

RFT\_CS

*Rocket fuel and trajectory computing system*

Автор: Булат Насыров

30 декабря 2022 г.

# Оглавление

<b>1</b>	<b>Введение</b>	<b>4</b>
1.1	Концепция . . . . .	4
1.2	Цель . . . . .	4
1.3	Технологии . . . . .	4
1.4	Декомпозиция задачи . . . . .	5
1.5	Обработка ошибок . . . . .	5

# Предисловие

О чем будет идти речь, когда будем обсуждать ПО?

Так же как любая метафора, описание программного обеспечения с точки зрения архитектуры может что-то скрыть, а что-то, наоборот, проявить; может обещать больше, чем давать, и давать больше, чем обещать.

Основа привлекательности архитектуры - это структура. А структура - это то, что доминирует над парадигмами и суждениями в мире разработки ПО - компонентами, классами, функциями, модулями, слоями и службами, микро или макро. Но макроструктура многих программных систем часто пренебрегает убеждениями или пониманием - организация советских предприятий, невероятные небоскребы-башни (манареты) Дженга, достигающие облаков, археологические слои, залегающие в горной породе. Структура ПО не всегда интуитивно очевидна, как структура зданий.

Здания имеют очевидную физическую структуру, независимо от материала, из которого они построены, от их высоты или ширины, от их назначения и от наличия или отсутствия архитектурных украшений. Их структура мало чем отличается - в значительной мере она обусловлена законом тяготения и физическими свойствами материалов. ПО, напротив, никак не связано с тяжестью, кроме чувства серьезности. И из чего же сделано ПО? В отличие от зданий, которые могут быть построены из кирпича, бетона, дерева, стали и стекла, программное обеспечение строится из меньших программных компонентов, и т.д., вплоть до основания. Говоря об архитектуре, можно сказать, что программное обеспечение по своей природе является фрактальным и рекурсивным, выгравированным и очерченным в коде. Здесь важные все детали. Переплетение уровней детализации также вносит свой вклад в архитектуру, но бессмысленно говорить о ПО в физических масштабах. Программное обеспечение имеет структуру - множество структур и множество их видов, - но их разнообразие затмевает диапазон физических структур, которые можно увидеть на примере зданий. Можно даже довольно убедительно утверждать, что при проектировании ПО архитектуре уделяется куда больше внимания, чем при проектировании зданий, - в этом смысле архитектура ПО является более многообразной, чем архитектура зданий!

Но физический масштаб привычнее людям, и они часто ищут его в окружающем мире. Несмотря на привлекательность и визуальную очевидность, прямоугольники на диаграммах PowerPoint не являются архитектурой ПО. Да, они представляют определенный взгляд на архитектуру,

значит не получить ни общей картины, ни понятия об архитектуре: архитектура ПО ни на что не похожа. Конкретный способ визуализации - не более чем частный выбор. Этот выбор основан на следующем наборе вариантов: что включить; что исключить; что подчеркнуть формой или цветом; что, наоборот, затенить. Никакой взгляд не имеет никаких преимуществ перед другим.

Возможно, нет смысла говорить о законах физики и физических масштабах применительно к архитектуре ПО, но мы действительно учитываем некоторые физические ограничения. Скорость работы процессора и пропускная способность сети могут вынести суровый приговор производительности. Объем памяти и дискового пространства может ограничить амбиции любого программного кода. ПО можно сравнить с такой материей, как мечты, но ему приходится работать в реальном, физическом мире.

# Глава 1

## Введение

### 1.1 Концепция

RFT\_CS (Rocket fuel and trajectory computing system) Система расчета ракетного топлива и траектории полета ракеты - это Python-библиотека для разработки математических моделей. RFT\_CS изначально был спроектирован так, чтобы его можно было внедрять постепенно. Другими словами, **вы можете начать с малого и использовать только ту функциональность RFT\_CS, которая необходима вам в данный момент.** Также в случае, если вам нужно изменить поведение/вычисления функции, есть возможность конфигурации методов под ваши нужды.

### 1.2 Цель

Основная цель - создать математическую модель процессов, связанных с полётом одно и многоступенчатых, твердо и жидко топливных ракет и для вычисления траектории полёта баллистических ракет. Данное ПО может быть использовано для создания космических/баллистических ракет или своих научных экспериментов.

### 1.3 Технологии

Программное обеспечение построено на высокоуровневом языке программирования Python и отдельные микропроцессоры написаны на языке С. Также для сложных математических вычислений использовались библиотеки, специально созданные для этой цели.

## 1.4 Декомпозиция задачи

Начнём с составных частей ракеты, а также внешние факторы, влияющие на полёт. Начнём с состава космической ракеты:

## 1.5 Обработка ошибок

Могут возникать ошибки связанных с некорректным математических операций, перегрузкой или долгим ожиданием ответа от системы и неверным вводом/выводом данных, и др. Подобные ошибки обрабатываются и выводятся в виде ответа, а также записываются в логи. **"При добавлении новых функций или использование встроенных функций важно не забывать обрабатывать ошибки и добавлять логирование для дальнейшего удобства исправления багов."**