|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **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)**

**Отчет**

|  |  |
| --- | --- |
| **по лабораторной работе №** | 4 |

**Название:**

Пользовательские структурные типы и операции ввода-вывода

**Дисциплина:** Алгоритмизация и программирование

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Студент | БМТ1-13Б |  |  | Н.А.Сухов |
|  | (Группа) |  | (Подпись, дата) | (И.О. Фамилия) |
|  |  |  |  |  |
| Преподаватель |  |  |  | Т.А.Ким |
|  |  |  | (Подпись, дата) | (И.О. Фамилия) |

Москва, 2022

# Задание. Пользовательские структурные типы и операции ввода-вывода.

Задан файл F, состоящий из записей. Каждая запись содержит название города и год его основания. Определить самый старый город и переписать в файл G сведения обо всех городах, основанных в том же году. Вывести на экран содержимое обоих файлов.

# Исходный код

MainModule.jl

module MainModule

  include("FunctionModule.jl")

  using .FunctionModule

  #Переменные

  data = []

  file\_path = "F.txt"

  gile\_path = "G.txt"

  #Основной блок

  in\_arr(file\_path, data)

  find\_ancient\_town(gile\_path, data)

  #Вывод данных

  println("Данные исходного файла:")

  file\_out(file\_path)

  println("Данные конечного файла: ")

  file\_out(gile\_path)

end

FunctionModule.jl

module FunctionModule

  export find\_minimal, in\_arr, file\_out, find\_ancient\_town

  function find\_minimal(arr::Array, row::Int64)

    minimal\_e = arr[1][row]

    for i in 1:length(arr)

      if arr[i][row] < minimal\_e

        minimal\_e = arr[i][row]

      end

    end

    return minimal\_e

  end

  function in\_arr(file\_path, arr::Array)

    file = open(file\_path)

    while (!eof(file))

      push!(arr, split(readline(file)))

    end

    close(file)

  end

  function file\_out(file\_path)

    file = open(file\_path)

    for i in eachline(file)

      println(i)

    end

    close(file)

  end

  function find\_ancient\_town(file\_path,arr::Array)

    file = open(file\_path, "w")

    minimal\_year = find\_minimal(arr, 2)

    for i in 1:length(arr)

      if  arr[i][2] == minimal\_year

        element = arr[i][1]\*" "\*arr[i][2]\*"\n"

        write(file, element)

      end

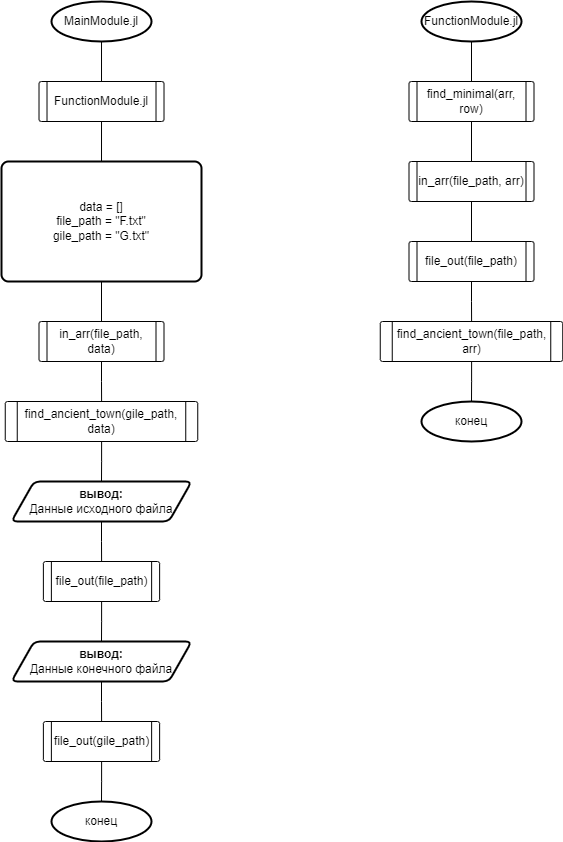
    end

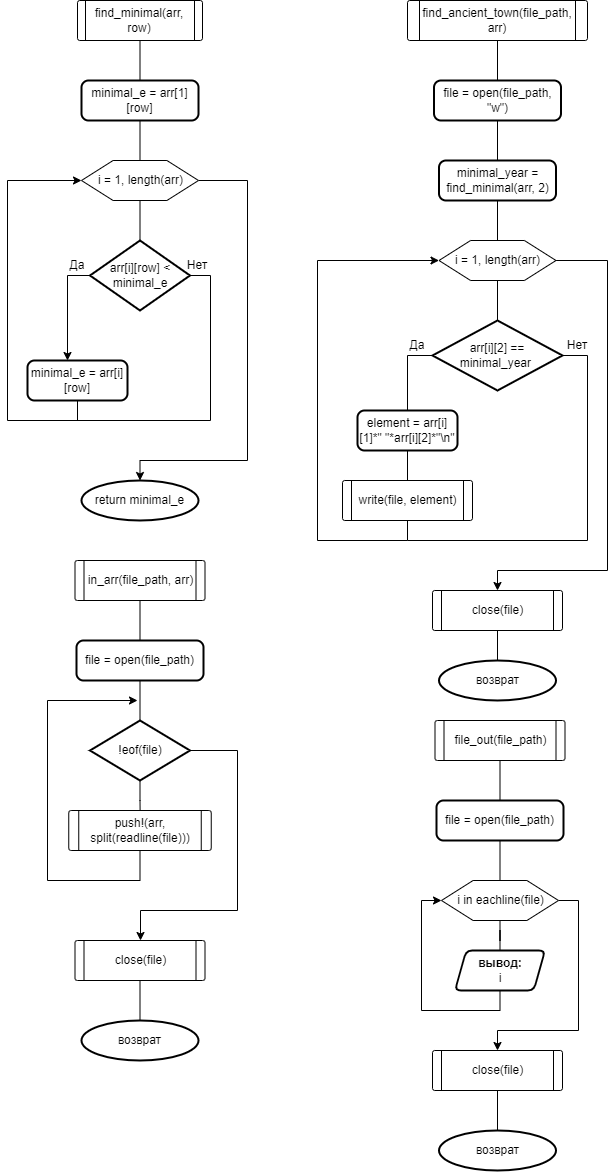
    close(file)

  end

end

# Схема алгоритма





# Тестирование алгоритма

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Наименование проверки** | **Данные на вход** | **Ожидаемый результат** | **Полученный результат** | **Вывод** |
| Нахождение древнейшего города | F.txt  { Groove 1245  Rowtown 1356  Freed 1800  Friburg 1917  hraptown 1654  Roomp 1245  Totanhomoncity 10  KleopatraParadise 10  Aphine 123  Sparta 120} | Totanhomoncity 10  KleopatraParadise 10 | Totanhomoncity 10  KleopatraParadise 10 | Программа сработала в соответствии с ожиданиями |

# Выводы

При выполнении лабораторной работы я научился работать с файлами: доставать содержимое, обрабатывать данные внутри файла и записывать данные в файл.