1. МПО
2. Диаграмма коммуникаций
3. Диаграмма анализа
4. МПО

**Изображение** – содержит два общедоступных атрибута целочисленного типа: высота и ширина. Используется другими сущностями, исходящих связей не имеет.

**Сессия** – текущая сессия, которая содержит некоторое изображение и историю его изменений.

**История изменений** – совокупность команд, использовавшихся для изменения текущего изображения. Также используется другими сущностями.

**Редактор** – отвечает за взаимодействие пользователя с выбранным изображением. Содержит в себе набор инструментов и коллекцию сессий. Также хранит в себе само изображение и историю его изменений.

**Команда** – применяет команду по изменению изображения к самому изображению.

**Инструмент** – создает команду для изменения изображения в соответствии с выбранным инструментом и записывает её в историю изменений.

**УстановщикТекста**, **Кисть**, **Трансформатор**, **Цветокорректор**, **Фильтр** – наследуются от Инструмента и содержат свой функционал по изменению изображения.

1. Диаграмма коммуникаций

На диаграмме показан пример изменения яркости изображения. Вне диаграммы пользователь при помощи UI выбирает инструмент и применяет его к изображению. Тогда, Editor посылает сообщение **ApplyToImage(image: Image)** выбранному инструменту **BrightnessEditor**, передавая аргументом изображение. **BrightnessEditor** создает команду изменения яркости **ChangeBrightnessCommand(image: Image, oldBrightness:float, newBrightness:float)** передавая аргументом изображение, яркость которая была у изображения и яркость на которую нужно поменять. После создания такой команды, **BrightnessEditor** отправляет сообщение **CommandHistory.ExecuteCommand(command),** передавая аргументам созданную команду. **CommandHistory** отправляет сообщение команде **command.Execute(),** которая в свою очередь применяет изменение яркости к изображению отправляя **image:Image** сообщение **SetBrightness(newBrightness),** при том, перед отправкой производится проверка на то, что **Image != null.** После применения команды, управление возвращается к **CommandHistory** и команда добавляется в коллекцию **AddCommand(command)**.

1. Диаграмма анализа

Первым действием пользователь изменяет яркость изображения в окне граничащего класса **Editor**(1).   
Граничащий класс посылает сообщение в управляющий объект **BrightnessEditor** для обработки изображения(2).

Далее управляющий объект запрашивает данные изображения в граничащий класс **InputWindow**(3).

Затем граничащий класс демонстрирует пользователю окно ввода(4).

Через это самое окно пользователь устанавливает данные о яркости(brightness) в **InputWindow**(5).

Граничащий класс отправляет контроллеру **BrightnessEditor** введенные данные (6)

Контроллер создаёт команду с информацией о яркости и изображении и отправляет их в контроллер **ComandHistory**(7).

Контроллер отправляет изменение яркости и обновляет сущность **Image(8).**

Уже обновленная сущность **Image** выводится ползователю(9)