```
x1 = 25
x2 = -48
print "x1 = " + x1 + ", " + "x2 = " + x2
```

выведет в текстовое окно следующий результат в виде текстовой строки: **x1 = 25, x2 = -48.**

Функция $y\$ = \text{string}(x)$ — преобразует число x в строковое представление $y\$$ (см. функцию `float`).

Функция $x = \text{Float}(y\$)$ — преобразует строковое представление числа $y\$$ в число с плавающей запятой. Функция `float` — обратная для функции `string` и наоборот. `Float("137.89") = 137.89`.

Функция $N = \text{Instr}(S1\$, S2\$)$ — выявляет вхождение строки $S2\$$ в строку $S1\$$. Если $S2\$$ входит в $S1\$$, функция возвращает номер позиции N , с которой начинается строка $S2\$$ внутри строки $S1\$$ (позиции в строке нумеруются слева направо, начиная с 1). Если строка $S2\$$ не содержится в $S1\$$, функция возвращает $N = 0$. Так, в примере:

```
N = instr("Что такое крокодил", "крокодил")
```

в результате получится $N = 11$.

Функция $N1 = \text{Length}(S\$)$ — возвращает длину (число символов) $N1$ в строке $S\$$.

Функция $S2\$ = \text{Mid}(S1\$, n1, n)$ — выделяет подстроку $S2\$$ из строки $S1\$$, начиная с позиции $n1$ длиной n символов. В примере $S2\$ = \text{mid}(\text{"Информатика"}, 3, 5)$ получится $S2\$ = \text{"форма"}$.

Функция $S3\$ = \text{left}(S1\$, n2)$ — выделяет из строки $S1\$$ подстроку $S3\$$, содержащую $n2$ символов с левого края, т. е. от начала строки: `left("собака", 3) = "соб"`.

Функция $S4\$ = \text{right}(S1\$, n3)$ — выделяет из строки $S1\$$ подстроку $S4\$$, содержащую $n3$ символов с правого края, т. е. от конца строки: `right("собака", 3) = "ака"`.

Функция `Lower(S$)` — преобразует латинские буквы строки $S\$$ в символы нижнего регистра, т. е. в строчные буквы: `Lower("Bear") = "bear"`. Русские буквы и специальные символы (знаки препинания, знаки математических операций) остаются без изменения.

Функция `Upper(S$)` — преобразует латинские буквы строки $S\$$ в символы верхнего регистра, т. е. в прописные буквы: `Upper("Leningrad") = "LENINGRAD"`. Русские буквы и специальные символы остаются без изменения.

Оператор `Say S$` — произносит текстовую строку $S\$$ на английском языке. Например, команда `Say "Do you speak English?"`

произнесет приведенную фразу. При этом используется системный преобразователь текста в речь (TTS, Text To Speech Engine). В Windows по умолчанию для этого служит звуковой интерфейс для прикладных программ SAPI (Sound Application Programming Interface). В Linux для той же цели должна быть установлена одна из библиотек FLite или eSpeak. Русские строки не произносятся, русский текст будет воспроизводиться только в виде строк, записанных латинскими буквами.

Строки комментариев включаются в программу операторами `Rem` или `#`. Интерпретатор BASIC-256 при просмотре программы строки комментариев пропускает без обработки.

Оператор `Font fn, p, w` — устанавливает характеристики шрифта для команд отображения текста в окнах текста (`print`) или графики (`text`). Здесь: `fn` (font name) — имя шрифта в операционной системе, например, Times New Roman, Arial; `p` (point) — размер шрифта в пунктах (1 пункт = 1/72 дюйма, 1|72"); `w` (weight) — интенсивность цвета или "жирность" символов от 1 до 100: `Light = 25` — слабый (бледный), `Normal = 50` — нормальный, `Bold = 75` — полужирный.

2.3. Управляющие операторы BASIC-256

В BASIC-256, как и в большинстве других сред программирования, строки программы обрабатываются и исполняются в порядке их записи. Управляющие операторы могут изменять порядок обработки и исполнения строк программы. Управляющие операторы условно можно разделить на следующие группы: операторы формирования условий, операторы анализа условий, операторы циклов, операторы для работы с подпрограммами.

2.3.1. Операторы формирования условий

Специального представления для логического типа данных вида условий, которые могут иметь два значения "выполнено" и "не выполнено", в BASIC-256, как и в большинстве других версий