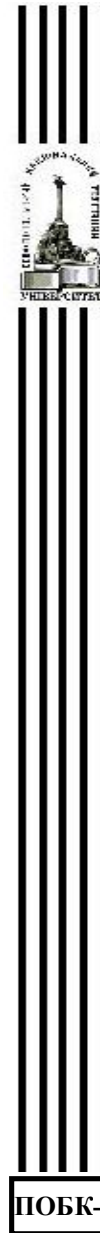


БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Микропроцессоры: в 3-х книгах / Под ред. Преснухина.-М.: Высш. шк., 1989.-285 с.
2. Токхайм Р. Микропроцессоры. Курс упражнений /Р. Токхайм.-М.: Энергоатомиздат, 1988.-336 с.
3. Абель П. Язык ассемблера для IBM PC и программирования: Пер. с англ. Ю.В. Сальникова / П. Абель.- М.: Высш. шк., 1992.- 447 с.
4. ГОСТ 19.701-90 Схемы алгоритмов, программ, данных и схем. Обозначения условные и правила выполнения.— Взамен ГОСТ 19.002-80, ГОСТ 19.003-80; Введ. 01.01.92.— М.: Изд-во стандартов, год.— 25с.

Министерство образования и науки Украины
Севастопольский национальный технический университет

**МЕТОДИЧЕСКИЕ УКАЗАНИЯ**

к практическому занятию №3

дисциплины "Программное обеспечение

бортовых компьютеров" по теме «Циклы, методика

программирования» для студентов специальности

07.09.0258 «Автомобили и автомобильное хозяйство»

дневной формы обучения

Севастополь

2006

ПОБК-3



УДК 628.113.004.58

Методические указания к практическому занятию №3 дисциплины "Программное обеспечение бортовых компьютеров" по теме «Циклы, методика программирования» для студентов специальности 07.09.0258 «Автомобили и автомобильное хозяйство» дневной формы обучения /Разраб. В.П. Долгин.– Севастополь: Изд-во СевНТУ, 2006.- 12с.

Методические указания составлены в соответствии с требованиями программы дисциплины «Программное обеспечение бортовых компьютеров» для студентов специальности 7.09.0258. Целью настоящих методических указаний является обеспечение возможности выполнения практического занятия и самостоятельной работы по теме «Циклы, методика программирования» для студентов дневной формы обучения.

Методические указания утверждены на заседании кафедры АТ
« 28 » апреля 2006 г. протокол № 4 .

Допущено учебно -методическим центром университета
в качестве методических указаний

Рецензент: Каинов Д.А., канд. техн. наук, доцент.

1.6. Порядок выполнения индивидуального задания

При выполнении индивидуального задания рекомендуется придерживаться следующего порядка.

- 1) Определить все необходимые исходные данные.
- 2) Пользуясь методикой, изложенной в разделе 2, написать программу с циклами.
- 3) Ввести программу, выполнить ее в пошаговом режиме, и проверить правильность ее работы на контрольном примере.
- 4) Записать комментарии к командам программы.
- 5) Оформить и сдать работу.

1.7. Контрольные вопросы

- 1) Виды циклов. Примеры.
- 2) Приведение условий к нормальному виду. Примеры.
- 3) Методика программирования циклов.
- 4) Команды отладчика. Примеры.



1.5. Индивидуальное задание

Каждый студент выполняет индивидуальное задание, выбрав его вариант в соответствии с номером в списке академической группы по таблице 1.

Таблица 3 - Индивидуальное задание

№ варианта	Функция	№ варианта	Функция
1	$F(x)=$	16	$F(x)=2x^2-9x+1$
2	$F(x)=3x^2-4x+7$	17	$F(x)=3x^2-8x+2$
3	$F(x)=4x^2+3x-2$	18	$F(x)=4x^2-7x+3$
4	$F(x)=2x^2-7x+4$	19	$F(x)=5x^2-6x+4$
5	$F(x)=x^2-3x+5$	20	$F(x)=6x^2-5x+5$
6	$F(x)=2x^2-4x+2$	21	$F(x)=7x^2-4x+6$
7	$F(x)=3x^2-6x+2$	22	$F(x)=8x^2-3x+7$
8	$F(x)=5x^2-2x+3$	23	$F(x)=9x^2-2x+8$
9	$F(x)=4x^2-3x+8$	24	$F(x)=2x^2-3x+9$
10	$F(x)=2x^2-8x+9$	25	$F(x)=3x^2-4x+1$
11	$F(x)=2x^2-5x+2$	26	$F(x)=4x^2-5x+2$
12	$F(x)=2x^2-7x+3$	27	$F(x)=5x^2-6x+3$
13	$F(x)=2x^2-9x+4$	28	$F(x)=6x^2-7x+4$
14	$F(x)=2x^2-4x+5$	29	$F(x)=7x^2-8x+5$
15	$F(x)=2x^2-3x+6$	30	$F(x)=8x^2-2x+6$

СОДЕРЖАНИЕ

1. Практическое занятие № 3.....	4
1.1. Общие сведения	4
1.2. Структуры циклов	5
1.3. Методика программирования	7
1.4. Пример реализации	9
1.5. Индивидуальное задание	10
1.6. Порядок выполнения индивидуального задания.....	11
1.7. Контрольные вопросы	11
Библиографический список.....	12



1. ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 3

Тема занятия: Циклы, методика программирования

Цель занятия – получить практические навыки при разработке программ с циклическими алгоритмами.

В результате практического занятия студент

должен знать:

- команды обработки данных и условных переходов, необходимые для организации циклов;
- порядок составления программ, содержащих циклы;

должен уметь:

- разрабатывать программы, содержащие циклы;
- запускать разработанные программы и выводить результаты на дисплей.

1.1. Общие сведения

При решении практических задач используется большое число алгоритмов, которые содержат повторение некоторых действий до тех пор, пока не выполнится некоторое условие. Такие алгоритмы называются циклическими, а соответствующие условия – условиями выхода из цикла.

Программирование цикла заключается в передаче управления (условном или безусловном) на некоторую предшествующую команду программы с целью повторного выполнения участка программы. При программировании циклов используются команды обработки данных, а также команды управления порядком обработки данных в форме условного или безусловного перехода.

3) Программы правильно работают для $N > 0$.

В результате анализа работы программ можно сделать следующие выводы:

- 1) Реализовать в программе ДО – цикл проще, чем ПОКА – цикл.
- 2) ПОКА – цикл более надежная конструкция и его применение можно считать предпочтительным.
- 3) ДО – цикл менее надежен и его следует употреблять только тогда, когда имеется уверенность, что тело цикла должно выполняться не менее, чем 1 раз.

1.4. Пример реализации

Разработать алгоритм и программу на ассемблере реализации вычисления функции $Y = X^N$ (X и N произвольные).

Условия: значения констант и переменных ввести в регистр данных.

СТАТУС - РГОВЛ		ОТЛАДКА		ИНДИКАТОРЫ	
СУММАТОР 0.0000		ZR = 0	PL = 1	NG = 0	OF = 0 DV = 0 UF = 0
1	Ст. пр. АВ-51д_Фамилия_И.О.	1	X	=	2.4000
2	Возведение в степень: N>=0	2	N	=	0.0000
3	INC ;ДОПОЛНЕНИЕ	3	Y	=	32.0000
4	ST Y;ДОПОЛНЕНИЕ	4		=	0.0000
5	M LD N;S<N	5		=	0.0000
6	ПРОВЕРКА УСЛОВИЯ N=0	6		=	0.0000
7	GO ZR END	7		=	0.0000
8	ТЕЛО ЦИКЛА	8		=	0.0000
9	LD Y;S<Y	9		=	0.0000
10	MUL X;S<S*X	10		=	0.0000
11	ST Y;Y<S	11		=	0.0000
12	LD N;S<N	12		=	0.0000
13	DEC ;S<N-1	13		=	0.0000
14	ST N;N<S	14		=	0.0000
15	БЕЗУСЛОВНЫЙ ПЕРЕХОД	15		=	0.0000
16	GO M	16		=	0.0000
17	END STOP	17		=	0.0000
18		18		=	0.0000

Рисунок 4 – Пример реализации



Шаг 3 - Кодирование программы.

Текст программы, реализующей цикл, выглядит следующим образом.

Таблица 2 – Шаг 3, пример реализации кодирования

ДО – ЦИКЛ:			ПОКА – ЦИКЛ:		
1	LD	ONE	LD	ONE	
2	ST	ФАКТ	ST	ФАКТ	
3	INP		INP		
4	ST	N	LOOP	ST	N
5	LOOP	LD	ФАКТ	GOZR	VIHOD
6	MUL	N	GONG	VIHOD	
7	ST	ФАКТ	LD	ФАКТ	
8	LD	N	MUL	N	
9	DEC		ST	ФАКТ	
10	GOPL	LOOP	LD	N	
11	LD	ФАКТ	DEC		
12	OUT		GOPL	LOOP	
13	STOP		VIHOD	LD	ФАКТ
14			OUT		
15			STOP		

Предполагается, что переменные N=0, ФАКТ=0, ONE=1 описаны в памяти данных и в ячейках N, ФАКТ, ONE хранятся соответствующие константы.

Проверка работоспособности вариантов реализации программ показывает, что:

- 1) Программы по-разному работают на отрицательных значениях N.
- 2) Программа с ДО – циклом дает неверный результат для N=0.

1.2. Структуры циклов

Простейший цикл имеет следующую структуру:

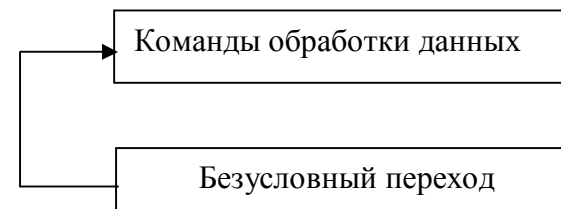


Рисунок 1 – Структура бесконечного цикла

Каждый цикл состоит из заголовка цикла (начало цикла) тела цикла (команд, выполняемых в цикле), конца цикла (условие выхода из цикла).

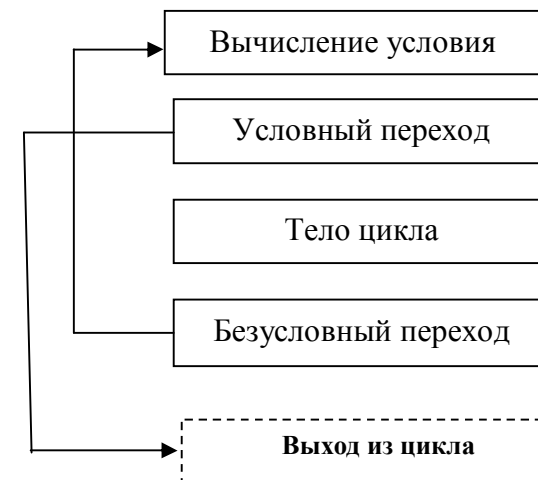


Рисунок 2 – Структура WHILE-цикла



Виды циклов:

- 1) Цикл с предусловием (проверкой условия до выполнения тела цикла, ПОКА (WHILE) - цикл):

ПОКА <УСЛОВИЕ> ЦИКЛ <ТЕЛО> КОНЕЦ .

ПОКА – цикл работает следующим образом: пока условие истинно, выполняется тело цикла, в противном случае, выполняется первая команда, следующая за телом цикла.

- 2) Цикл с постусловием (проверкой условия после выполнения тела цикла ДО (DO) - цикл):

ЦИКЛ <ТЕЛО> ДО <УСЛОВИЕ> .

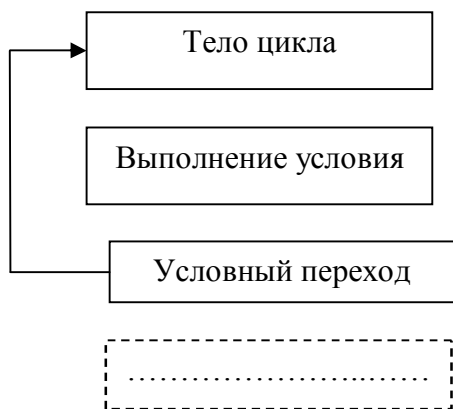


Рисунок 3 – Структура ДО–цикла

DO – цикл работает следующим образом: вычисляется тело цикла, затем проверяется условие, если условие ложно, то управление

передается на начало цикла, иначе управление получает первая команда, следующая за телом цикла.

1.3. Методика программирования

Пример. Пусть требуется написать программу вычисления значения $N!$ для любого неотрицательного целого N .

Запишем условия нахождения $N!$ в зависимости от значения N

$$N! = \begin{cases} \text{неопределено,} & \text{если } N < 0; \\ 1, & \text{если } N = 0 \text{ или } N = 1; \\ 1 \cdot 2 \cdot \dots \cdot N, & \text{если } N > 1. \end{cases}$$

Выполним программирование по шагам.

Шаг 1 - распределение памяти для констант и переменных
 $N=0$, $ФАКТ=0$, $ОПЕ=1$.

Шаг 2 - проектирование программы.

Алгоритм программы выглядит следующим образом.

Таблица 1 – Шаг 2, варианты реализации

	ДО – ЦИКЛ:	ПОКА – ЦИКЛ:
1	ФАКТ ← 1	ФАКТ ← 1
2	N ← Ввод	N ← Ввод
3 3.1	ФАКТ ← ФАКТ x N	ФАКТ ← ФАКТ x N
4 3.2	N ← N – 1	N ← N – 1
	ДО N=0	КОНЕЦ
5	Сумматор ← ФАКТ	Сумматор ← ФАКТ
6	ВЫВОД	ВЫВОД

