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. 6304	 Рыбин А.С.
Преподаватель	 Кирьянчиков В.А.

Санкт-Петербург 2020

Задание

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 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 ј-го оператора в тексте программы;
 - число вхождений ј-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 работы

1. Определение метрических характеристик для программы на Pascal.

Код программы представлен в приложении A. Измеримые характеристики рассчитанные вручную представлены в таблице 1.

Таблица 1 – Измеримые характеристики программы на Pascal (ручной ... \

подсчёт)

					подсчет	
N₂	Оператор	Количество	N₂	Операнд	Количество	
1	program	1	1	X	2	
2	procedure	1	2	y	1	
3	linfit	1	3	y_calc	1	
4	0	2	4	a	2	
5	•	27	5	b	1	
6	:=	20	6	n	6	
7	real	4	7	linear_fit	1	
8	integer	2	8	i	6	
9	for do	2	9	sum_x	7	
10	+	6	10	sum_y	7	
11	*	9	11	sum_xy	5	
12	/	6	12	sum_x2	5	
13	begin	4	13	sum_y2	4	
14	end	4	14	xi	5	
15	[]	4	15	yi	5	
16	•	1	16	SXX	3	
	Всего	94	17	syy	1	
			18	sxy	2	
			19	0.0	5	
	Bcero 70					

Измеримые характеристики рассчитанные с помощью программы представлены в таблице 2. Файл с результатами программных расчётов представлен в приложении Б.

Таблица 2 – Измеримые характеристики программы на Pascal (программный расчёт)

	_	1	1		расчет
No	Оператор	Количество	No	Операнд	Количество
1	0	6	1	0.0	5
2	*	9	2	1	2
3	+	6	3	a	3
4	-	4	4	b	3
5	/	6	5	i	5
6	;	40	6	linear_fit	1
7	=	18	7	n	7
8	[]	4	8	sum_x	8
9	for	2	9	sum_x2	6
10	integer	2	10	sum_xy	6
11	linfit	1	11	sum_y	8
12	procedure	1	12	sum_y2	5
13	program	1	13	SXX	4
14	real	4	14	sxy	3
	Bcero	105	15	syy	2
			16	X	3
			17	xi	6
			18	у	2
			19	y_calc	2
			20	yi	6
				Всего	88

Определение расчетных характеристик представлено в таблице 3.

Таблица 3 – Расчётные характеристики программы на Pascal

Характеристика	Ручной подсчёт	Программный расчёт
Число простых операторов n ₁	16	14
Число простых операндов n ₂	19	20
Общее число всех операторов N ₁	94	105
Общее число всех операндов N_2	70	88
Словарь п	35	34
Длина N _{опыт}	164	193
Теоретическая длина N _{теор}	133.86	150.84
Объём V	841.16	997.80
Потенциальный объём V*	19.65	19.65
Уровень программы L	0.020	0.020
Оценка уровня программы L~	0.034	0.032
Интеллектуальное содержание I	28.54	31.75
Работа программирования Е	35993.94	50662.60
Оценка времени программирования Т^	3600.76	1361.63
Время программирования Т	2479.20	2814.59
Уровень языка λ	0.39	0.39
Ожидаемое число ошибок в программе В	1	1

2. Определение метрических характеристик для программы на **Си.**

Код программы представлен в приложении В. Измеримые характеристики рассчитанные вручную представлены в таблице 4.

Таблица 4 – Измеримые характеристики программы на Си (ручной подсчёт)

1аол	ица 4 – измерим	ые характери		и программы на Си	<u>(ручнои подсчет)</u>
No	Оператор	Количество	No	Операнд	Количество
1	void	1	1	0	3
2	linfit	1	2	0.0	5
3	0	5	3	a	2
4	{}	3	4	b	2
5	float	6	5	i	10
6	int	5	6	n	6
7	;	23	7	sum_x	7
8	=	20	8	sum_x2	5
9	for	2	9	sum_xy	5
10	++	2	10	sum_y	7
11	<	2	11	sum_y2	4
12	[]	4	12	SXX	3
13	+	10	13	sxy	2
14	-	4	14	syy	1
15	/	6	15	X	2
16	*	9	16	xi	5
	* -				
17	разыменование	4	17	y	1
	указателя			_	
18	* указатель	5	18	y_calc	1
	Всего	114	19	yi	5
				Bcero	76

Измеримые характеристики рассчитанные с помощью программы представлены в таблице 5. Файл с результатами программных расчётов представлен в приложении Γ .

Таблица 5 – Измеримые характеристики программы на Си (программный расчёт)

No	Оператор	Количество	No	Операнд	Количество
1	()	6	1	0	3
2	*	9	2	0.0	5
3	+	6	3	a	3
4	++	2	4	b	3
5	,	15	5	i	10
6	-	4	6	n	7
7	/	6	7	sum_x	8
8	;	28	8	sum_x2	6
9	<	2	9	sum_xy	6
10	=	20	10	sum_y	8
11	[]	4	11	sum_y2	5
12	* _	4	12	SXX	4
13	_[]	1	13	sxy	3
14	_*	6	14	syy	2
15	char	1	15	X	3
16	const	2	16	xi	6
17	float	6	17	у	2
18	for	2	18	y_calc	2
19	int	5	19	yi	6
20	linfit	1		Всего	92
21	void	1			
	Всего	131	1		

Определение расчетных характеристик представлено в таблице 6.

Таблица 6 – Расчётные характеристики программы на Си

	Ручной подсчёт	ристики программы на Си Программный расчёт	
Характеристика	Ручнои подсчет	программный расчет	
Число простых операторов	18	21	
n_1	10	21	
Число простых операндов	10		
n.	19	19	
n ₂ Общее число всех			
Оощее число всех	114	131	
операторов N_1		_	
Общее число всех	T 2	0.5	
операндов \mathbf{N}_2	76	92	
Словарь п	37	40	
Длина Попыт	190	223	
Теоретическая длина N _{теор}	155.77	196.28	
Объём V	989.71	1239.29	
Потенциальный объём V*	19.65	19.65	
Уровень программы L	0.020	0.016	
Оценка уровня программы			
	0.028	0.019	
L°			
Интеллектуальное	27.49	24.08	
содержание I	_,	55	
Работа программирования	40.405.50	70454	
E	49485.50	78154	
Оценка времени			
	4984.86	3064.49	
программирования Т^	2-22-22		
Время программирования Т	3562.96	4341.89	
Уровень языка λ	0.39	0.31	
Ожидаемое число ошибок в	1	1	
программе В	1	1	
F F			

3. Определение метрических характеристик для программы на **Ассемблере**.

Код программы представлен в приложении Д. Ручной расчёт измеримых характеристик представлен в таблице 7.

Таблица 7 – Измеримые характеристики программы на Ассемблере (ручной подсчёт)

No	Оператор	Количество	No	Операнд	Количество
1	push	1	1	rbp	3
2	mov	23	2	rsp	1
3	pxor	5	3	rdi	1
4	movss	40	4	rsi	1
5	jmp	2	5	rdx	1
6	cdqe	4	6	rcx	1
7	lea	4	7	r8	1
8	add	6	8	r9d	1
9	addss	6	9	xmm0	75
10	mulss	9	10	eax	8
11	jl	2	11	rax	24
12	cvtsi2ss	4	12	xmm1	25
13	divss	6	13	0	6
14	movaps	3	14	xmm2	2
15	subss	4	15	1	2
16	nop	2	16	linfit	1
17	pop	1	17	.L3	2
18	ret	1	18	.L2	2
	Всего	123	19	.L4	2
			20	.L5	2
			21	QWORD PTR [rbp-56]	3
			22	QWORD PTR [rbp-64]	2
			23	QWORD PTR [rbp-72]	2
			24	QWORD PTR [rbp-80]	3
			25	QWORD PTR [rbp-88]	3
			26	DWORD PTR [rbp-92]	7
			27	DWORD PTR [rbp-4]	6
			28	DWORD PTR [rbp-8]	6
			29	DWORD PTR [rbp-12]	5
			30	DWORD PTR [rbp-16]	5
				l .	

31	DWORD PTR [rbp-20]	4
32	DWORD PTR [rbp-24]	5
33	DWORD PTR [rax]	8
34	[0+rax*4]	4
35	DWORD PTR [rbp-48]	4
	Bcero	228

Определение расчетных характеристик представлено в таблице 8.

Таблица 8 – Расчётные характеристики программы на Ассемблере

Характеристика	Ручной расчёт
Число простых операторов n ₁	18
Число простых операндов n ₂	35
Общее число всех операторов N ₁	123
Общее число всех операндов N ₂	228
Словарь п	53
Длина N _{опыт}	351
Teopeтическая длина N _{теор}	254.58
Объём V	2010.50
Потенциальный объём V*	19.65
Уровень программы L	0.007
Оценка уровня программы L~	0.017
Интеллектуальное содержание I	34.29
Работа программирования Е	205705.36
Оценка времени программирования Т^	29612.53
Время программирования Т	11787.27
Уровень языка λ	0.01
Ожидаемое число ошибок в программе В	19

4. Сравнение результатов определения метрических характеристик.

Таблица 9 – Сводная таблица расчетов для всех языков

Характеристика Ручиой подочёт Программны й расчёт Ручной подочёт Программны подочёт Ручной подочёт Ассемблер Ручной подочёт Число простых операторов п, операторов п, операторов п, операторов N, операторов	Таблица 9 — Сводная таблица расчетов для всех языков								
Характеристика подсчёт й расчёт подсчёт ый расчёт подсчёт Число простых операторов п₁ Число простых операндов п₂ общее число всех операторов N₁ 19 20 19 19 35 Общее число всех операндов N₂ операндов N₂ общее число всех операндов N₂ общее число общее общее число общее число общее общее число общее общее число общее общее число общее общее общее общее число общее					Си	Ассемблер			
Число простых операторов n₁ (число простых операторов n₁ (число простых операторов n₁ (число простых операндов n₂ (число всех операндов n₂ (число в	Хапактепистика	Ручной	Программны	Ручной	Программн	Ручной			
16	Zupuixicpiicinia	подсчёт	й расчёт	подсчёт	ый расчёт	подсчёт			
операторов n₁ 19 20 19 19 35 Общее число всех операндов N₁ 94 105 114 131 123 Общее число всех операндов N₂ 70 88 76 92 228 Словарь n 35 34 37 40 53 Длина N₀пыт 164 193 190 223 351 Теоретическая длина N₁тор 133.86 150.84 155.77 196.28 254.58 Объём V 841.16 997.80 989.71 1239.29 2010.50 Потенциальный объём V* 0.020 0.020 0.016 0.007 Уровень программы 0.020 0.020 0.016 0.007 Интеллектуальное содержание I 28.54 31.75 27.49 24.08 34.29 Программирования Е 4 049485.50 78154 205705.36 программирования Пограммирования Т 2479.20 2814.59 3562.96 4341.89 11787.27 Уровень языка λ 0.39 0.39<	Число простых	16	•	18	•	18			
Операндов п₂ Общее число всех операторов № 1 19 20 19 19 35 Общее число всех операндов № 2 Общее число всех операндов № 2 70 88 76 92 228 Словарь п Длина № 2 Словарь п На Объём № 841.16 193 190 223 351 Теоретическая Длина № 2 Объём № 841.16 133.86 150.84 155.77 196.28 254.58 Объём № 841.16 997.80 989.71 1239.29 2010.50 Потенциальный объём № 7 19.65 2010.50 2010.50 Уровень программы Пограммы № 2 Объём № 7 0.034 0.032 0.020 0.016 0.007 Оценка уровня программы Г Работа Программирования Е Оценка времени Программирования В Оценка времени Программирования В Оценка времени Программирования В Оценка времени Программирования В Оценка время В Оценка в Программирования В Оценка время В Оценка время В Оценка время В Оценка в Программирования В Оценка	операторов n ₁	10	17	10	21	10			
Общее число всех операторов N₁ Общее число всех операндов N₂ Общее число всех операндов N₂ Словарь п 35 34 37 40 53 351 Длина N₀тыт 164 193 190 223 351 Теоретическая длина N₁теор Объём V 841.16 997.80 989.71 1239.29 2010.50 Потенциальный объём V* 133.86 150.84 155.77 196.28 254.58	1	19	20	19	19	35			
операторов N₁ 94 105 114 131 123 Общее число всех операндов N₂ 70 88 76 92 228 Словарь п 35 34 37 40 53 Длина N₀шьг 164 193 190 223 351 Теоретическая длина N₁сор 133.86 150.84 155.77 196.28 254.58 Объём V 841.16 997.80 989.71 1239.29 2010.50 Потенциальный объём V* 19.65 19.65 2010.50 Уровень программы 0.020 0.020 0.016 0.007 Оценка уровня программы L° 0.034 0.032 0.028 0.019 0.017 Интеллектуальное содержание І 28.54 31.75 27.49 24.08 34.29 Программирования Е 4 04 49485.50 78154 205705.36 Программирования Т 2479.20 2814.59 3562.96 4341.89 11787.27 Программирования Т 0.39 0.39 <td>-</td> <td></td> <td></td> <td></td> <td></td> <td></td>	-								
Общее число всех операндов № 2 операндов № 2 Словарь п 35 34 37 40 53 Длина № 164 193 190 223 351 Теоретическая длина № 133.86 150.84 155.77 196.28 254.58 Длина № 100 841.16 997.80 989.71 1239.29 2010.50 Потенциальный объём № 19.65 19.65 19.65 2010.50 Уровень программы № 10.020 0.020 0.020 0.016 0.007 Оценка уровня программы № 28.54 31.75 27.49 24.08 34.29 Интеллектуальное содержание І Работа программирования Е 4 50662.60 49485.50 78154 205705.36 программирования Е 4 1361.63 4984.86 3064.49 29612.53 Т^ Время программирования Т Уровень языка № 0.39 0.39 0.39 0.31 0.01 Ожидаемое число ошибок в программе 1 1 1 1 19	·	94	105	114	131	123			
операндов № 1									
Операндов № Словарь п 35 34 37 40 53 Длина № Пеоретическая длина № Пеоретическая длина № Пеоретическая длина № № В41.16 150.84 155.77 196.28 254.58 Объём № В41.16 997.80 989.71 1239.29 2010.50 Потенциальный объём № Ууровень программы № Программи № Прог	Общее число всех	70	88	76	92	228			
Длина N _{опыт} 164 193 190 223 351 Теоретическая длина N _{теор} 133.86 150.84 155.77 196.28 254.58 Объём V 841.16 997.80 989.71 1239.29 2010.50 Потенциальный объём V* 19.65 19.6	операндов N ₂	70	00	7.0	32	220			
Теоретическая длина N _{теор} Объём V 841.16 997.80 989.71 1239.29 2010.50 Потенциальный объём V* Уровень программы программы L° Интеллектуальное содержание І Работа программирования Е Оценка времени программирования Т Т Уровень языка λ Ожидаемое число ошибок в программе 133.86 150.84 155.77 196.28 254.58 254.58 150.84 155.77 196.28 254.58 254.58 254.58 254.58 254.58 254.58 254.59 389.71 1239.29 2010.50 0.007 0.0020 0.016 0.007 0.017 0.017 0.028 0.019 0.017 27.49 24.08 34.29 24.08 34.29 254.58 26.68 26.68 26.68 26.68 26.68 26.68 26.68 26.68 26.68 26.68 27.49 24.08 24.08 24.08 24.08 24.29 24.20 24.2									
Длина N _{теор} Объём V В41.16 997.80 989.71 1239.29 2010.50 Потенциальный объём V* Уровень программы программы L Интеллектуальное содержание I Работа программирования Е Оценка времени программирования программирования Т Уровень языка λ Ожидаемое число ошибок в программе 133.86 150.84 155.77 196.28 254.58 160.84 155.77 196.28 254.58 254.58 160.84 155.77 196.28 254.58 254.58 254.58 1239.29 2010.50 0.007 0.0020 0.016 0.007 0.017 0.017 0.017 0.017 0.017 0.017 0.017 0.017 0.017 0.017 0.017 0.017 0.017 0.017 0.017 0.017 0.017 0.018 0.019 0.017 0.017 0.017 0.018 0.019 0.017 0.017 0.017 0.018 0.019 0.019 0.017 0.017 0.017 0.018 0.019 0.010 0.010 0.020 0.020 0.020 0.020 0.016 0.007 0.019 0.019 0.017 0.017 0.017 0.017 0.017 0.017 0.017 0.018 0.019 0.019 0.010		164	193	190	223	351			
длина N _{теор} 841.16 997.80 989.71 1239.29 2010.50 Потенциальный объём V* 19.65 19.01 <td>Теоретическая</td> <td>122.06</td> <td>150.04</td> <td>155 77</td> <td>106.20</td> <td>25450</td>	Теоретическая	122.06	150.04	155 77	106.20	25450			
Объём V 841.16 997.80 989.71 1239.29 2010.50 Потенциальный объём V* 19.65 29.61 <t< td=""><td>длина N_{теор}</td><td>133.00</td><td>150.04</td><td>155.//</td><td>190.20</td><td>254.50</td></t<>	длина N _{теор}	133.00	150.04	155.//	190.20	254.50			
объём V* 19.65 Уровень программы 0.020 0.020 0.016 0.007 Оценка уровня программы L° 0.034 0.032 0.028 0.019 0.017 Интеллектуальное содержание I 28.54 31.75 27.49 24.08 34.29 Работа программирования Е 4 50662.60 49485.50 78154 205705.36 Оценка времени программирования Т Уровень языка λ 2479.20 2814.59 3562.96 4341.89 11787.27 Уровень языка λ 0.39 0.39 0.39 0.31 0.01 Ожидаемое число ошибок в программе 1 1 1 1 19		841.16	997.80	989.71	1239.29	2010.50			
объём V* Уровень программы 0.020 0.020 0.020 0.016 0.007 Оценка уровня программы L~ 0.034 0.032 0.028 0.019 0.017 Интеллектуальное содержание I 28.54 31.75 27.49 24.08 34.29 Работа программирования Е 4 4 49485.50 78154 205705.36 Программирования Программирования Т 3600.76 1361.63 4984.86 3064.49 29612.53 Время программирования Т 2479.20 2814.59 3562.96 4341.89 11787.27 Уровень языка λ 0.39 0.39 0.39 0.31 0.01 Ожидаемое число ошибок в программе 1 1 1 1 19	Потенциальный			10.05					
Оценка уровня программы L° Интеллектуальное содержание I Работа 35993.9 программирования Е Оценка времени программирования Т Уровень языка \(\lambda\) Ожидаемое число ошибок в программе Оценка уровня долоз доло	объём V*			19.65					
программы L° 0.034 0.032 0.028 0.019 0.017 Интеллектуальное содержание I 28.54 31.75 27.49 24.08 34.29 Работа программирования E 35993.9 50662.60 49485.50 78154 205705.36 Программирования программирования Т Уровень языка λ 2479.20 2814.59 3562.96 4341.89 11787.27 Ожидаемое число ошибок в программе 1 1 1 1 19	Уровень программы	0.020	0.020	0.020	0.016	0.007			
программы L~ 28.54 31.75 27.49 24.08 34.29 Работа программирования Е Оценка времени программирования Т Уровень языка λ Ожидаемое число ошибок в программе 35993.9 50662.60 49485.50 78154 205705.36 3600.76 1361.63 4984.86 3064.49 29612.53 35993.9 2479.20 2814.59 3562.96 4341.89 11787.27 11787.27 11787.27 11787.27 11787.27 11787.27 11787.27 11787.27	Оценка уровня	0.024	0.022	0.020	0.010	0.017			
Содержание I 28.54 31.75 27.49 24.08 34.29 Работа 35993.9 50662.60 49485.50 78154 205705.36 программирования Е 4 4 4984.86 3064.49 29612.53 Программирования Т 2479.20 2814.59 3562.96 4341.89 11787.27 Уровень языка λ 0.39 0.39 0.39 0.31 0.01 Ожидаемое число 0шибок в программе 1 1 1 1 19	программы L~	0.034	0.032	0.028	0.019	0.017			
содержание І З5993.9 35993.9 49485.50 78154 205705.36 программирования Е 4 4984.86 3064.49 29612.53 Программирования Т 3600.76 1361.63 4984.86 3064.49 29612.53 Время программирования Т 2479.20 2814.59 3562.96 4341.89 11787.27 Уровень языка \(\rightarrow\) Ожидаемое число ошибок в программе 1 1 1 1 19	Интеллектуальное	20.54	04.55	27.40	24.00	24.20			
Работа программирования Е Программирования Е Оценка времени Программирования Т^ Время программирования Т3600.76 2479.20 0.391361.63 2814.59 0.394984.86 3562.963064.49 4341.8929612.53 11787.27Уровень языка λ Ожидаемое число ошибок в программе0.39 10.390.390.310.01	солержание I	28.54	31.75	27.49	24.08	34.29			
программирования E 4 50662.60 49485.50 78154 205705.36 Оценка времени программирования 3600.76 1361.63 4984.86 3064.49 29612.53 Т^ Время программирования Т Уровень языка λ 0.39 0.39 0.39 0.31 0.01 Ожидаемое число ошибок в программе 1 1 1 1 1 19		35993.9							
Оценка времени 3600.76 1361.63 4984.86 3064.49 29612.53 Т^ Время программирования Т 2479.20 2814.59 3562.96 4341.89 11787.27 Уровень языка λ 0.39 0.39 0.39 0.31 0.01 Ожидаемое число ошибок в программе 1 1 1 1 19			50662.60	49485.50	78154	205705.36			
программирования 3600.76 1361.63 4984.86 3064.49 29612.53 Т^ Время программирования Т Уровень языка λ 0.39 0.39 0.39 0.31 0.01 Ожидаемое число ошибок в программе 1 1 1 1 1 19	 	4							
T / T / T / T / T / T / T / T / T / T /	, ,								
Время программирования Т Уровень языка λ 0.39 0.39 0.39 0.31 0.01 Ожидаемое число ошибок в программе 1 1 1 1 1 1 19	программирования	3600.76	1361.63	4984.86	3064.49	29612.53			
1 2479.20 2814.59 3562.96 4341.89 11787.27 Уровень языка λ 0.39 0.39 0.39 0.31 0.01 Ожидаемое число ошибок в программе 1 1 1 1 1 19	T^								
программирования Т Уровень языка λ 0.39 0.39 0.39 0.31 0.01 Ожидаемое число ошибок в программе 1 1 1 1 1 19	Время	2.470.20	2014 50	2502.00	40.41.00	11707 27			
Уровень языка λ 0.39 0.39 0.39 0.31 0.01 Ожидаемое число ошибок в программе 1 1 1 1 19	программирования Т	24/9.20	2814.59	3562.96	4341.89	11/8/.2/			
ошибок в программе 1 1 1 1 19	<u> </u>	0.39	0.39	0.39	0.31	0.01			
	Ожидаемое число								
В		1	1	1	1	19			
	В								

В результате сравнения видно, что уровень программы самый низкий у программы на Ассемблере (на порядок меньше чем у Си и Pascal), а самый

высокий у программы на Pascal. Наибольшие показатели времени программирования, работы программирования и ожидаемого числа ошибок, наоборот, соответствуют Ассемблеру, а наименьший – Pascal. Показатели для Си практически не отличаются от Pascal за исключением высокого ожидаемого числа ошибок в программе, однако во всех случаях они хуже.

Выводы

В результате выполнения данной лабораторной работы была изучена система метрик Холстеда. Было проведено сравнение программ, реализующих алгоритм линеаризации, на языках Pascal, Си и Ассемблер.

В результате сравнения видно, что уровень программы самый низкий у программы на Ассемблере (на порядок меньше чем у Си и Pascal), а самый высокий у программы на Pascal. Наибольшие показатели времени программирования, работы программирования и ожидаемого числа ошибок, наоборот, соответствуют Ассемблеру, а наименьший – Pascal. Показатели для Си практически не отличаются от Pascal за исключением высокого ожидаемого числа ошибок в программе, однако во всех случаях они хуже.

ПРИЛОЖЕНИЕ А

КОД ПРОГРАММЫ НА Pascal.

```
procedure linfit(const x,y: array of real; var y calc: array of real; var a,b
: real; n : integer);
{ fit a straight line (y_calc) through n sets of x and y pairs of points }
var i : integer;
sum x, sum y, sum xy, sum x2, sum y2, xi, yi, sxx, syy, sxy : real;
begin
  sum x := 0.0:
  sum y := 0.0;
  sum xy := 0.0;
  sum x2 := 0.0;
  sum_y^2 := 0.0;
  for i := 1 to n do
    begin
      xi := x[i];
      yi := y[i];
      sum x := sum x + xi;
      sum_y := sum_y + yi;
      sum xy := sum xy + xi * yi;
      sum_x2 := sum_x2 + xi * xi;
      sum_y^2 := sum_y^2 + yi * yi;
    end;
  sxx := sum x2 - sum x * sum x / n;
  sxy := sum_xy - sum_x * sum_y / n;
  syy := sum y2 - sum y * sum y / n;
  b := sxy / sxx;
  a := ((sum_x2 * sum_y - sum_x * sum_xy) / n) / sxx;
  for i := 1 to n do
    begin
      y_{calc[i]} := a + b * x[i];
    end;
end;
```

ПРИЛОЖЕНИЕ Б

РЕЗУЛЬТАТЫ РАБОТЫ ПАРСЕРА parser_pas.exe

Statistics for module linfit_pas_pa	arsed.lxm
The number of different operators The number of different operands The total number of operators The total number of operators	:== : 15 : 21 : 105
Time estimation (^T) Programming language level (lambo Work on programming (E)	: 150.842 : 997.796 : 19.6515 : 38.2071 : 0.0196949 : 0.0318182 : 31.748 : 2814.59 : 1361.63 da) : 0.387034 : 50662.6 : 0.456391

Table:

_____ Operators: 1 | 6 | () 2 9 j * 3 6 İ+ 4 4 5 | / 6 40 | ; 6 18 j = 7 8 4 [[] 9 j 2 | for 10 | 2 | integer 1 | linfit 11 12 1 | procedure 13 j 1 | program j real 14 j 4 15 İ 1 | writeln Operands: | 'Hello world!' 1 1 5 2 2 0.0 | 1 3 4 3 | a 5 3 | b 5 6 | i 1 7 | linear_fit 8 7 l n | sum_x 9 | 8 10 j 6 | sum_x2 11 6 | sum_xy | sum_y | sum_y2 12 8 5 13 14 4 SXX

	15	3	sxy
ĺ	16	2	syy
ĺ	17	3	x
ĺ	18	6	xi
ĺ	19	2	ĺУ
ĺ	20	2	y_calc
ĺ	21 j	6	yi

Summary:

The number of different operat The number of different operan The total number of operators The total number of operands		:	15 21 105 88
The cocac hamber of operands		•	00
Programming level (Programming level estimation (Intellect (Time of programming (Time estimation (E)		36 193 150.842 997.796 19.6515 38.2071 0.0196949 0.0318182 31.748 2814.59 1361.63 0.387034 50662.6 0.456391 0.332599

ПРИЛОЖЕНИЕ В

КОД ПРОГРАММЫ НА Си

```
void linfit(const float* x, const float* y, float* y calc, float* a, float*
b, int n) {
    float sum x, sum y, sum xy, sum x2, sum y2, xi, yi, sxx, syy, sxy;
    sum x = 0.0;
    sum_y = 0.0;
    sum xy = 0.0;
    sum x^2 = 0.0;
    sum_{v2} = 0.0;
    for (int i = 0; i < n; i++) {
        xi = x[i];
        yi = y[i];
        sum x = sum x + xi;
        sum y = sum y + yi;
        sum_xy = sum_xy + xi * yi;
        sum x2 = sum x2 + xi * xi;
        sum y2 = sum y2 + yi * yi;
    }
    sxx = sum_x^2 - sum_x * sum_x / n;
    sxy = sum_xy - sum_x * sum_y / n;
    syy = sum y2 - sum y * sum y / n;
    *b = sxy / sxx;
    *a = ((sum x2 * sum y - sum x * sum xy) / n) / sxx;
    for (int i = 0; i < n; i++) {
        y calc[i] = *a + *b * x[i];
}
```

ПРИЛОЖЕНИЕ Г

РЕЗУЛЬТАТЫ РАБОТЫ ПАРСЕРА parser_c.exe

Statistics for module linfit c parsed.lxm : 23 : 21 : 133 The number of different operators The number of different operands The total number of operators : 133 The total number of operands : 94 Dictionary : 44 D) N) : 227 Length Length estimation ^N) : 196.281 (V) Volume : 1239.29 (*V) Potential volume : 19.6515 : 38.2071 Limit volume (**V) Programming level (L) : 0.015857

: 0.0194265 : 24.075 Time of programming : 4341.89 : 3064.49 T) Time estimation (^T) Programming language level (lambda): 0.311614 Work on programming (E) : 78154 Error B) : 0.609321 Error estimation (^B) : 0.413097

I)

Table:

Intellect

Programming level estimation (^L)

Operators: 1 | 6 | () 2 9 3 6 4 2 | ++ 15 | , 5 6 4 | -7 6 | / 8 28 | ; 9 2 | < 10 20 | = i [] 11 | 4 4 12 İ 13 1 14 İ 6 char 15 İ 1 i const 16 2 17 6 | float 18 | | for 2 19 | | int 5 20 | 1 | linfit 21 | 1 | main 22 | return 1 1 | void 23 | Operands: 3 | 0 1 2 5 | 0.0 3 | a 3 4 1 | argc 5 1 argv 6 3 l b

```
7
     10 | i
8
      7
         j n
9
     8
        j sum x
     6
10
        | sum x2
     6
11
        | sum_xy
     8
5
12
        | sum_y
13
          sum_y2
14 j
     4
          SXX
     3
15 j
          sxy
16 j
          syy
     3
17
         | x
     6
18 j
         | xi
     2
19
         Ìу
         | y_calc
20 j
21
     6
         j yī
```

Programming language level

Work on programming

Error estimation

Error

Summary:

The number of different opera	:	23 21					
The total number of operators	:	133					
The total number of operands		:	94				
Dictionary	(D)	:	44				
Length	(N)	:	227				
Length estimation	(^N)	:	196.281				
Volume	(V)	:	1239.29				
Potential volume	(*V)	:	19.6515				
Limit volume	(**V)	:	38.2071				
Programming level	(L)	:	0.015857				
Programming level estimation	(^L)	:	0.0194265				
Intellect	(I)	:	24.075				
Time of programming	(T)	:	4341.89				
Time estimation	(^T)	:	3064.49				

(lambda) : 0.311614

E)

B)

(^B)

: 78154

: 0.609321

: 0.413097

приложение д

КОД ПРОГРАММЫ НА Ассемблер

```
linfit:
                 rbp
        push
        mov
                rbp, rsp
        mov
                QWORD PTR [rbp-56], rdi
        mov
                QWORD PTR [rbp-64], rsi
                QWORD PTR [rbp-72], rdx
        mov
                QWORD PTR [rbp-80], rcx
        mov
                QWORD PTR [rbp-88], r8
        mov
                DWORD PTR [rbp-92], r9d
        mov
                xmm0, xmm0
        noxg
                DWORD PTR [rbp-4], xmm0
        movss
                xmm0, xmm0
        pxor
                DWORD PTR [rbp-8], xmm0
        movss
                xmm0, xmm0
        pxor
                DWORD PTR [rbp-12], xmm0
        movss
                xmm0, xmm0
        pxor
                DWORD PTR [rbp-16], xmm0
        movss
        pxor
                xmm0, xmm0
                DWORD PTR [rbp-20], xmm0
        movss
                DWORD PTR [rbp-24], 0
        mov
                 .L2
        jmp
.L3:
        mov
                eax, DWORD PTR [rbp-24]
        cdae
                rdx, [0+rax*4]
        lea
                rax, OWORD PTR [rbp-56]
        mov
                rax, rdx
        add
                xmm0, DWORD PTR [rax]
        movss
                DWORD PTR [rbp-44], xmm0
        movss
        mov
                eax, DWORD PTR [rbp-24]
        cdae
        lea
                rdx, [0+rax*4]
                rax, QWORD PTR [rbp-64]
        mov
                rax, rdx
        add
                xmm0, DWORD PTR [rax]
        movss
                DWORD PTR [rbp-48], xmm0
        movss
        movss
                xmm0, DWORD PTR [rbp-4]
                xmm0, DWORD PTR [rbp-44]
        addss
                DWORD PTR [rbp-4], xmm0
        movss
                xmm0, DWORD PTR [rbp-8]
        movss
                xmm0, DWORD PTR [rbp-48]
        addss
                DWORD PTR [rbp-8], xmm0
        movss
                xmm0, DWORD PTR [rbp-44]
        movss
                xmm0, DWORD PTR [rbp-48]
        mulss
                xmm1, DWORD PTR [rbp-12]
        movss
                xmm0, xmm1
        addss
                DWORD PTR [rbp-12], xmm0
        movss
                xmm0, DWORD PTR [rbp-44]
        movss
        mulss
                xmm0, xmm0
        movss
                xmm1, DWORD PTR [rbp-16]
        addss
                xmm0, xmm1
        movss
                DWORD PTR [rbp-16], xmm0
                xmm0, DWORD PTR [rbp-48]
        movss
                xmm0, xmm0
        mulss
                xmm1, DWORD PTR [rbp-20]
        movss
        addss
                xmm0, xmm1
                DWORD PTR [rbp-20], xmm0
        movss
```

```
bbs
                DWORD PTR [rbp-24], 1
.L2:
                eax, DWORD PTR [rbp-24]
        mov
                eax, DWORD PTR [rbp-92]
        cmp
        įΙ
                .L3
                xmm0, DWORD PTR [rbp-4]
        movss
                xmm0, xmm0
        mulss
                         xmm1, DWORD PTR [rbp-92]
        cvtsi2ss
        divss
                xmm0, xmm1
        movaps
                xmm1, xmm0
                xmm0, DWORD PTR [rbp-16]
        movss
        subss
                xmm0, xmm1
        movss
                DWORD PTR [rbp-32], xmm0
        movss
                xmm0, DWORD PTR [rbp-4]
        mulss
                xmm0, DWORD PTR [rbp-8]
        cvtsi2ss
                         xmm1, DWORD PTR [rbp-92]
        divss
                xmm0, xmm1
        movaps
                xmm1, xmm0
                xmm0, DWORD PTR [rbp-12]
        movss
        subss
                xmm0, xmm1
                DWORD PTR [rbp-36], xmm0
        movss
                xmm0, DWORD PTR [rbp-8]
        movss
                xmm0, xmm0
        mulss
                         xmm1, DWORD PTR [rbp-92]
        cvtsi2ss
                xmm0, xmm1
        divss
                xmm1, xmm0
xmm0, DWORD PTR [rbp-20]
        movaps
        movss
                xmm0, xmm1
        subss
                DWORD PTR [rbp-40], xmm0
        movss
                xmm0, DWORD PTR [rbp-36]
        movss
        divss
                xmm0, DWORD PTR [rbp-32]
                rax, OWORD PTR [rbp-88]
        mov
                DWORD PTR [rax], xmm0
        movss
                xmm0, DWORD PTR [rbp-16]
        movss
        mulss
                xmm0, DWORD PTR [rbp-8]
                xmm1, DWORD PTR [rbp-4]
        movss
                xmm1, DWORD PTR [rbp-12]
        mulss
        subss
                xmm0, xmm1
        cvtsi2ss
                         xmm1, DWORD PTR [rbp-92]
        divss
                xmm0, xmm1
        divss
                xmm0, DWORD PTR [rbp-32]
                rax, QWORD PTR [rbp-80]
        mov
                DWORD PTR [rax], xmm0
        movss
        mov
                DWORD PTR [rbp-28], 0
        jmp
.L5:
        mov
                rax, QWORD PTR [rbp-80]
                xmm1, DWORD PTR [rax]
        movss
                rax, QWORD PTR [rbp-88]
        mov
                xmm2, DWORD PTR [rax]
        movss
                eax, DWORD PTR [rbp-28]
        mov
        cdae
        lea
                rdx, [0+rax*4]
                rax, QWORD PTR [rbp-56]
        mov
                rax, rdx
        add
        movss
                xmm0, DWORD PTR [rax]
        mulss
                xmm0, xmm2
        mov
                eax, DWORD PTR [rbp-28]
        cdge
        lea
                rdx, [0+rax*4]
                rax, QWORD PTR [rbp-72]
        mov
                rax, rdx
        add
```

```
addss xmm0, xmm1
movss DWORD PTR [rax], xmm0
add DWORD PTR [rbp-28], 1

.L4:

mov eax, DWORD PTR [rbp-28]
cmp eax, DWORD PTR [rbp-92]
jl .L5
nop
nop
pop rbp
ret
```