**2. СТРУКТУРНОЕ ПРОЕКТИРОВАНИЕ**

**2.1 Описание модулей**

На начальном этапе разработки проекта удобно представить его в виде абстрактных модулей, чье взаимодействие друг с другом и определяет работу программы. Для этого была составлена структурная схема программы, приведенная в ПРИЛОЖЕНИИ Б.

Программа была разбита на 4 модуля: графический интерфейс, управляющий модуль, исполняющий модуль и модуль работы с файловой системой.

Каждый модуль реализуется классом или группой классов, которые представлены на диаграмме классов (см. ПРИЛОЖЕНИЕ А). Ниже приведено краткое описание каждого из модулей.

Модуль графического интерфейса полностью реализуется классом MainWindow, унаследованного от класса QMainWindow. Через этот модуль идет основное взаимодействие программы с пользователем.

Через графический интерфейс пользователь взаимодействует с управляющим модулем, где, по сути, и ведется вся основная работа программы. Этот модуль реализуется классом DrawingArea, наследником QWidget.

Модуль работы с файловой системой реализован классами MainWindowиDrawingArea. При помощи метода класса MainWindow мы получаем путь к файлу, после чего метод класса DrawingArea производит загрузку файла. Если же требуется сохранить изображение, то эта операция реализуется методом класса DrawingArea.

Исполняющий модуль представлен сразу несколькими классами : DrawCommand, RotateCommand, FlipCommand, ResizeCommand, FiltratingCommand, CorrectionCommand, наследниками QUndoCommand. Этот модуль реализует основной функционал программы.

Когда пользователь совершает какое-либо действие в графическом интерфейсе, оно интерпретируется управляющим модулем, как команда, которую необходимо выполнить. В зависимости от поступившей команды начинает действовать либо исполняющий модуль, либо модуль работы с файловой системой. Если срабатывает модуль работы с файловой системой, то он просто открывает или сохраняет файл, и на этом его функции окончены. Если срабатывает исполняющий модуль, то он выполняет отданную команду, после чего модуль графического интерфейса обновляется, отображая результат проведенной операции.

**2.2 Реализация связей модулей**

Рассмотрим взаимосвязь модуля работы с файловой системой с управляющим модулем. Можно сказать, что оба модуля реализуются одним классом DrawingArea. Когда пользователь хочет открыть файл, срабатывает метод класса MainWindow,

void open();

который получает строку, содержащую путь к нужному файлу. Эта строка передается в метод класса DrawingArea

bool openImage(const QString &filename);

который в свою очередь загружает выбранный файл.

Сохранение файла происходит по похожей схеме: сначала срабатывает метод класса MainWindow,

void save();

который получает формат, в котором будет сохранен файл. Затем срабатывает метод этого же класса,

bool saveFile(const QByteArray &fileFormat);

который получает строку имени нового файла, включая путь его сохранения ,после чего вызывается метод класса DrawingArea,

bool saveImage(const QString &filename, const char\* fileformat);

сохраняющий файл по указанному пути.

Это можно интерпретировать так: управляющий модуль получает от пользователя команду открыть или сохранить файл при помощи вышеупомянутых методов класса MainWindow, после чего он отдает команду открытия или сохранения модулю работы с файловой системой, которая реализуется вышеупомянутыми методами класса DrawingArea.

Модули графического интерфейса и управления связаны следующими методами класса MainWindow:

void penColor();

void penWidth();

void flipHorizontal();

void flipVertical();

void rotateLeft();

void rotateRight();

void changeSize();

void filtrate();

void correction();

эти методы отвечают соответственно за подачи команд: изменения цвета, толщины ручки, отражения по горизонтали, вертикали, поворота налево, направо, изменения размера изображения, его фильтрации, коррекции гаммы.

Модули графического интерфейса и исполнения связаны стандартным методом класса QWidget,

void update();

который вызывается в конце каждой команды, тем самым обеспечивая постоянное обновление виджета рабочей зоны, с которой взаимодействует пользователь. При обновлении виджета выводится и новая информация.

Модули управления и исполнения связаны следующими методами класса DrawingArea:

void setPenColor(const QColor &newColor);

void setPenWidth(int newWidth);

void resizeIamge(const QSize newSize);

void flip(bool horizontal, bool vertical);

void rotate(qreal deg);

void medianFilter(const int ratio);

void gammaCorrection(const double gamma);

эти методы отвечают соответственно за исполнение команд: изменения цвета, толщины ручки, изменения размера изображения, отражения по горизонтали или вертикали, поворота налево или направо, фильтрации изображения, коррекции его гаммы.