МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МО ЭВМ

ОТЧЕТ

по лабораторной работе №1
по дисциплине «Качество и метрология программного обеспечения»
ТЕМА: Расчет метрических характеристик качества разработки программ по метрикам Холстеда

Студент гр. 8303 _____ Ивченко А.А.

Преподаватель _____ Кирьянчиков В.А.

Санкт-Петербург
2022

Цель работы

Для заданного варианта программы обработки данных, представленной на языке Паскаль, разработать вычислительный алгоритм и также варианты программ его реализации на языках программирования Си и Ассемблер. Добиться, чтобы программы на Паскале и Си были работоспособны и давали корректные результаты (это потребуется в дальнейшем при проведении с ними измерительных экспериментов). Для получения ассемблерного представления программы можно либо самостоятельно написать код на ассемблере, реализующий заданный алгоритм, либо установить опцию "Code generation/Generate assembler source" при компиляции текста программы, представленной на языке Си. Во втором случае в ассемблерном представлении программы нужно удалить директивы описаний и отладочные директивы, оставив только исполняемые операторы.

Для каждой из разработанных программ (включая исходную программу на Паскале) определить следующие метрические характеристики (по Холстеду):

1. Измеримые характеристики программ:
   - число простых(отдельных)операторов, в данной реализации;
   - число простых (отдельных) операндов, в данной реализации;
   - общее число всех операторов в данной реализации;
   - общее число всех операндов в данной реализации;
   - число вхождений j-го оператора в тексте программы;
   - число вхождений j-го операнда в тексте программы;
   - словарь программы;
   - длину программы.
2. Расчетные характеристики программы:
   - длину программы;
   - реальный и потенциальный объемы программы;
   - уровень программы;

- интеллектуальное содержание программы;

- работу программиста;

- время программирования;

- уровень используемого языка программирования;

- ожидаемое число ошибок в программе.

Для характеристик длина программы, уровень программы, время программирования следует рассчитать как саму характеристику, так и ее оценку.

Ход выполнения.

Был выбран 5 вариант задания – процедура сортировки методом

«пузырька». Из исходного варианта программы был удалён вызов процедуры печати массива – с целью оптимизации для последующих работ. Был произведён вручную расчёт операторов и операндов.

Результат подсчета операторов и операндов на языках Паскаль, C и Assembler представлены в таблицах 1 - 3. Код программ в приложениях А, Б и В.

№	Оператор	Количество	№	Операнд	Количество
1	;	15	*1*	max	3
2	:=	10	*2*	80	1
3	>	1	*3*	1	5
4	()	2	*4*	hold	2
5	begin end	4	*5*	p	2
6	[]	7	*6*	q	2
7	if … then	1	*7*	no_change	3
8	for … to … do	2	*8*	true	1
9	+	2	*9*	j	3
10	-	1	*10*	n	5
11	randomize	1	*11*	a	2
12	random(100)	1	*12*	false	1
13	sort(x, n)	1	*13*	i *(в Main)*	2
14	swap(a[j], a[j+1])	1	*14*	x	2
15	repeat … until	1			

Таблица 1 – Ручной расчёт операторов и операндов в программе на Pascal.

№	Оператор	Количество	№	Операнд	Количество
1	!	1	1	0	4
2	%	2	2	1	4
3	()	3	3	100	1
4	+	2	4	80	1
5	++	2	5	A	5
6	,	4	6	Hold	2
7	-	1	7	I	4
8	;	13	8	J	7
9	<	2	9	Max	2
10	=	9	10	N	3
11	>	1	11	No_change	3
12	[]	6	12	P	3
13	* (*указатель*)	7	13	Q	3
14	Do … while	1	14	x	3
15	For	2			
16	Main	1			
17	Rand	2			
18	Return	1			
19	Sort	1			

21	Swap				
22	void	2			

Таблица 2 – Ручной расчёт операторов и операндов в программе на C.

№	Оператор	Количество	№	Операнд	Количество
1	push	8	1	rbp	8
2	mov	61	2	rsp	10
3	sub	8	3	rdi	4
4	nop	3	4	rsi	3
5	pop	7	5	rax	30
6	ret	3	6	eax	25
7	cdqe	5	7	edx	15
8	lea	6	8	esi	3
9	add	10	9	rdx	18
10	cmp	3	10	rcx	6
11	jmp .L3	1	11	al	2
12	jle .L4	1	12	rbx	4
13	call swap	1	13	ecx	9
14	movzx	1	14	QWORD PTR [rbp-24]	8
15	xor	1	15	QWORD PTR [rbp-32]	3
16	test	1	16	DWORD PTR [rax]	6
17	jl .L5	1	17	DWORD PTR [rbp-4]	2
18	leave	1	18	DWORD PTR [rbp-28]	2
19	jne .L6	1	19	BYTE PTR [rbp-1]	3
20	call rand	2	20	DWORD PTR [rbp-8]	7
21	call sort	1	21	DWORD PTR [rbp-56]	4
22	cdq	1	22	QWORD PTR [rbp-64]	1
23	idiv	1	23	QWORD PTR [rbp-72]	3
24	movsx	5	24	DWORD PTR [rbp-52]	4
25	div	1	25	DWORD PTR [rax+*4]	1
26	imul	3	26	[0+rax*4]	5

27	shr	2	27	[rbp-40]	1
28	sal	1	28	r15d	1
29	sar	2	29	r13d	1
30	jmp .L8	1	30	r15	2
31	jl .L9	1	31	r14	3
			32	r13	2
			33	r12	3
			34	80	1
			35	32	2
			36	40	1
			37	16	3
			38	31	1
			39	100	1
			40	1	9
			41	0	7
			42	3	1
			43	2	2
			44	5	1
			45	1374389535	1

Таблица 3 – Ручной расчёт операторов и операндов в программе на ассемблере.

В таблице 4 представлены сводные результаты расчетных характеристик вручную.

Для расчётов значение коэффициента Стауда S принято 10; значение $\eta 2 *$ принято 3, считаем что используется 2 массива: исходный и

отсортированный, а также их размер n , поэтому $\eta_2^* = 3$.

	Паскаль	**Си**	**Ассемблер**
Число уникальных операторов (n1):	15	22	31
Число уникальныхоперандов (n2):	14	14	45
Общее число операторов (N1):	50	63	144
Общее число операндов (N2):	34	45	229
Алфавит (n):	29	36	76
Экспериментальная длина программы (Nэ):	84	108	373
Теоретическая длина программы (Nт):	111.906	151.410	400.713
Объём программы (V):	408.070	558.352	2330.477
Потенциальный объём (V*):	11.610	11.610	11.610
Уровень программы (L):	0.028	0.021	0.0050
Оценка уровня программы(L)	0.055	0.028	0.013
Интеллект программы (I):	22.404	15.792	29.545
Работа по программированию (E):	14343.377	26853.273	467811.461
Время программирования (T):	1434.337	2685.327	46781.146
Оценка времени программирования (T)	743.271	1974.173	18382.284
Уровень языка программирования (Lambda):	0.330	0.241	0.057
Число ошибок (B):	1	1	3

Расчет метрик с помощью программы автоматизации

Для программы на Паскале:

Table:

Operators:

1	6	()
2	2	+
3	1	-
4	25	;
5	9	=
6	1	>
7	5	[]
8	2	ary
9	1	boolean
10	1	const
11	2	for
12	1	if
13	3	integer
14	2	procedure
15	1	program
16	1	random
17	1	randomize
18	3	real
19	1	repeat
20	2	sort
21	2	swap
22	1	type

Operands:

1	6	1
2	1	100
3	1	80
4	1	B_sort1
5	5	a
6	1	ary
7	1	false
8	3	hold
9	2	i
10	5	j
11	3	max
12	6	n
13	4	no_change
14	3	p
15	3	q
16	1	true
17	3	x

Statistics for module BubblePas.lxm

```
The number of different operators     : 22
The number of different operands      : 17
The total number of operators         : 73
The total number of operands          : 49

Dictionary                    (  D)   : 39
Length                        (  N)   : 122
Length estimation             ( ^N)   : 167.594
Volume                        (  V)   : 644.819
Potential volume              ( *V)   : 19.6515
Limit volume                  (**V)   : 38.2071
Programming level             (  L)   : 0.030476
Programming level estimation  ( ^L)   : 0.0315399
Intellect                     (  I)   : 20.3375
Time of programming           (  T)   : 1175.46
Time estimation               ( ^T)   : 1560.29
Programming language level    (lambda) : 0.598898
Work on programming           (  E)   : 21158.3
Error                         (  B)   : 0.254995
Error estimation              ( ^B)   : 0.21494
```

Для программы на Си:

```
Statistics for module BubbleC.lxm
=====================================
The number of different operators      : 27
The number of different operands       : 14
The total number of operators          : 95
The total number of operands           : 49

Dictionary                    (  D)    : 41
Length                        (  N)    : 144
Length estimation             ( ^N)    : 181.685
Volume                        (  V)    : 771.487
Potential volume              ( *V)    : 19.6515
Limit volume                  (**V)    : 38.2071
Programming level             (  L)    : 0.0254722
Programming level estimation  ( ^L)    : 0.021164
Intellect                     (  I)    : 16.3278
Time of programming           (  T)    : 1682.63
Time estimation               ( ^T)    : 2555.14
Programming language level    (lambda) : 0.500567
Work on programming           (  E)    : 30287.4
Error                         (  B)    : 0.323882
Error estimation              ( ^B)    : 0.257162
```

```
Table:
===================================
Operators:
|  1   |  1   | !
|  2   |  2   | %
|  3   |  6   | ()
|  4   |  2   | +
|  5   |  2   | ++
|  6   |  4   | ,
|  7   |  1   | -
|  8   |  22  | ;
|  9   |  2   | <
|  10  |  10  | =
|  11  |  1   | >
|  12  |  5   | []
|  13  |  2   | _&
|  14  |  4   | _*
|  15  |  1   | _[]
|  16  |  3   | __*
|  17  |  1   | const
|  18  |  1   | dowhile
|  19  |  2   | for
|  20  |  1   | if
|  21  |  12  | int
|  22  |  1   | main
|  23  |  2   | rand
|  24  |  1   | return
|  25  |  2   | sort
|  26  |  2   | swap
|  27  |  2   | void
```

```
Operands:
|  1   |  4   | 0
|  2   |  4   | 1
|  3   |  1   | 100
|  4   |  1   | 80
|  5   |  5   | a
|  6   |  2   | hold
|  7   |  4   | i
|  8   |  7   | j
|  9   |  2   | max
|  10  |  6   | n
|  11  |  4   | no_change
|  12  |  3   | p
|  13  |  3   | q
|  14  |  3   | x
```

Выводы.

В ходе выполнения лабораторной работы были изучены метрические характеристики качества разработки программ на основе метрик Холстеда. В результате были вручную рассчитаны метрики Холстеда для программ на Pascal, C и ассемблере, а также аналогичные расчёты были произведены с помощью специальных программ автоматизации расчёта для языков Pascal и C. На основе полученных характеристик было установлено, что программа на ассемблере обладает гораздо большим объёмом и следовательно требует гораздо больше времени для написания, что также увеличивает число потенциальных ошибок в ней.