

Система 'ГРАФ'

В.Куликов, для НПП "ЭиС", 2020

Table of Contents

Главная

- [ГРАФ - Графический Редактор Алгоритмов и Функций, v.0.5 \(beta\)](#)
 - [Предварительные пояснения](#)
 - [Структура документации](#)

Руководство разработчика

1. Правила

- [Руководство разработчика](#)
 - [Вечные проблемы разработки](#)
 - [Основные правила](#)
 - [Виды документации](#)
 - [Цикл разработки](#)

2. Спецификация

- [Техническая часть](#)
 - [ОС разработки](#)
 - [Языки, фреймворки, IDE](#)
 - [Инструмент документирования](#)
 - [UML, схемы, иллюстрации](#)
 - [Временное размещение в веб](#)
 - [Хранение кода](#)
 - [Строка деплоя](#)

3. История разработки

- [История разработки](#)
 - [06/2020](#)
 - [25/06/2020](#)
 - [25/06/2020](#)
 - [PlantUML](#)
 - [Внедрение эффективных правил разработки "В одну каску"](#)
 - [Переписывание с TornadoFx на чистый JavaFx](#)
 - [05/2020](#)
 - [Проблема лицензирования Oracle](#)

Руководство пользователя

1. Кухня

- [Руководство пользователя](#)

ГРАФ - Графический Редактор Алгоритмов и Функций, v.0.5 (beta)

Описание и алгоритмы

Предварительные пояснения

Программа базируется на исходниках мощного графического проекта с открытой лицензией [Eclipse Public License 1.0](#).
Насколько я понял из описания этой лицензии вот [тут](#) и [тут](#), на ее основе можно писать коммерческие приложения.

Единственное ограничение в том, что в последующем следует оформить весь исходный базовый код в качестве библиотеки, чтобы избежать возможных коллизий и [требований открыть исходники](#).

Структура документации

- [Руководство разработчика](#)
- [Руководство пользователя](#)

Руководство разработчика

Руководство разработчика

Вечные проблемы разработки

- Отсутствие адекватной документации в коде и для пользователя
- Непонимание "чего тут понаписано" при смене разработчика
- Плохо протестированная функциональность

Следует придерживаться основных правил и шагов разработки

Основные правила

- Вся документация и код должны разрабатываться одновременно

- Код должен храниться в системе контроля версий
- Комментарии в коде
- Стараться тестировать всю логику (принцип TDD)

Виды документации

- Документация разработчика, включая технические вопросы сборки и запуска
- Руководство пользователя
- Комментарии в коде

Цикл разработки

1. Описание задачи в свободной форме (Тезисы, User Story)
2. Формализация в алгоритм
3. Тестовый код
4. Реализация + комментарии в коде
5. Проверка сборки и запуска
6. Анализ полученного, проверка соответствия задачи, документации и работы приложения
7. Итеративное редактирование (пп. 1-6) или переход к другой задаче



Далее: [спецификация](#)

Техническая часть

Приложение разрабатывается на Ubuntu, но регулярно тестируется на Win10 для проверки на кроссплатформенность.

ОС разработки

Ubuntu 20.04.

Языки, фреймворки, IDE

- Языки: Java, JavaFx, Kotlin
- Фреймворк: openjdk 11
- Тул сборки: POM

- IDE: IntelliJ IDEA Community Edition

Инструмент документирования

Документация пишется в MkDocs, генерится в 2 формата - статический сайт HTML-страниц и PDF.

Благодаря этому инструментарию и методам `mkdocs build && mkdocs serve` можно всегда автоматически получать самую актуальную версию документации

UML, схемы, иллюстрации

PlantUML...

Временное размещение в веб

Для проверки доступа по сети сайт документации временно размещается по адресу valeriusgc.github.io/.

Там же находится ссылка на актуальный PDF.

Хранение кода

Код хранится в приватном репозитории по адресу github.com/ValeriusGC/graph-editor

Строка деплоя

При разработке следует регулярно запускать сборку, создавая нужную среду окружения. Я делаю вот так:

```
export PATH="$PATH:/usr/lib/jvm/java-11-openjdk-amd64/bin/java" \
&& export JAVA_HOME="/usr/lib/jvm/java-11-openjdk-amd64" \
&& ./build.sh \
&& ./start.sh
```

История разработки

Ниже основные моменты разработки в обратной хронологической последовательности.

06/2020

25/06/2020

Иллюстрация разработки "как у нас" и "Как принято" (см. *life-flow*)

25/06/2020

Окончательная систематизация методологии разработки "alone-dev";

PlantUML

Внедрение PlantUML

Внедрение эффективных правил разработки "В одну каску"

В одиночку очень сложно держать в голове все изменения и сложность алгоритмов и кода. В силу этого буду придерживаться продвижения по всем фронтам одновременно. Тогда документация, алгоритмы, иллюстрации, тесты всегда будет соответствовать коду, и на любом этапе можно быть уверенным, что не надо возвращаться назад и мучительно соединять разрозненные части.

Переписывание с TornadoFx на чистый JavaFx

Так как TornadoFx перестал развиваться, несмотря на его очевидные преимущества, использованные в **СДКУ-РК**, от него приходится отказываться. Уже много было написано, все это надо переводить на чистый JavaFx.

05/2020

Проблема лицензирования Oracle

Переход на OpenJDK.

Руководство пользователя

Руководство пользователя

[Меню "Вид"](#)