|  |
| --- |
| Министерство образования Республики Беларусь  УО «Полоцкий государственный университет им. Ефросинии Полоцкой» |

|  |  |
| --- | --- |
|  | Факультет информационных технологий  Кафедра технологий программирования |

|  |
| --- |
| ОТЧЕТ |
| **к лабораторной работе №6**  **Дисциплина**: «Программирование на платформе .NET» |

|  |  |
| --- | --- |
| Выполнил: | Студент группы 19-ИТ-2  Гилимович А.Н. |

|  |  |
| --- | --- |
| Проверила: | Преподователь  Виноградова А.Д. |

|  |
| --- |
| Полоцк, 2022 г. |

**Задание**

1. Создайте два варианта функции, которая возвращает кортеж значений. Первый вариант принимает на вход параметры в виде кортежа, второй вариант параметры в каррированном виде.

2. Выберите простой алгоритм, который может быть реализован в виде рекурсивной функции и реализуйте его в F#. Пример – вычисление суммы целых чисел в заданном диапазоне.

3. Преобразуйте разработанную рекурсивную функцию в форму хвостовой рекурсии.

4. По аналогии с пунктом 3.7.4 разработайте конечный автомат из трех состояний и реализуйте его в виде взаимно-рекурсивных функций.

5. На основе пунктов 3.7.7 и 3.7.8 разработайте функцию, которая принимает 3 целых числа и лямбда-выражение для их суммирования в виде кортежа и в каррированном виде.

**Ход работы**

Изучив теоритический материал, преступил к выполнению заданий.

**Листинг 1** – задание 1

printf "\nЗадание 1"

let func1(a:int, b:int, c:int) = a + b + c;

let func2(a: int)(b: int)(c: int) = a + b + c;

let a = func1(1, 2, 2)

let b = func2(1)(2)(3)

printf "\nОтвет: %A" a

printf "\nОтвет: %A" b

Рузультат работы показан на рисунке 1.



**Рисунок 1** – Задание 1

**Листинг 2** – задание 2

let rec recursFunc(a:int, b:int):int =

if b<=a then a

else b + recursFunc(a, b-1)

let ex2 = recursFunc(2, 10)

printf "\nСумма от 2 до 10 = %A" ex2

Рузультат работы показан на рисунке 2.



**Рисунок 2** – Задание 2

**Листинг 3** – задание 3

let rec tailRec(a:int, b:int, acc:int):int =

if b <= a then acc+a

else tailRec(a, b-1, b+acc)

let rec tailRec2(a, b) = tailRec(a, b, 0)

let ex3 = tailRec2(3, 7)

printf "\nСумма от 3 до 7 = %A" ex3

Рузультат работы показан на рисунке 3.



**Рисунок 3** – Задание 3

**Листинг 4** – задание 4

let rec State1(x:int) =

printfn "%i - (+1) %i" x (x+1)

let x\_next = x+1

if x\_next>5 then State2(x\_next)

else

State1(x\_next)

and State2(x:int) =

printfn "%i - (^2) %i" x (x\*x)

let x\_next = x+1

if x\_next>10 then State3(x\_next)

else

State2(x\_next)

and State3(x:int) =

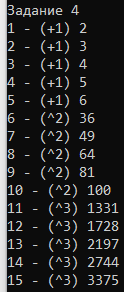
printfn "%i - (^3) %i" x (x\*x\*x)

let x\_next = x+1

if x\_next<=15 then State3(x\_next)

State1(1)

Рузультат работы показан на рисунке 4.



**Рисунок 4** – Задание 4

**Листинг 5** – задание 5

let lambda1 = fun (a:int, b:int, c:int) -> a+b+c

let ex4\_1 = lambda1(1,2,3)

let del1(a:int, b:int, func1: int\*int->int) = func1(a, b)

let ex4\_2 = del1(1, 3, fun(a, b) -> a+b);

printf "\nЛямбда = %A" ex4\_1

printf "\nПринимает лямбду = %A" ex4\_2

Рузультат работы показан на рисунке 5.



**Рисунок 5** – Задание 5

**Вывод:** изучили основы F# функций, фналог делигатов, лямбда-выражения, каррирование, понятие хвостовых и обычных рекурсий, реализовали их, а также взаимно-рекурсивные функции и эмуляцию конечного автомата.