Диаграмма развёртывания

@startuml

node ПК {

artifact Критические\_пути

component Расчет

file Цепочки\_работ

}

Цепочки\_работ -> Расчет

Расчет -> Критические\_пути

@enduml

Диаграмма потоков данных

@startuml

person Пользователь

file Файл\_Цепочки\_Работ

rectangle Программа\_расчета

file Файл\_Критические\_пути

Пользователь -> Файл\_Цепочки\_Работ :Исходные данные

Файл\_Цепочки\_Работ -> Программа\_расчета :Подготовленные данные для расчета

Программа\_расчета -> Файл\_Критические\_пути :Результат

Файл\_Критические\_пути -> Пользователь

@enduml

Диаграмма деятельности

@startuml

|User|

start

:Ввод файла списка работ и\n их длительности;

:Чтение данных из файл списка работ;

|Программа|

:Расчет критических путей;

|User|

:Вывод результатов в файл;

stop

@enduml

rectangle "Интерфейс\_программы\nИсходные\_данные\_груза\_и\_расстояния\nФайл\_стоимость\_провоза" as Интерфейс\_программы

component [Стоимость\_груза\nРасчет] as Стоимость\_груза

component Характеристики\_паллат

component Стоимость\_провоза

component Список\_работ

Интерфейс\_программы -(0- [Стоимость\_груза]

Стоимость\_провоза..> Стоимость\_груза

Продолжительность\_работ ..> Стоимость\_груза

Характеристики\_паллат..> Стоимость\_груза

Диаграмма компонентов

@startuml

rectangle "Интерфейс\_программы\nФайл\_исходных\_данных\nФайл\_критических\_путей" as Интерфейс\_программы

component [Критические\_пути\nРасчет] as Критические\_пути

component Данные\_о\_предшествующей\_работе

component Продолжительность\_работ

component Список\_работ

Интерфейс\_программы -(0- [Критические\_пути]

Список\_работ ..> Критические\_пути

Продолжительность\_работ ..> Критические\_пути

Данные\_о\_предшествующей\_работе ..> Критические\_пути

@enduml

Диаграмма вариантов использования

@startuml

actor Пользователь

rectangle Расчет {

usecase "Закрыть\nприложение" as Закрыть\_приложение

usecase "Ввод\вывод\nданных" as Ввод\_данных

usecase "Открыть\nприложение" as Открыть\_приложение

usecase "Расчет\nрезультата" as Расчет\_результата

}

Пользователь - Ввод\_данных

Пользователь - Закрыть\_приложение

Закрыть\_приложение ..> Открыть\_приложение

Ввод\_данных ..> Открыть\_приложение

Расчет\_результата -> Ввод\_данных

@enduml

Диаграмма последовательности

@startuml

Пользователь -> Программа: Открыть программу

Программа -> БД: Ввести данные

БД --> Программа: Данные сохранены

Программа -> Программа: Расчет

Программа --> Пользователь: Вывод итоговых данных

@enduml

Диаграмма классов

@startuml

class Входные\_Данные {

Название файла

Количество работ

Список работ

Продолжительность работ

}

class Критические\_Пути {

Количество работ

Список работ

Продолжительность работ

Пути выполнения работ

Критические пути

}

class Выходные\_Данные {

Критические пути

Название текстового файла

}

class Юнит\_тесты {

Тестовый файл

Количество работ

Список работ

Продолжительность работ

}

Входные\_Данные --o Критические\_Пути

Критические\_Пути -o Выходные\_Данные

Юнит\_тесты --o Критические\_Пути

@enduml

Диаграмма состояний

@startuml

[\*] --> Входные\_данные

state Входные\_данные {

Массив\_данных: Количество работ\n Список работ\n Продолжительность работ

[\*] --> Файл\_данных

Файл\_данных --> Массив\_данных

}

state Критические\_пути {

Массив\_данных --> Поиск\_всех\_путей

Поиск\_всех\_путей --> Поиск\_критических\_путей

}

state Выходные\_Данные {

Поиск\_критических\_путей --> Файл\_выходных\_данных

}

Файл\_выходных\_данных --> [\*]

@enduml

@startuml

class Входные\_Данные\_Груза {

Длина

Ширина

Высота груза

Средняя плотность груза

}

class Данные\_Кузова {

Вес

Объем кузова

Длина кузова

Ширина кузова

Одна паллета, если не превысит грузоподъемность()

}

class Выходные\_Данные {

Стоимость поездки()

}

class Юнит\_тесты {

Тестовые данные груза

Общие данные груза

}

Входные\_Данные\_Груза --o Данные\_Кузова

Данные\_Кузова -o Выходные\_Данные

Юнит\_тесты --o Данные\_Кузова

@enduml