**3. ФУНКЦИОНАЛЬНОЕ ПРОЕКТИРОВАНИЕ**

**3.1 Класс DrawingArea**

Класс DrawingArea – это основной класс программы,, который представляет собой виджет рабочей зоны. Именно на нем происходит все рисование а также редактирование: Его основными элементами являются:

1. QUndoStack \_undoStack – стек команд, реализующих DrawingArea
2. Shape\* \_currentShape – указатель на фигуру.
3. Shape\*(\*\_createShape) (QImage\*, const QPoint&, int, const QColor&) – указатель на функцию создания формы рисования.
4. DrawingArea(QUndoStack \*undoStack, QWidget \*parent = 0)– конструктор класса.
5. void setCreateEllipse()– слот, задающий создание эллипса.
6. void setCreateRectangle()– слот, задающий создание прямоугольника.
7. void setCreateLine()– слот, задающий создание линии.
8. void setCreateCurve()– слот, задающий создание кривой.
9. void setCreateEraser()– слот, задающий создание ластика.
10. void setCreateFilledShape()– слот, задающий создание заливки.
11. void *mousePressEvent*(QMouseEvent \*event) – перегруженное события нажатия клавиши мышки, которое отвечает за начало рисования
12. void *mouseReleaseEvent*(QMouseEvent \*event) – перегруженное события движения мыши, которое отвечает за изменение фигур при рисовании
13. void mouseMoveEvent(QMouseEvent \*event) – перегруженное событие отпускания кнопки мышки, которое создает объект класса DrawCommand и заносит его в \_undoStack.
14. void medianFilter(const int ratio) – метод, который создает объект класса FiltratingCommand и добавляет его в undoStack.
15. void gammaCorrection(const double gamma) – метод, который создает объект класса CorrectionCommand и добавляет его в undoStack.
16. void resizeIamge(const QSize newSize) – метод, который создает объект класса ResizeCommand и добавляет его в undoStack.

Таким образом классы команд реализуют класс DrawingArea, чьи методы завязаны на них. Сам же класс DrawingArea управляет этими командами, т.е. вызывает их в случае необходимости, совершает откат или повторение.

**3.2 Класс MainWindow**

Класс MainWindow– это класс, представляющий собой графический интерфейс программы: именно через него идет большая часть взаимодействия программы с пользователем. Данный класс реализует все кнопки приложения, панель инструментов и меню. Основными методами MainWindow являются:

1. MainWindow(QWidget \*parent = 0) – конструктор класса
2. void filtrate() – слот, являющийся графической оберткой метода void medianFilter(const int ratio) класса DrawingArea для взаимодействия с пользователем.
3. void correction() – слот, являющийся графической оберткой метода void gammaCorrection(const double gamma) класса DrawingArea для взаимодействия с пользователем.
4. void changeSize() – слот, являющийся графической оберткой метода void resizeIamge(const QSize newSize) класса DrawingArea для взаимодействия с пользователем.
5. void create DrawActiongroup()– метод создания действий, связанных с рисованием. Кнопки этих действий подключаются к соответствующим слотом класса DrawingArea.
6. void createActions()– метод создания действий.
7. QToolBar\* createToolBar()– метод создания панели инструментов.
8. void createMenues() – метод создания меню.

Таким образом, класс MainWindow, при взаимодействии с пользователем лишь перенаправляет все команды в «управляющий модуль», представленный классом DrawingArea.

**3.3 Класс Shape**

Класс Shape– это абстрактный класс, который в контексте данной программы используется для хранения объектов производных классов. Использование полиморфизма позволяет значительно упростить процесс рисования и сделать его шаблонным для любых форм. Основными методами Shapeявляются:

1. Shape(QImage\* image, int penWidth, const QColor &penColor) – защищенный конструктор, который нужен производным классам для создания родительского объекта, хранящего общие для всех фигур атрибуты.
2. virtual void *draw*(QPainter &painter) = 0 – чисто виртуальный метод, отвечающий за нанесение фигур на изображение. Переопределяется в производных классах фигур.
3. virtual void *update*(const QPoint &toPoint) = 0 – чисто виртуальный метод, отвечающий за то, чтобы размер прямоугольника фигуры и его содержимое были актуальны. Переопределяется в производных классах фигур.
4. virtual QRect *rect*() = 0 – чисто виртуальный метод, отвечающий за нахождение прямоугольника, содержащего фигуру. Переопределяется в производных классах фигур.

От класса Shape наследуется множество фигур, однако они лишь являются частными случаями и узкими специализациями широког функционала рисования этого класса.