

Министерство образования Республики Беларусь
Учреждение образования «Белорусский государственный университет
информатики и радиоэлектроники»

Факультет компьютерных систем и сетей

Кафедра информатики

Дисциплина «Метрология, стандартизация и сертификация (в
информационных технологиях)»

ОТЧЕТ

к лабораторной работе №1

на тему:

«Метрики размера программ»

БГУИР 1-40-04-01

Выполнили студенты группы 253504
Фроленко Кирилл Юрьевич
Давыдовский Дмитрий Владимирович

Проверил ассистент кафедры
ПОИТ

Болтак Светлана Владимировна

(дата, подпись преподавателя)

Минск 2024

Листинг 1 – Исходный код программы

```
func functionName1(parameter1: Type1, parameter2:
Type2, parameter3: Type1) -> ReturnType1 {
    let x = 42
        var y = x + 10
        if y > 0 {
            print("Positive")
        } else if x < 0 {
            print("Negative")
        } else {
            print("Zero")
        }

}

func functionName2(_ parameter: Type) -> ReturnType2 {
    switch value {
        case 1:
            print("Case 1")
        case 2:
            print("Case 2")
        default:
            print("Default Case")
    }

    switch anotherValue {
        case "A":
            print("Case A")
        case "B":
            print("Case B")
    }
}

func functionName3() -> ReturnType3 {
    # function body
}

func outerFunction() -> ReturnType3 {
    let outerVariable = 10

    for item in array {
        print(item)
    }
}
```

```

        for number in 1...5 {
            print(number)
        }

switch value {
    case 1:
        print("Case 1")
    case 2:
        print("Case 2")
    default:
        print("Default Case")
}

func innerFunction() {
    print("Inside innerFunction, outerVariable is")
}

innerFunction()
let result = someFunction(arg1: "value", arg2: 42)
let anotherResult = anotherFunction()
}

func generateFibonacci(limit: Int) -> [Int] {
    var fibonacciSeries: [Int] = []

    var a = 0, b = 1
    while a <= limit {
        fibonacciSeries.append(a)
        let temp = a + b
        a = b
        b = temp
    }

    var c = 0, d = 1
    while true {
        let nextValue = c + d
        if nextValue > limit {
            break
        }
        fibonacciSeries.append(nextValue)
        c = d
        d = nextValue
    }
}

```

```

    var x: Int = 42;
    let y: Double = 3.14;
    let aa: Int = 42;
    return fibonacciSeries
}

```

Таблица операторов и операндов				
#	Операторы	Количество	Операнды	Количество
1	func	6	parameter1	1
2	functionName1	1	parameter2	1
3	ReturnType1	1	parameter3	1
4	Type1	2	parameter	1
5	Type2	1	limit	1
6	functionName2	1	"Positive"	1
7	ReturnType2	1	"Negative"	1
8	Type	1	"Zero"	1
9	functionName3	1	"Case 1"	2
10	ReturnType3	2	"Case 2"	2
11	outerFunction	1	"Default Case"	2
12	innerFunction	1	"Case A"	1
13	enerateFibonacci	1	"Case B"	1
14	[Int]	1	item	2
15	Int	3	number	2
16	print()	14	side innerFunction	1
17	innerFunction()	1	outerVariable is'	1

Словарь программы: $n = 35 + 34$
 Длина программы: $N = 94 + 43$
 Объем программы: $y = 137 * \log_2(69)$

Рисунок 1 – результат работы программы

Таблица 1 – самостоятельно вручную рассчитанные метрики

#	Операторы	f_{ij}	I	Операнды	f_{2i}
1.	func	6	1.	parameter1	1
2.	functionName1	1	2.	parameter2	1
3.	ReturnType1	1	3.	parameter3	1
4.	Type1	2	4.	parameter	1
5.	Type2	1	5.	limit	1
6.	functionName2	1	6.	“Positive”	1
7.	ReturnType2	1	7.	“Negative”	1
8.	Type	1	8.	“Zero”	1
9.	functionName3	1	9.	“Case 1”	2
10.	ReturnType3	2	10.	“Case 2”	2
11.	outerFunction	1	11.	“Default Case”	2
12.	innerFunction	1	12.	“Case A”	1
13.	generateFibonacci	1	13.	“Case B”	1
14.	[int]	1	14.	Item	2
15.	int	3	15.	number	2
16.	print()	14	16.	“Inside innerFunction”	1
17.	innerFunction()	1	17.	“outerVariable is”	1
18.	someFunction()	1	18.	arg1: “value”	1
19.	anotherFunction()	1		arg2: 42	1
20.	append()	2		a	1
21.	switch	3		nextValue	1
22.	default	2		value	2
23.	case	6		1	2
24.	for	2		2	2
25.	while	2		anotherValue	1
26.	->	5		“A”	1
27.	>	2		“B”	1

28	=	19		array	1
29.	+	3		1...5	1
30.	<	1		x	1
31.	<=	1		42	2
32.	If	2		y	1
33.	else if	1		3.14	1
34.	else	1		aa	1
35.	Double	1			
	n1 = 35	N1 = 94		n2 = 34	N2 = 43

Словарь программы:

$$n = n1 + n2 = 35 + 34 = 69$$

Длина программы:

$$N = N1 + N2 = 94 + 43 = 137$$

Объем программы:

$$y = N * \log_2(n) = 137 * \log_2(69) \approx 837$$