

# 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](#)

#### 1. Спецификация

- [Техническая часть](#)
  - [ОС разработки](#)
  - [Языки, фреймворки, IDE](#)
  - [Инструмент документирования](#)
  - [Строка деплоя](#)

### 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 по (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 - HTML и PDF.

### Строка деплоя

```
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
```

## Руководство пользователя

### Руководство пользователя

Руководство пользователя

Руководство пользователя

Руководство пользователя

Руководство пользователя

Руководство пользователя

Руководство пользователя

Руководство пользователя

Руководство пользователя

Руководство пользователя

Руководство пользователя

Руководство пользователя

Руководство пользователя

Руководство пользователя

Руководство пользователя

Руководство пользователя

Руководство пользователя