



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ БИОМЕДИЦИНСКАЯ ТЕХНИКА

КАФЕДРА БИОМЕДИЦИНСКИЕ ТЕХНИЧЕСКИЕ СИСТЕМЫ (БМТ-1)

НАПРАВЛЕНИЕ ПОДГОТОВКИ 09.03.03 Прикладная информатика (Цифровые биомедицинские системы)

О Т Ч Е Т

по лабораторной работе № 4

Название: Пользовательские структурные типы и операции ввода-вывода

Дисциплина: Алгоритмизация и программирование

Студент БМТ1-13Б

(Группа)

Каримова
Алина
Равильевна

(И.О. Фамилия)

(Подпись, дата)

Преподаватель

Т.А.Ким

(И.О. Фамилия)

(Подпись, дата)

Задание 1

Решить задачу, используя функциональную декомпозицию кода, самостоятельно разработанные структурные типы и рассмотренные на лекции приёмы корректной работы с файловым вводом-выводом. В работе необходимо показать умение:

1. Отделения вычисляющих(чистых) функций от функций ввода-вывода.
2. Использование структурных типов для промежуточного хранения результатов вычислений и передачи их между функциями вычисления и функциями ввода-вывода.
3. Написания юнит-тестов написанных функций.

Задан файл F, состоящий из записей. Каждая запись содержит название города и год его основания. Переписать в файл G сведения обо всех городах, название которых начинается на букву «А». Вывести на экран содержимое обоих файлов.

Исходный код

```
mutable struct info
    name::String
    age::Int

    function info(name, age)
        new(name, age)
    end
end

function input()
    f =
open("C:/Users/Lenovo/Desktop/Данные/User/Desktop/Study/Programming/Julia/Laba4/data.
txt", "r")
    mas = []
    while (!eof(f))#end of file
        str = readline(f)
        name = split(str, " ")[1]
        age = parse{Int, split(str, " ")[2]}
        push!(mas, info(name, age))
    end
    close(f)
    return mas
end

function get(mas) #(дать: сет, вернуть: сет с городами на А)
    new_mas = []
    for i in range(1, length(mas))
        if mas[i].name[1] == 'A'
            push!(new_mas, mas[i])
        end
    end
end
```

```

        return new_mas
    end

function new_file_add(new_mas)
    open("C:/Users/Lenovo/Desktop/Данные/User/Desktop/Study/Programming/Julia/Laba4/new_data.txt", "w") do file
        for i in range(1, length(new_mas))
            println(file, new_mas[i].name, " ", new_mas[i].age)
        end
    end
    return
end

open("C:/Users/Lenovo/Desktop/Данные/User/Desktop/Study/Programming/Julia/Laba4/new_data.txt", "r")
end

function output(f)
    while (!eof(f))#end of file
        str = readline(f)
        println(str)
    end
    println()
    close(f)
end

new_mas = get(input())
new_file_add(new_mas)

println("File:")
output(open("C:/Users/Lenovo/Desktop/Данные/User/Desktop/Study/Programming/Julia/Laba4/data.txt", "r"))

println("New file:")
output(new_file_add(new_mas))

```

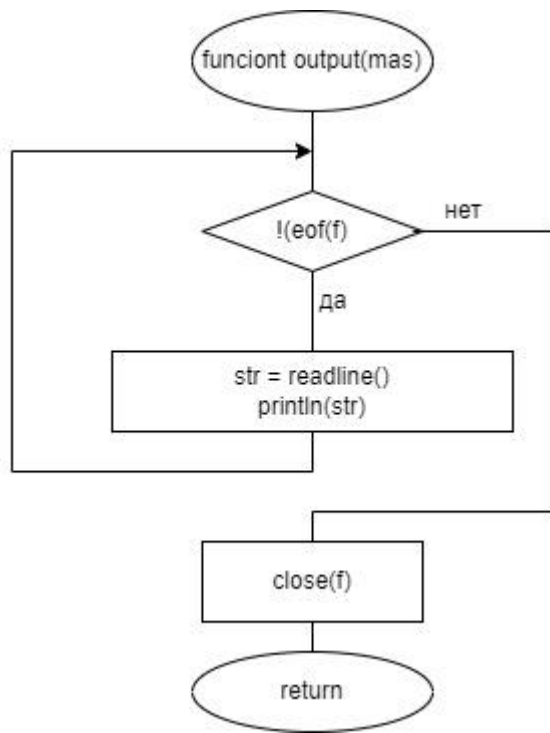
Схема алгоритма

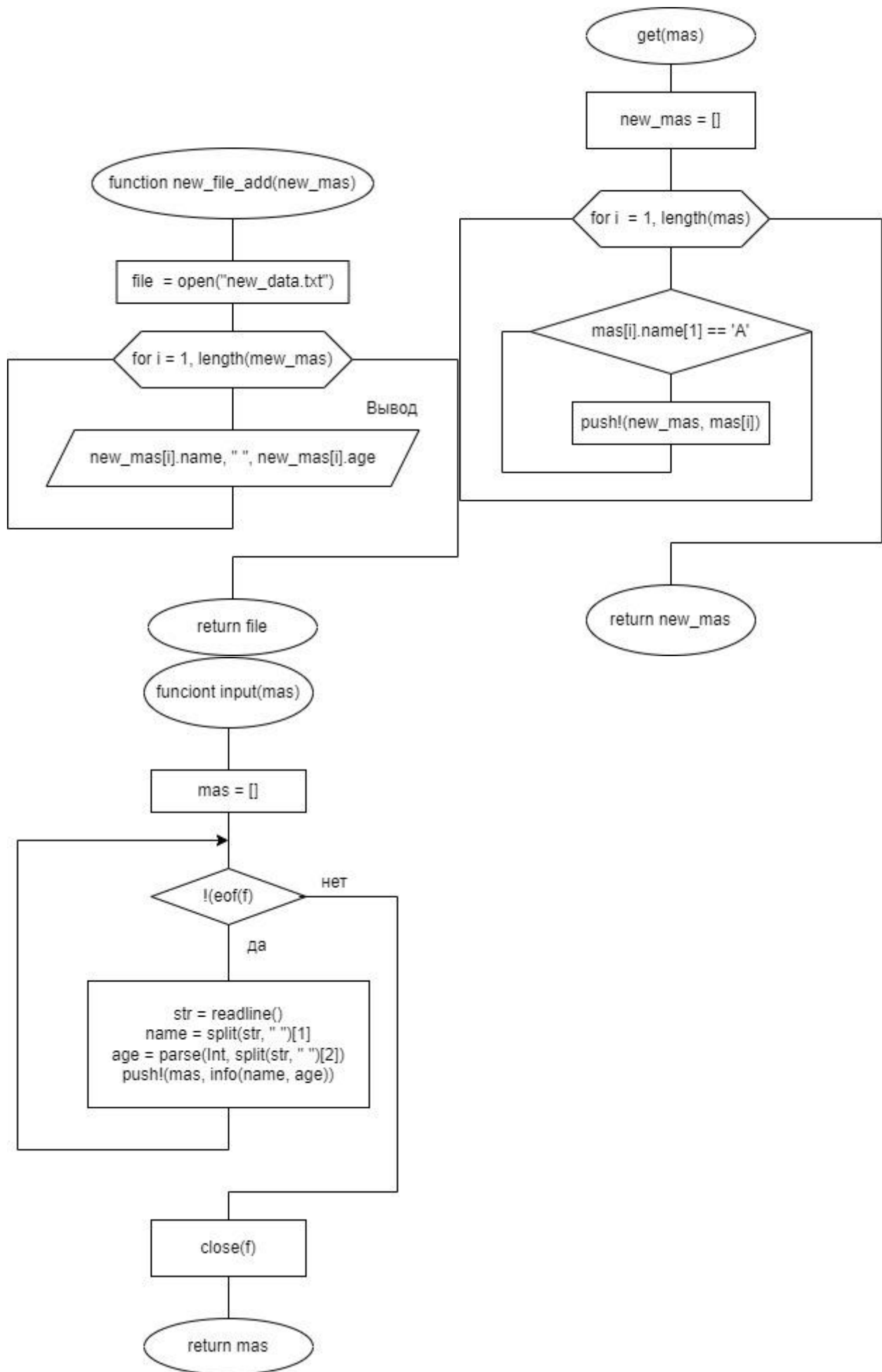
```

mutable struct info
    name::String
    age::Int

    function info(name, age)
        new(name, age)
    end
end

```





Тестирование алгоритма

Приводим результаты тестирования. Описываем сценарии проверок, ожидаемый результат и полученный результат. Делаем выводы...

Наименование проверки	Исходные данные	Ожидаемый результат	Полученный результат	Вывод
Основная	Kot 1528 Afada 1372 Proom 856 Neet 1200 Ayiro 300 Banda 100 Alfonso 87 Agrant 4006	File: Kot 1528 Afada 1372 Proom 856 Neet 1200 Ayiro 300 Banda 100 Alfonso 87 Agrant 4006 New file: Afada 1372 Ayiro 300 Alfonso 87 Agrant 4006	File: Kot 1528 Afada 1372 Proom 856 Neet 1200 Ayiro 300 Banda 100 Alfonso 87 Agrant 4006 New file: Afada 1372 Ayiro 300 Alfonso 87 Agrant 4006	+

Выводы

Научилась работать с файлами, структурными типами данных. Познакомилась с особенностями оператора сравнения в Julia.