

Отчёт по лабораторной работе № 2

Волгина А. Д., ИУ5-33Б

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 лямбда-выражение для их суммирования в виде кортежа и в каррированном виде.

open System

[<EntryPoint>]

let main argv =

let AddOne1 (x, y) =

printfn "Pair %A %A" (x + 1) (y + 1)

(x + 1, y + 1)

let AddOne2 x y =

printfn "Pair %A %A" (x + 1) (y + 1)

(x + 1, y + 1)

let x = 1

let y1 = 3

let y2 = 7

let CurriedAddOne = AddOne2 x

printfn "Карринг"

```

CurriedAddOne y1
CurriedAddOne y2
printfn "Функция с кортежом"
AddOne1 (x, y1)
let rec Summ (a : int) (b : int) : int =
    if a < b then a + Summ (a + 1) b
    else b
printfn "Рекурсивная функция"
let s = Summ y1 y2
printfn "%i" s
printfn "Хвостовая рекурсия"
let rec TailSumm (a : int) (b : int) (c : int) : int =
    if a < b then TailSumm (a + 1) b (c + a + 1)
    else c
let s = TailSumm (y1 + 1) (y2 + 1) (y1 + 1)
printfn "%i" s
let rec Rem0 x =
    if x = 0 then
        printf("Первая цифра 0\n")
        true
    else if x % 3 = 0 then Rem0 (x / 3)
    else if x % 3 = 1 then Rem1 (x / 3)
    else Rem2(x / 3)
and Rem1 x =
    if x = 0 then
        printf("Первая цифра 1\n")

```

```

    true

    else if x % 3 = 0 then Rem0 (x / 3)

    else if x % 3 = 1 then Rem1 (x / 3)

    else Rem2(x / 3)
and Rem2 x =
    if x = 0 then
        printf("Первая цифра 2\n")
        true
    else if x % 3 = 0 then Rem0 (x / 3)
    else if x % 3 = 1 then Rem1 (x / 3)
    else Rem2(x / 3)
printf("Конечный автомат, перевод в третичную систему\n")
let s = Rem0 15

let Apply (transform : int*int*int -> int) (a, b, c) = transform (a, b, c)
let s = Apply (fun (a, b, c) -> a + b + c) (1, 2, 3)
printfn "Функция с лямбда-выражением, с кортежом"
printfn "%i" s

let CurriedApply (transform : int -> int -> int -> int) a b c = transform a b c
let Apply2 = CurriedApply (fun a b c -> a + b + c) 3 4
let s1 = Apply2 5
let s2 = Apply2 6

printfn "Функция с лямбда-выражением, с каррингом"
printfn "%i" s1
printfn "%i" s2
0

```

Карринг
Pa1r 2 4
Pa1r 2 8
Функция с кортежом
Pa1r 2 4
Рекурсивная функция
25
Хвостовая рекурсия
30
Конечный автомат, перевод в третичную систему
Первая цифра 1
Функция с лямбда-выражением, с кортежом
6
Функция с лямбда-выражением, с каррингом
12
13
C:\Users\Anna\Desktop\Учёба\Labs\F#\Lab2\Lab2\bin\Debug\netcoreapp3.1\Lab2.exe (процесс 17340) завершил работу с кодом 0
Нажмите любую клавишу, чтобы закрыть это окно...