1. Диаграмма деятельности

@startuml

start

:Запуск программы;

:Считывание информации из файла;

if(информация считалась не корректно)

:Введите корректную информацию;

:Считывание информации из файла;

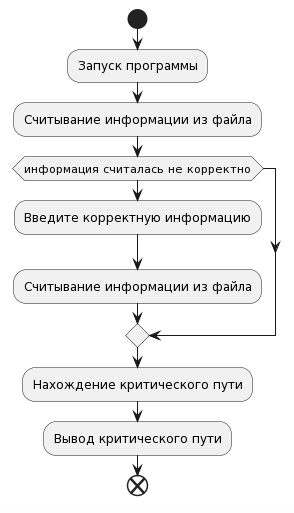
endif

:Нахождение критического пути;

:Вывод критического пути;

end

@enduml



1. Диаграмма компонентов

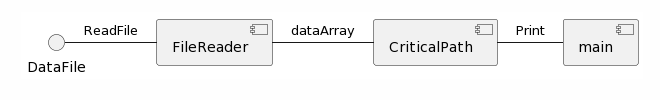
@startuml

DataFile - [FileReader] : ReadFile

[FileReader] - [CriticalPath] : dataArray

[CriticalPath] - [main] : Print

@enduml

****

1. Диаграмма классов

@startuml

class Program

{

path : string

}

Program --> FileReader

class FileReader

{

dataArray : string[,]

ReadFile(string)

}

class CriticalPath

{

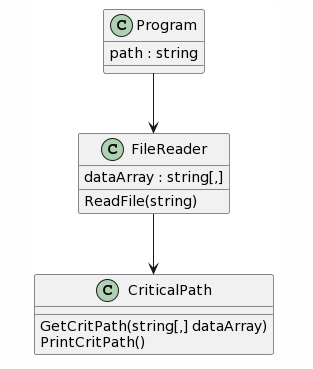
GetCritPath(string[,] dataArray)

PrintCritPath()

}

FileReader --> CriticalPath

@enduml



1. Диаграмма вариантов использования

@startuml

left to right direction

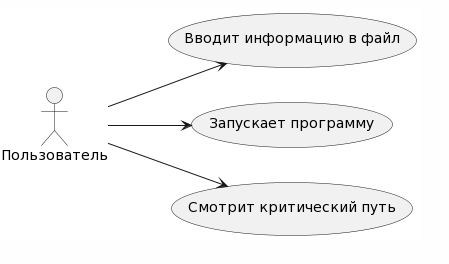
:Пользователь:

Пользователь --> (Вводит информацию в файл)

Пользователь --> (Запускает программу)

Пользователь --> (Смотрит критический путь)

@enduml



1. Диаграмма последовательности

@startuml

Пользователь -> Программа : Запускает прогармму

activate Программа

Программа -> FileReader : Считывание из файла

activate FileReader

FileReader -> CriticalPath : Получить критический путь

deactivate FileReader

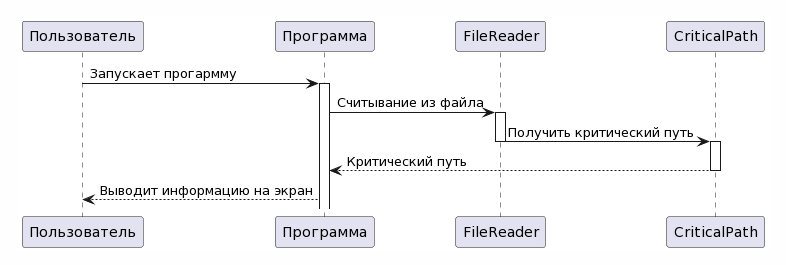
activate CriticalPath

CriticalPath --> Программа : Критический путь

deactivate CriticalPath

Программа --> Пользователь : Выводит информацию на экран

@enduml



1. Диаграмма развертывания
2. Диаграмма состояний

@startuml

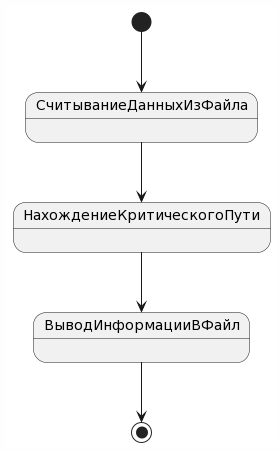
[\*] --> СчитываниеДанныхИзФайла

СчитываниеДанныхИзФайла --> НахождениеКритическогоПути

НахождениеКритическогоПути --> ВыводИнформацииВФайл

ВыводИнформацииВФайл --> [\*]

@enduml



1. Диаграмма потоков данных

