

# Требования к CLI для курсовой работы весеннего семестра

Для возможности проведения защиты курсовой работы весеннего необходимо пройти предварительную проверку на moodle: [ссылка на модуль](#)

Базовые сведения про cli можно прочитать здесь: [Требования к курсовым \(весенний семестр\)](#)

Если сборка курсовой работы реализована через системы автоматизации, то программа с курсовой работы должна называться **cw!**

## Обязательные флаги

- `-input, -i`: Задаёт имя входного изображения. При необходимости (если требуется по заданию), флаг может принимать несколько аргументов. Если флаг отсутствует, то предполагается, что имя входного изображения передаётся последним аргументом
- `-output, -o`: Задаёт имя выходного изображения. При необходимости (если требуется по заданию), флаг может принимать несколько аргументов
- `-info`: Печатает информацию об изображении и завершает работу
- `-help, -h`: Выводит справку. **ВАЖНО**: первой строкой при запуске программы нужно выводить информацию о варианте курсовой работе и об авторе программы в строго определенном формате: `Course work for option <V>, created by <Name> <Surname>`. Где V – вариант курсовой и Имя и Фамилия, как указано в репозитории группы. Данное предложение должно быть строго первым предложением в выводе программы и является отдельной строкой (заканчивается знаком `'\n'`). Например: `Course work for option 3.2, created by Ivan Ivanov.`

## Обработка ошибок

В случае ошибки и невозможности выполнить функцию по какой-либо причине, нужно вывести строку с ошибкой и завершить работу программы с кодом возврата из диапазона: **[40-49]**

Обработка ошибок при некорректных аргументах у флагов:

- Если флаг отвечает за цвет и передано некорректное значение (например `500.0.10`), то необходимо **остановить** выполнение и завершить работу с ошибкой
- Если флаг отвечает за координаты для рисования (например, для рисования линии) и координаты находятся вне границ изображения, то необходимо **продолжить обработку** так, как если бы можно было выходить за рамки изображения. Пример: допустим на вход подано изображение размером `100×100` пикселей, начало отрезка `0.0`, а конец отрезка `200.200`. В таком случае должна быть нарисована диагональ. Если же начало отрезка было `150.130`, то на изображении ничего не должно быть нарисовано, так как линия проходить за границами обрабатываемого изображения
- Если флаг отвечает за координаты для области и координаты находятся вне границ изображения, то необходимо **продолжить обработку** до максимально/минимально

возможного значения координат

- Если были введены аргументы для флага, который не принимает аргументов или введены лишние аргументы, то
  - **Завершить работу** с ошибкой
  - **Вывести пользователю сообщение**, что данные аргументы будут проигнорированы. Например, Аргументы 'best\_arg' и 'another\_best\_arg' проигнорирован, так как флаг '-info' не принимает аргументов
- Если введён аргумент, который не соответствует ожидаемому шаблону, то необходимо **завершить работу** с ошибкой
- Если введён некорректный флаг или не хватает флагов для задания параметров, то необходимо **завершить работу** с ошибкой

## Необязательные дополнения к курсовой работе

Описанные в данной секции дополнения являются **не обязательными**, получить желаемую оценку можно без них.

### GUI

Рядовые пользователи очень плохо умеют пользоваться терминалом, поэтому для их удобства зачастую реализуют графический интерфейс. В программе по обработке изображений такое напрашивается само собой (иначе убийца photoshop'a будет выглядеть не солидно). Подробнее про требования к GUI: [Требования к курсовым \(весенний семестр\)](#)

### Документация

В любом хорошем проекте должна быть документация. Документация означает, что для каждой функции, структуры (и классов для C++) есть адекватное описание. Фразы в духе “Данная структура содержит данные для изображения” или “Функция обработки изображения” являются **плохим описанием**.

- Для функций дополнительно расписаны аргументы (описание, назначение, возможные значения)
- Для структур (и классов в C++) дополнительно расписаны все поля

Документацию можно написать сразу в коде, обратите внимание на [Doxygen](#)

### Создание .so библиотеки

Зачастую код пишется не в формате отдельной программы, а в формате библиотеки, чтобы вашими наработками могли воспользоваться в другом проекте другие разработчики. Поэтому можно разделить курсовую на две части:

1. Библиотека формата .so , которая содержит только функции для работы с

изображениями: открытие и сохранение изображения, обработка изображений, ...

2. Главная часть, которая содержит в себе только main с CLI (и опционально GUI) обработкой и вызовом функций из разработанной библиотеки

**ВАЖНО:** Не нужно по умолчанию выполнять установку библиотеки в систему, сделайте для этого отдельную цель, например `make install`

From:

<http://se.moevm.info/> - **se.moevm.info**

Permanent link:

[http://se.moevm.info/doku.php/courses:programming:pr\\_cw\\_spring\\_requirements?rev=1711562773](http://se.moevm.info/doku.php/courses:programming:pr_cw_spring_requirements?rev=1711562773)



Last update: **2024/03/27 18:06**