**Московский государственный технический**

**университет им. Н.Э. Баумана**

Факультет «Радиотехнический»

Кафедра «Системы обработки информации и управления»

Курс «Парадигмы и конструкции языков программирования»

Отчет по лабораторной работе №5

«Функциональное программирование на F#»

|  |  |  |
| --- | --- | --- |
| Выполнил: |  | Проверил: |
| студент группы РТ5-31Б: |  | преподаватель кафедры ИУ5 |
| Савельева В. О. |  | Гапанюк Ю. Е. |
| Подпись и дата: |  | Подпись и дата: |

Москва, 2024 г.

Постановка задачи

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. Разработайте конечный автомат из трех состояний и реализуйте его в виде взаимно-рекурсивных функций.
5. Разработайте функцию, которая принимает 3 целых числа и лямбдавыражение для их суммирования в виде кортежа и в каррированном виде.

Текст программы

let cort1 (x, y, z) = (x, y, z)

let cort2 x y z = (x, y, z)

let rec sumRec start finish =

    if start > finish then 0

    else start + sumRec (start + 1) finish

let sumTailRec start finish =

    let rec aux current acc =

        if current > finish then acc

        else aux (current + 1) (acc + current)

    aux start 0

let rec stateA input =

    match input with

    | "toB" -> stateB input

    | \_ -> printfn "State A: %s" input; stateA input

and stateB input =

    match input with

    | "toC" -> stateC input

    | \_ -> printfn "State B: %s" input; stateB input

and stateC input =

    match input with

    | "toA" -> stateA input

    | \_ -> printfn "State C: %s" input; stateC input

let sumTuple (a, b, c) =

    (a + b + c)

let sumCarr a b c =

    a + b + c

[<EntryPoint>]

let main argv =

    let a = 1;

    let b = -2;

    let c = 3;

    let sc = sumCarr a b c;

    let st = sumTuple(a,b,c);

    let r1 = sumRec 10 20;

    let r2 = sumTailRec 10 20;

    // stateA "";

    printfn "sumRec: %d" r1;

    printfn "sumTailRec: %d" r2;

    printfn "sumTuple: %d" st;

    printfn "sumCarr: %d" sc;

    0

Результат

