

Министерство науки и высшего образования Российской Федерации
Муромский институт (филиал)
федерального государственного бюджетного образовательного учреждения высшего образования
«Владимирский государственный университет
Имени Александра Григорьевича и Николая Григорьевича Столетовых»
(МИВлГУ)

Факультет _____ ИТР _____

Кафедра _____ ПИН _____

ЛАБОРАТОРНАЯ РАБОТА №6

по _____ Тестирование ПО _____

Тема _____ Оценка программного средства _____

Руководитель

_____ Колпаков А. А. _____
(фамилия, инициалы)

_____ (подпись) _____ (дата)

Студент ПИН-121 _____
(группа)

_____ Ермилов М.В. _____
(фамилия, инициалы)

_____ (подпись) _____ (дата)

Тема: Оценка программного средства

Задание на лабораторную работу:

1.4 Задания для самостоятельной работы

В задачах, предлагаемых для самостоятельного решения, необходимо выполнить следующее:

- разработать программу, реализующую заданный алгоритм (рекомендуется использовать языки программирования C, C++, C#);
- сформировать словарь программы, охватывающий операнды, операторы и операции;
- словари оформить в виде таблиц;
- рассчитать метрики Холстеда, оформив результат в виде итоговой таблицы;
- провести анализ полученных результатов, сформировав содержательные выводы содержащие оценку качества разработанного текста программы:

- 1) оценить уровень программы, сравнив потенциальный и реальный объемы;
- 2) выявить наличие несовершенств на основе сравнения длины реализации и расчетной длины программы, указать строки программы, содержащие несовершенства и тип несовершенств;
- 3) оценить уровень использования возможностей языка реализации.

Задача 15. Написать и протестировать функцию, которая по заданной строке Str, содержащей буквы и цифры, формирует новую строку, состоящую только из цифр, входящих в Str

Задача 2. Дано натуральное число N. Вывести на экран число, которое получится после выписывания цифр числа N в обратном порядке. Для получения нового числа составить функцию.

					МИВУ.09.03.04							
Из	Лис	№ докум.	Подп.	Дат								
Разраб.												
Провер.												
Н.контр												
Утв.												
					Лит.			Лист		Листов		
								2		12		
					МИ ВлГУ ПИИ - 121							

Задача 15

Код программы

namespace labaTEST

```
{
    internal class Program
    {
        static void Main(string[] args)
        {

            Console.Write("Введите натуральное число: ");

            Console.WriteLine($"Число в обратном порядке: {Reverse(int.Parse(Console.ReadLine()))}");
        }
        static int Reverse(int n)
        {
            int reversedNumber = 0;
            while (n > 0)
            {
                reversedNumber = (reversedNumber * 10) + (n % 10);
                n /= 10;
            }
            return reversedNumber;
        }
    }
}
```

Словарь операторов и операций

№ п/п	Операторы, операции	Номера строк	Количество повторений
1	using	1	1
2	class	2	1
4	static	4	1
5	Main	4	1
6	void	6	1
7	Console.Write	9,11	2
8	Console.WriteLine	10,12	2
9	=	18,21	2
10	%	21	1
11	*	21	1
12	/=	22	1
13	{}	5(50),7(13),17(25),20(23), 10,3(51)	6
14	()	9,10,11,12,16,21,	8
15	namespace	2	1
16	internal	4	1
17	string	6	1

					МИВУ.09.03.04	Исчм
						3
Из	Лис	№ докум.	Подп.	Дат		

18	int	16,18,10	3
19	while	19	1
20	return	24	1
21	+	21	1

Словарь операндов программы:

№ п/п	Операнды	Номера строк	Количество повторений
1	System	1	1
2	Program	4	1
3	Reverse	4	1
5	n	16,19,21,22	4
7	reversedNumber	18,21,24	4
8	0	18,19	2

Входные и выходные переменные программы

Входные переменные	Выходные переменные
args (строковый массив)	- reversedNumber (целое число):

Значения метрик Холстеда для программы

Наименование характеристики	Обозначение и формула для вычисления	Значение
Число простых (уникальных) операторов и операций	$n1$	21
Число простых (уникальных) операндов	$n2$	8
Общее число всех операторов и операций	$N1$	37
Общее число всех операндов	$N2$	13
Число входных и выходных переменных (параметров)	$N*2$	2
Словарь программы	$n=n1+n2$	29
Длина реализации программы	$N = N1 + N2$	50
Длина программы	$N = n1 * \log_2 n1 * n2 * \log_2 n2$	116
Объем программы (в битах)	$V = N \log_2 n$	242
Потенциальный объем программы	$V = n2 + 2 \log_2 n*2 + 2$	33
Уровень реализации программы	$L = V*/V$	0,13
Уровень языка	$\lambda = LV*$	4,29
Работа программирования	$E = V / L$	1861
Число переданных ошибок в программе	$B = LE/3000$	0,08

Выводы:

- 1) Длина программы превышает расчетную длину на 66%. Это может указывать на наличие избыточных операторов и операндов в программе, что усложняет её понимание и поддержку.
- 2) Уровень реализации программы низкий, так как потенциальный объем программы составляет всего 13% от реального объема. Это может свидетельствовать о неэффективном использовании ресурсов программы.
- 3) Уровень языка программирования также невысокий, что может указывать на то, что возможности языка не были полностью использованы.
- 4) Количество переданных ошибок в программе невелико (0,08), что говорит о том, что программа скорее всего стабильна и имеет небольшое количество дефектов.

Задача 15

					МИВУ.09.03.04	Лист
						5
Из	Лис	№ докум.	Подп.	Дат		

Код программы

```
using System;
namespace labaTEST
{
    internal class Program
    {
        static void Main(string[] args)
        {
            Console.Write("Введите строку, содержащую буквы и цифры: ");
            Console.WriteLine($"Строка, состоящая только из цифр:
{ExtractDigits(Console.ReadLine())}");
        }
        static string ExtractDigits(string str)
        {
            string digits = "";
            foreach (char c in str)
            {
                if (char.IsDigit(c))
                {
                    digits += c;
                }
            }
            return digits;
        }
    }
}
```

Словарь операторов и операций

№ п/п	Операторы, операции	Номера строк	Количество повторений
1	using	1	1
2	class	2	1
4	static	4	1
5	Main	4	1
6	void	6	1
7	Console.Write	11	1
8	Console.WriteLine	12	1
12	+=	25	1
13	{ }	3(33),5(32),7(13),15(31),20(28),23(26)	6
14	()	12,11	8
15	namespace	2	1

16	internal	4	1
17	string	6,14,16	4
19	foreach	19	1
20	return	24	1
21	in	19	1
22	char.IsDigit	22	1
23	Console.ReadLine()	12	1
24	char	19,22	2

Словарь операндов программы:

№ п/п	Операнды	Номера строк	Количество повторений
1	System	1	1
2	Program	4	1
3	ExtractDigits	14	1
5	digits	16,25,30	3
6	""	16	1

Входные и выходные переменные программы

Входные переменные	Выходные переменные
args (строковый массив)	digits (строка)

Значения метрик Холстеда для программы

Наименование характеристики	Обозначение и формула для вычисления	Значение
Число простых (уникальных) операторов и операций	n1	24
Число простых (уникальных) операндов	n2	6
Общее число всех операторов и операций	N1	35
Общее число всех операндов	N2	7
Число входных и выходных переменных (параметров)	N*2	2
Словарь программы	n=n1+n2	30
Длина реализации программы	N = N1 + N2	42

					МИВУ.09.03.04	Лист
						7
Из	Лист	№ докум.	Подп.	Дат		

Длина программы	$N = n_1 * \log_2 n_1 * n_2 * \log_2 n_2$	1706
Объем программы (в битах)	$V = N \log_2 n$	206
Потенциальный объем программы	$V = n_2 + 2 \log_2 n^2 + 2$	24
Уровень реализации программы	$L = V^*/V$	0,11
Уровень языка	$\lambda = LV^*$	2,64
Работа программирования	$E = V / L$	1872
Число переданных ошибок в программе	$B = LE/3000$	0,068

Выводы:

Длина реализации незначительно превышает расчетную длину программы.** Это может указывать на наличие небольших несовершенств в программе, таких как избыточное использование комментариев или пробелов.

Уровень реализации относительно низкий.** Потенциальный объем программы меньше ее реального объема, что означает, что возможности языка программирования использованы не в полной мере.

Уровень языка также относительно низкий.** Этот показатель определяется как произведение уровня реализации и потенциального объема программы. Низкий уровень языка указывает на недостаточное использование возможностей языка программирования.

Работа программирования высокая.** Это показатель трудоемкости создания программы. Высокая работа программирования может быть связана с использованием неэффективных или плохо структурированных элементов в программе.

Число переданных ошибок низкое.** Это положительный показатель, указывающий на отсутствие серьезных ошибок в программе.

В целом, программа имеет удовлетворительные метрики Холстеда. Однако есть потенциал для улучшения уровня реализации и использования возможностей языка программирования.

Задача 1. «Зеркальное число»

Словарь для операторов и операций:

№ п/п	Операторы, операции	Номера строк	Количество повторений
1	using	1	1
2	class	2	1
3	public	4	1
4	static	4	1
5	Main	4	1
6	void	4	1
7	Console.Write	8	1
8	Console.WriteLine	23	1
9	Console.ReadLine	24	1
10	uint.Parse	10	1
11	=	7, 11, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22	12

					МИВУ.09.03.04	Лист
						8
Из	Лист	№ докум.	Подп.	Дат		

12	%	13, 16, 19,	3
13	*=	14, 17, 20,	3
14	/=	15, 18, 21,	3
15	{}	3,5,25,26	2
16	()	9,14	2
17	uint	6	1

Словарь операндов программы:

№ п/п	Операнды	Номера строк	Количество повторений
1	System	1	1
2	Revers	2	1
4	ch	6, 10, 11, 13, 16, 19, 20, 21	8
5	copia	6, 11, 13, 15, 18, 19, 20	7
6	cifra	6, 13, 16, 19	4
7	newch	6, 12, 14, 17, 20	5
8	string	7	1
9	str	7, 8, 9, 22	4
10	"Введите трехзначное натуральное число: "	8	1
11	"Если перевернуть "	22	1
12	" будет "	22	1
Всего 35 операндов			

Значения метрик Холстеда для программы

Наименование характеристики	Обозначение и формула для вычисления	Значение
Число простых (уникальных) операторов и операций	n1	17
Число простых (уникальных) операндов	n2	12
Общее число всех операторов и операций	N1	27
Общее число всех операндов	N2	34
Число входных и выходных переменных (параметров)	N*2	2

					МИВУ.09.03.04	Лист
						9
Из	Лист	№ докум.	Подп.	Дат		

Словарь программы	$n=n_1+n_2$	29
Длина реализации программы	$N = N_1 + N_2$	61
Длина программы	$N = n_1 * \log_2 n_1 * n_2 * \log_2 n_2$	130
Объем программы (в битах)	$V = N \log_2 n$	296
Потенциальный объем программы	$V = n_2 + 2 \log_2 n^*2 + 2$	8
Уровень реализации программы	$L = V^*/V$	2,2
Уровень реализации языка	$\lambda = LV^*$	17,6
Работа программирования	$E = V / L$	134
Число переданных ошибок в программе	$B = LE/3000$	0,09

Выводы по таблице:

- 1) Длина реализации программы (61) меньше длины программы (130), что говорит о том, что в программе отсутствуют существенные несовершенства.
- 2) Уровень реализации исследуемой программы (2,2) весьма низкий, так как потенциальный объем программы (8) значительно меньше ее реального объема. Это может указывать на возможные улучшения и оптимизации программы.
- 3) Уровень использования возможностей языка программирования (17,6) находится на достаточно высоком уровне, что свидетельствует о эффективном использовании языка при разработке программы.

Задача 2 «Вычисление суммы элементов массива»

Словарь для операторов и операций:

№ п/п	Операторы, операции	Номера строк	Количество повторений
1	using	1	1
2	namespace	2	1
3	class	4	1
4	static	6, 16, 23, 30	4
5	void	6, 16, 23, 30, 47	5
6	int	6, 8, 13, 16, 18, 23, 25, 32, 39, 41, 33, 40	19
7	[,]	6, 16, 23, 33,	4
8	int[,],	6, 23, 16	3
9	ConsoleKeyInfo	34	1
10	ConsoleKey.Escape	50	1
11	!=	50	1
12	for	9, 19, 24, 45	4
13	Console.Write	12, 38, 44, 48	4
14	Console.ReadLine	14, 39, 49	3
15	int.Parse	14, 39	2
16	=	6, 14, 16, 20, 26, 27, 28, 32, 33, 36, 37, 39, 41, 42, 43, 45, 46, 50	19

17	Console.Clear	37	1
18	+	41,45	2
19	-		
20	{ }	3(63), 5(19), 10(18), 12(17), 21(44), 25(27), 29(31), 33(35), 37(39), 41(43), 46(62), 48(61)	12
21	[]	21, 26, 45	3
22	do	35	1
23		50	1
24	true	50	1
25	/	28	1
26	*	28	1
27	Console.ReadKey	49	1
28	while	50	1

Словарь операндов программы:

№ п/п	Операнды	Номера строк	Количество повторений
1	System	1	1
2	Prim	2	1
3	Summ	4	1
4	Read	6	1
5	a	6, 13,16,23,27,33,41,42,44,46	10
6	n	6,9,16,19,32,39,41,44,45	9
7	m	6,10,16,20,32,40,41,42,44	9
8	i	8,9,12,13,18,19,21,	11
9	j	8,10,13,18,20,21,25,26,27	15
10	klav	34,49,50	3
11	0,8	44	1
12	s	45,46,47	
13	nom	23,27,32,45,46	7
14	perv	23,26,32,45,46	6
15	posl	23,26,32,45,46	6

Наименование характеристики	Обозначение и формула для вычисления	Значение
Число простых (уникальных) операторов и операций	n1	28
Число простых (уникальных) операндов	n2	15
Общее число всех операторов и операций	N1	98

					МИВУ.09.03.04	Лист
						11
Из	Лист	№ докум.	Подп.	Дат		

Общее число всех операндов	N_2	81
Число входных и выходных переменных (параметров)	N^*2	4
Словарь программы	$n=n_1+n_2$	43
Длина реализации программы	$N = N_1 + N_2$	179
Длина программы	$N = n_1 * \log_2 n_1 * n_2 * \log_2 n_2$	193
Объем программы (в битах)	$V = N \log_2 n$	971
Потенциальный объем программы	$V = n_2 + 2 \log_2 n^*2 + 2$	69
Уровень реализации программы	$L = V^*/V$	14
Уровень языка	$\lambda = LV^*$	966
Работа программирования	$E = V / L$	69
Число переданных ошибок в программе	$B = LE/3000$	0,2

Выводы:

- 1) Длина реализации (179) больше расчетной длины программы (193) на 8%. Это может указывать на наличие несовершенств в программе, таких как избыточный код или неоптимальные алгоритмы.
- 2) Уровень реализации программы (14) весьма низкий, поскольку потенциальный объем программы (69) значительно меньше ее реального объема (971). Это означает, что программа не использует возможности языка программирования в полной мере.
- 3) Возможности языка программирования использованы на низком уровне, поскольку уровень языка (966) значительно выше уровня реализации (14). Это может указывать на то, что программа написана на высокоуровневом языке, но не использует его преимущества, такие как автоматическое управление памятью или объектно-ориентированное программирование.

Вывод: получены навыки работы с оценкой программного средства на основе метрик Холстеда.