Министерство науки и высшего образования РФ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Ярославский государственный технический университет»

Кафедра «Информационные системы и технологии»

Курсовой проект защищен

с оценкой \_\_\_\_\_\_\_\_\_\_\_\_\_

Преподаватель

\_\_\_\_\_\_\_\_\_\_ А.Н. Прозоров

«\_\_\_» \_\_\_\_\_\_\_\_\_\_ 2025

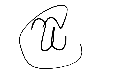
**РАзработка ГРАФИЧЕСКОГО РЕДАКТОРА**

расчетно-пояснительная записка к курсовому проекту по дисциплине

«Технологии программирования»

ЯГТУ 09.03.02 – 007 КП

Работу выполнил

студент группы ЗЦИС-26

\_\_\_\_\_\_\_\_\_\_ И.С. Рубцов

« » 2025

2025

Министерство науки и высшего образования РФ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Ярославский государственный технический университет»

Кафедра «Информационные системы и технологии»

ЗАДАНИЕ №7

ПО КУРСОВОМУ ПРОЕКТИРОВАНИЮ

Студенту Рубцову И.С. факультет заочный курс 2 группа ЗЦИС-26

1. Тема проекта и исходные данные Разработка приложения графический редактор в С# с использованием Windows Forms MDI.
2. Представить следующие материалы
   1. текстовые а)\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ б)\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ в)\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ г)\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ д)\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ е)\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
   2. графические а)\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ б)\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ в)\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
3. Рекомендуемая литература и материалы
   1. Троелсен и Джепикс. Язык программирования C#
   2. Скит. C# для профессионалов. Тонкости программирования
   3. Документация по C# - https://docs.microsoft.com/ru-ru/dotnet/csharp/
4. Дата выдачи задания\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
5. Срок сдачи законченного проекта\_\_\_\_\_\_\_\_\_\_
6. Отметка о явке на консультацию:

1)\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2)\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

3)\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 4)\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

5)\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 6)\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |  |
| --- | --- |
| Руководитель проекта | Заведуюший кафедрой |
| Язев В.А. | Бойков С.Ю. |
|  |  |

Оглавление

[Введение 4](#_Toc194858937)

[Описание задания 4](#_Toc194858938)

[Подробное описание используемых классов и алгоритмов 6](#_Toc194858939)

[Базовые классы 6](#_Toc194858940)

[Используемые алгоритмы 6](#_Toc194858941)

[Инструкция пользователя по работе с программой 8](#_Toc194858942)

[Инсталляция 8](#_Toc194858943)

[Запуск 8](#_Toc194858944)

[Работа с родительской формой 8](#_Toc194858945)

[Работа в редакторе 10](#_Toc194858946)

[Общий вывод 16](#_Toc194858947)

Введение

Современные цифровые технологии требуют от программного обеспечения высокой эффективности, особенно когда речь идет о графических инструментах. Редакторы изображений играют ключевую роль в цифровом искусстве, веб-дизайне, инженерной графике и других областях. Создание подобного приложения предполагает не только знание основ обработки графики, но и умение применять актуальные методы программирования.

Использование языка C# и платформы .NET открывает широкие возможности для разработки графических программ, включая инструменты для работы с растровыми и векторными изображениями, создание интерфейса и управление событиями. В рамках данного проекта ставится цель разработать графический редактор с базовым функционалом: рисование, редактирование и сохранение изображений.

Актуальность работы объясняется постоянным спросом на графические редакторы в разных сферах, а также возможностью углубить знания в программировании на C#. В процессе реализации проекта будут изучены принципы обработки графики в .NET, управление событиями и способы повышения производительности приложения.

Основная задача проекта - создать удобный и функциональный редактор, поддерживающий стандартные операции с изображениями. В ходе разработки предстоит:

Внедрить основные инструменты рисования (кисть, карандаш, ластик, геометрические фигуры).

Реализовать загрузку и сохранение файлов.

Оптимизировать работу программы и провести тестирование.

Выполнение курсового проекта позволит улучшить навыки в объектно-ориентированном программировании, работе с графикой и проектировании интерфейсов. Кроме того, готовое приложение может послужить базой для дальнейшего расширения его возможностей.

Описание задания

Создать в С# с использованием Windows Forms MDI приложение - графический редактор, позволяющий создавать и редактировать графические файлы с помощью рисования мышью простейших фигур в контурном и заполняемом вариантах, пером, с возможностью операций вырезания, копирования и вставки фрагментов.

Подробное описание используемых классов и алгоритмов

Базовые классы

Все элементы расположены на формах и унаследованы от класса Form. С помощью базовых контролов (классы: RadioButton, TrackBar, Label, Button, ToolStripMenu, PictureBox, StatusStrip) пользователь может взаимодействовать с программой. Для работы с базовыми средствами операционной системы, а конкретно диалогами сохранения и открытия файла используются классы (OpenFileDialog, SaveFileDialog).

Для работы с изображением используется класс Bitmap, для хранения основного и временного изображений, Color – для работы с цветом, Rectangle – для рисования фигур и служебных операций, Pen, Graphics и Brush – для рисования фигур.

Для хранения служебной информации используются Dictionary и Stack. Для быстрой обработки изображения используется класс Marshall и BitmapData. Для работы с буфером обмена используется класс Clipboard.

Для работы с выделенным фрагментом изображения разработан собственный класс DraggedFragment.

Используемые алгоритмы

Выбор цвета (основной, дополнительный) реализуется через системный диалог выбора цвета, выбранный цвет в диалоге устанавливается как фоновый цвет кнопки.

Выбор цвета для инструмента реализован внутри события MouseMove объекта PicureBox. Цвет выбирается определением зажатой клавиши мыши. При нажатии левой клавиши инструменту присваивается основной цвет, при нажатии правой – дополнительный.

Рисование мышью (ластик, карандаш, кисть) реализуется через 3 события мыши на PictureBox:

1. MouseDown – при зажатой клавише мыши определяется выбранный инструмент, проставляется флаг рисования: CanPaint.

2. MouseMove – при движении указателя, если установлен флаг CanPaint, то на позиции указателя мыши рисуется линия.

3. MouseUp – при отпускании клавиши мыши флаг CanPaint сбрасывается, чтобы движение по холсту не вызывало рисование.

Рисование фигур (эллипс, параллелограмм, линия) производится через наложение временного Bitmap на область рисования. Также как и для рисования мышью производится выбор фигуры, цвета через радиокнопки. При зажатии кнопки противоположный край располагается по курсору мыши и при отпускании кнопки мыши, временное изображение, на котором отображалась фигура, применяется к полотну. Также обрабатывается нажатие клавиши CTRL для пропорционального изменения фигуры, размер высчитывается по длине противоположной стороны.

Изменение размера холста реализовано через сохранение текущего изображения в Bitmap и применении Bitmap новому холсту. Размер холста меняется при выборе соответствующего инструмента и зажатии правого нижнего угла области рисования.

Выделение области реализовано по принципу рисования прямоугольника, рисуется область выделения и если курсор внутри области выделения, то при зажатии кнопки мыши он перетаскивается.

Пипетка реализована по событию MouseDown, цвет извлекается из пикселя с помощью метода GetPixel в который передается координата курсора события MouseUp.

Изменение толщины инструмента реализовано с помощью контрола TrackBar. При изменении значения новое значение записывается в поле ToolWidth, которое используется при инициализации инструментов.

Заливка реализована через сравнение соседних пикселей относительно цвета выбранного для заливки фрагмента, если цвет совпадает, то координата вносится в массив пикселей для изменения. Далее замена пикселей производится напрямую в памяти через метод Marshall.Copy.

История изменения реализована через сохранение изображения в Stack перед применением инструмента. При откате из истории извлекаем последнее состояние холста и удаляем из Stack.

Операции над выделенным фрагментом (удаление, копирование, вырезание, вставка) реализованы с использованием класса Clipboard, который позволяет работать с системным буфером обмена.

Открытие и сохранение изображения реализованы с помощью выбора пути к изображению (SaveFileDialog и OpenFileDialog).

Для родительской формы (MDI) был выставлено свойство IsMdiContainer, что позволяет открывать дочерние окна внутри родительского MDI контейнера.

Открытие изображения из родительской формы сразу в дочерней выполнено через дополнительный конструктор дочерней формы, в который передается новое изображение.

Инструкция пользователя по работе с программой

Инсталляция

Программа не требует установки.

Запуск

Для запуска дважды кликните на CourseProject.exe.

Работа с родительской формой

При запуске всегда открывается главная родительская форма, представленная на рисунке 1.

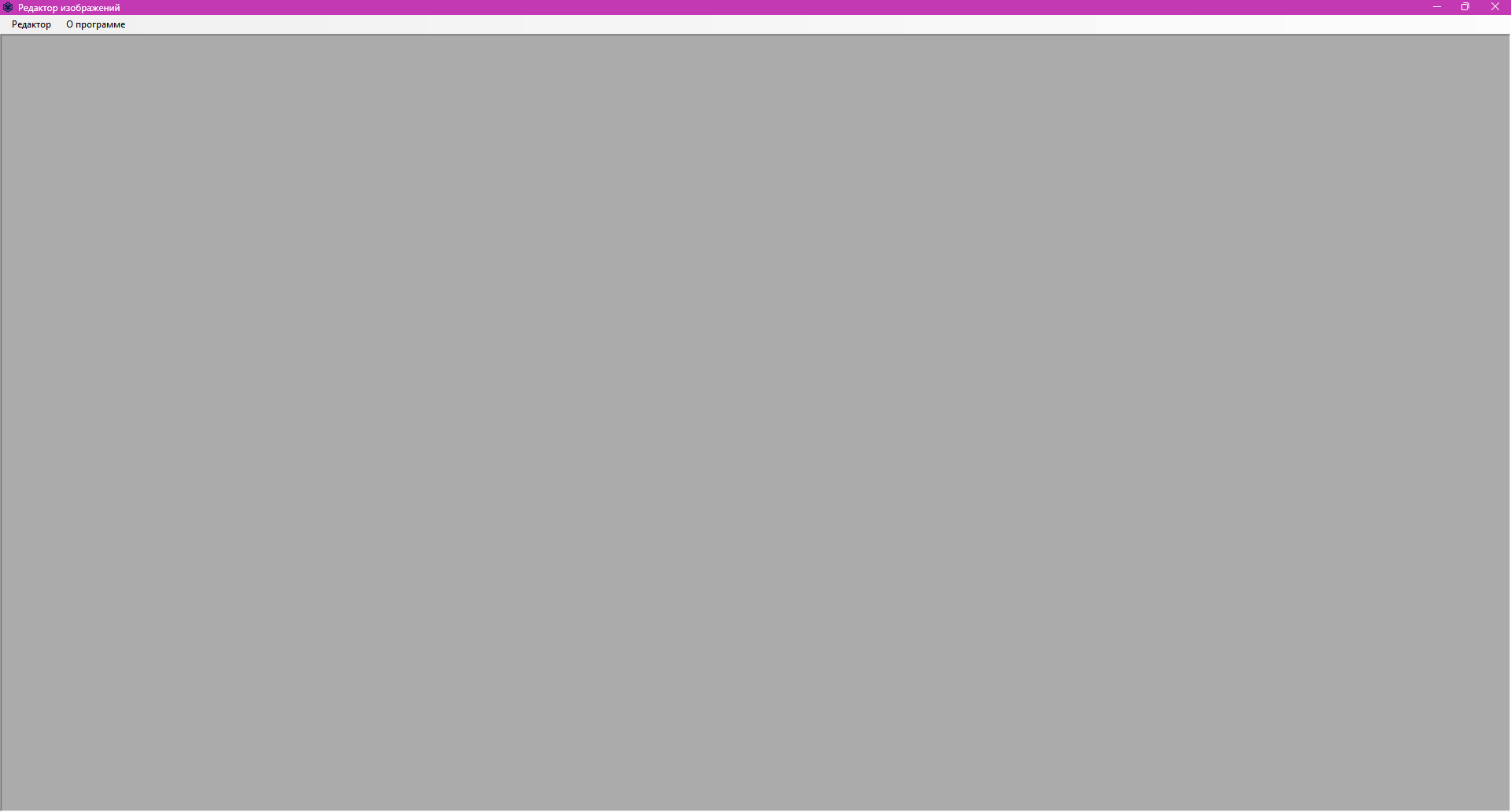


Рисунок 1. Главная форма

Чтобы открыть редактор с чистым холстом перейдите Редактор -> Пустой редактор, инструкция представлена на рисунке 2.

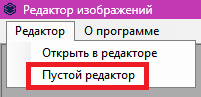


Рисунок 2. Пустой редактор

Далее требуется выбрать цвет холста, инструкция представлена на рисунке 3.

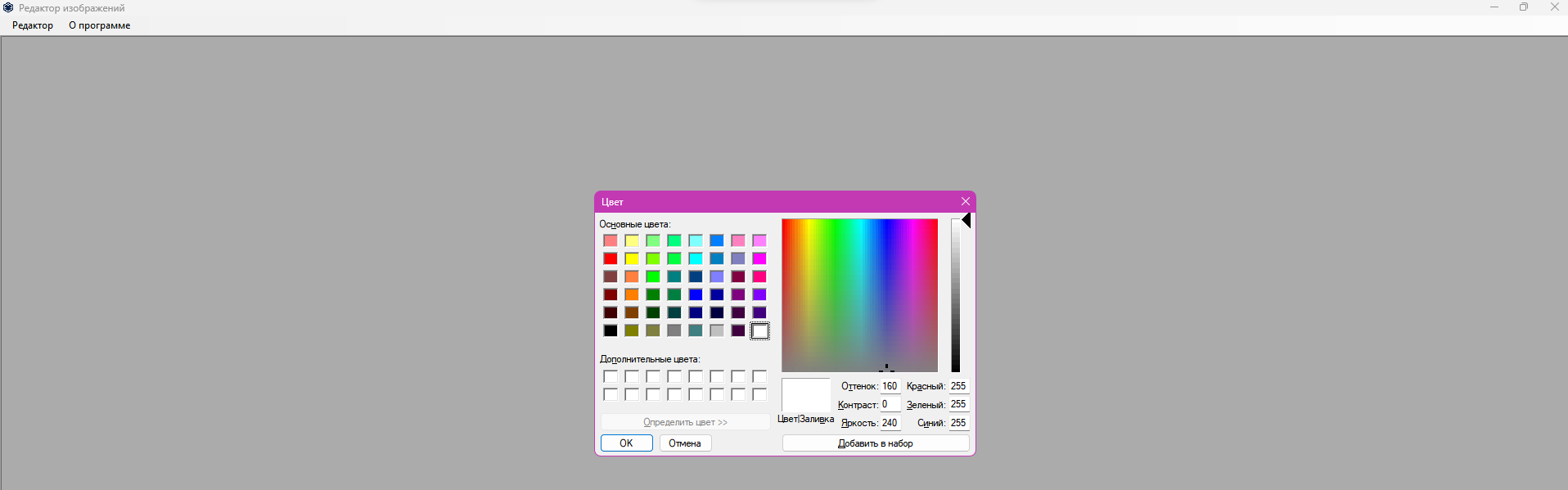


Рисунок 3. Выбор цвета холста

Чтобы открыть изображение в редакторе, перейдите Редактор -> Открыть в редакторе инструкция представлена на рисунке 4.

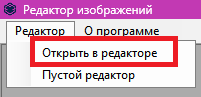


Рисунок 4. Открыть в редакторе

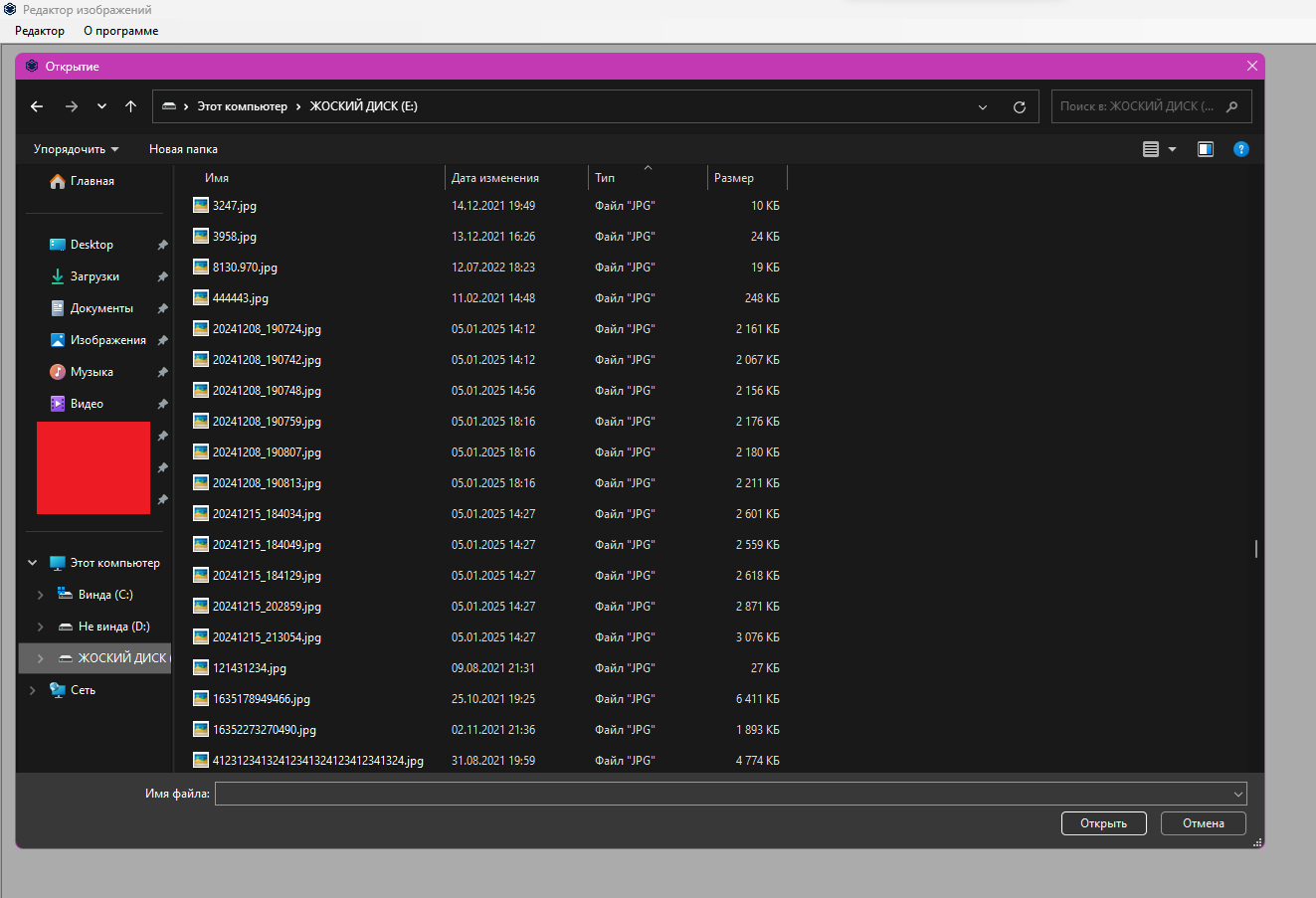


Рисунок 5. Выбор изображения для открытия в редакторе

В зависимости от ранее выбранной опции будет отображаться картинка, либо пустой холст в окне редактора изображения.

Работа в редакторе

Редактор предоставляет следующий инструментарий:

1. Карандаш,
2. Кисть,
3. Ластик,
4. Пипетка,
5. Заливка,
6. Линия,
7. Эллипс,
8. Квадрат,
9. Холст,
10. Выделение
11. Цвет,
12. Толщина,
13. Отмена последнего действия,
14. Открытие файла,
15. Сохранение файла.

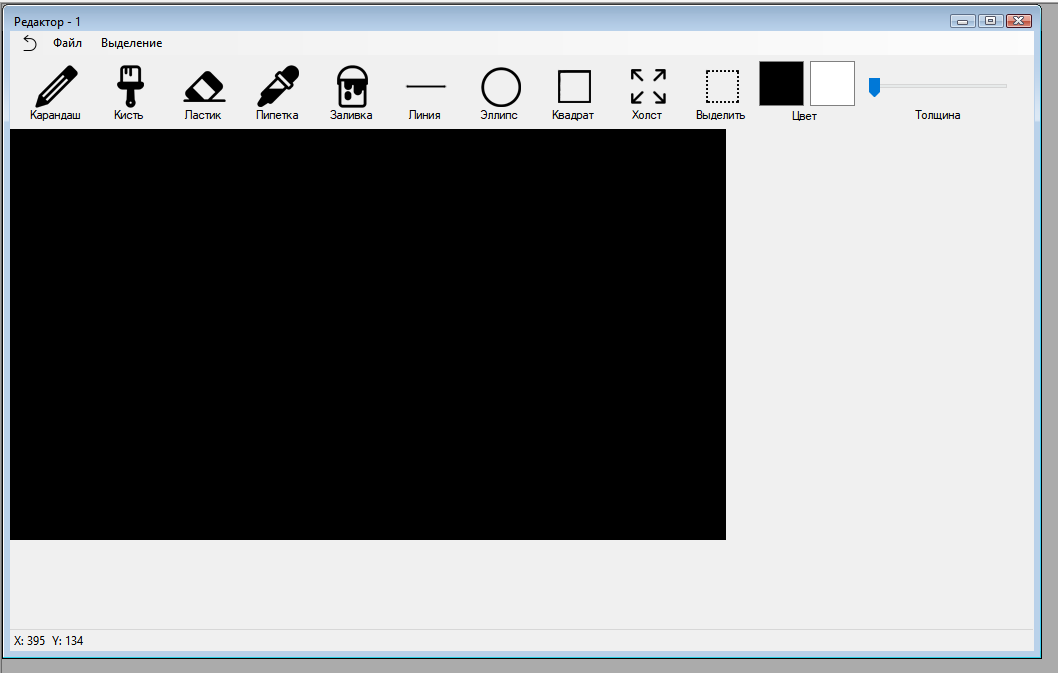


Рисунок 6. Окно редактора

Для работы в редакторе выберите инструмент из меню инструментов, представленных на рисунке 7 и выполняйте стандартные действия.

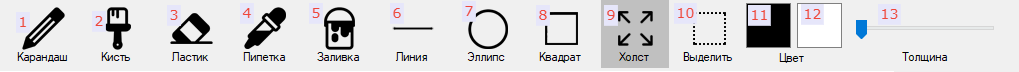


Рисунок 7. Инструменты

Рисование производится зажатием левой или правой клавиши мыши на холсте.

Перед применением инструмента установите основной и дополнительный цвета, которые изображены на рисунке 7 номер 11 и рисунке 7 номер 12, также установите толщину инструмента, ползунок представлен на рисунке 7 номер 13.

Для рисования основным цветом, зажимайте левую кнопку мыши. Для рисования дополнительным цветом используйте правую кнопку мыши.

В режиме свободного рисования различных кривых линий используйте инструменты «Карандаш» или «Кисть», которые изображены на рисунке 7 номер 1 или рисунке 7 номер 2.

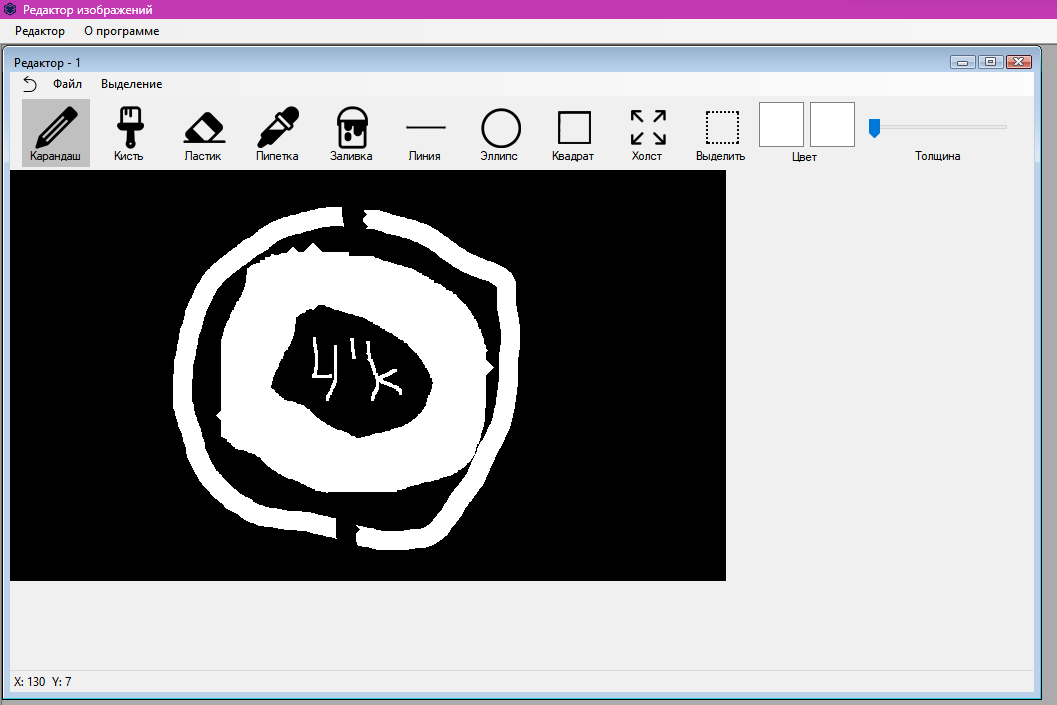


Рисунок 8. Рисование карандашом и кистью

Для исправления неточностей в изображении используйте инструмент «Ластик». Ластик затирает область цветом холста, если открывалось изображение, то цвет холста считается белым.

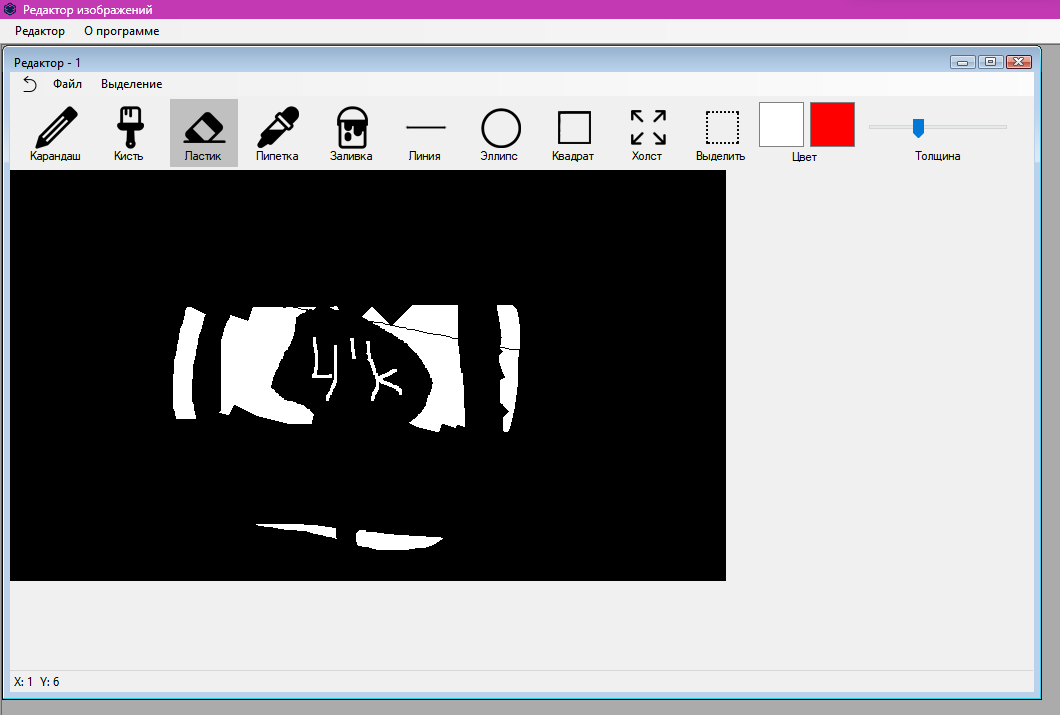


Рисунок 9. Рисование ластиком

Для определения цвета пикселя используйте инструмент «Пипетка». Цвет пикселя установится в основной или дополнительный цвет в зависимости от нажатой клавиши мыши.

Для заливки зоны изображения используйте инструмент «Заливка». Цвет заливки зависит от нажатой клавиши мыши.

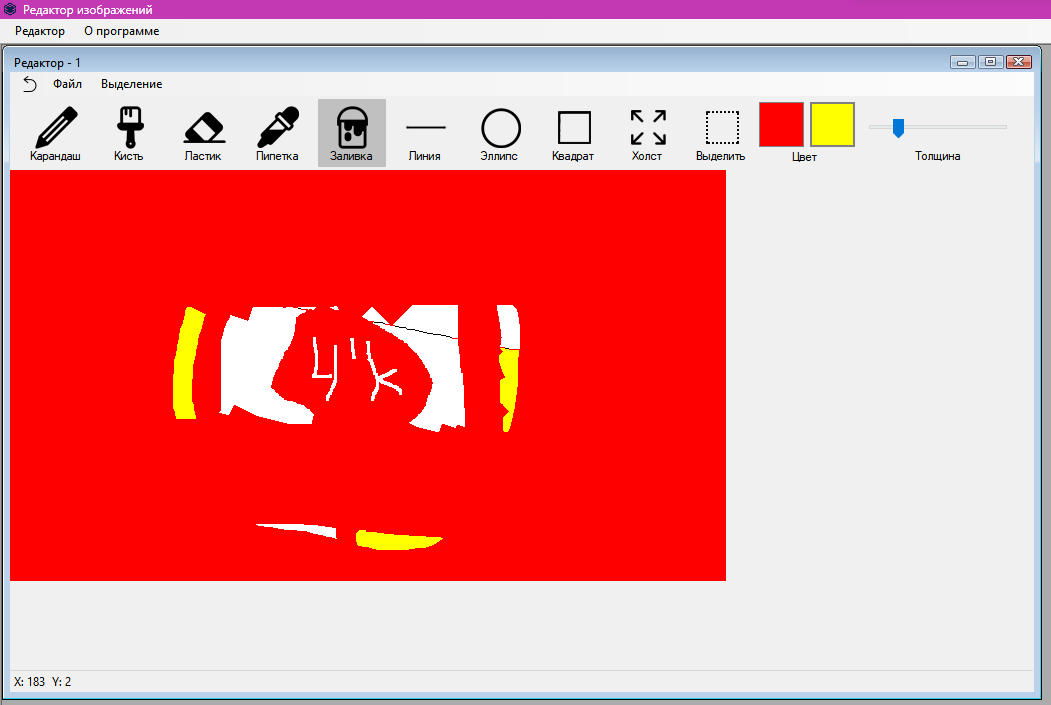


Рисунок 10. Заливка изображения

Для рисования прямой линии используйте инструмент «Линия». Цвет линии зависит от зажатой клавиши мыши.

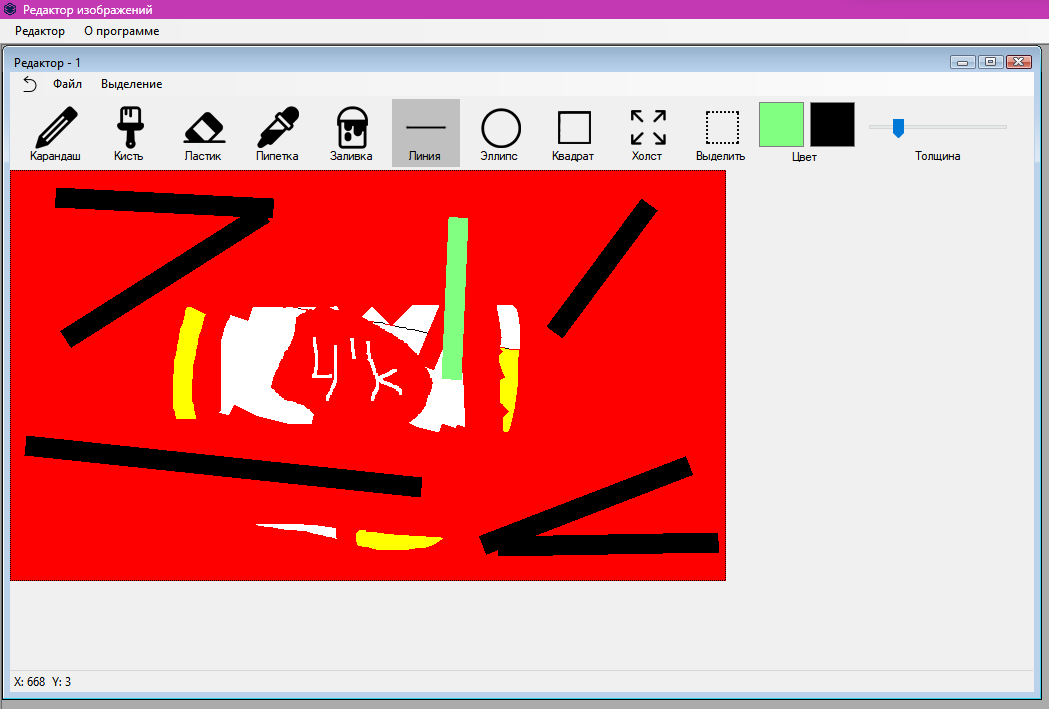


Рисунок 11. Рисование линий

Для рисования овалов и кругов используйте инструмент «Эллипс». Для рисования ровного круга нажмите и удерживайте клавишу CTRL. Чтобы нарисовать полый круг используйте левую кнопку мыши. Чтобы нарисовать заполненный круг используйте правую кнопку мыши.

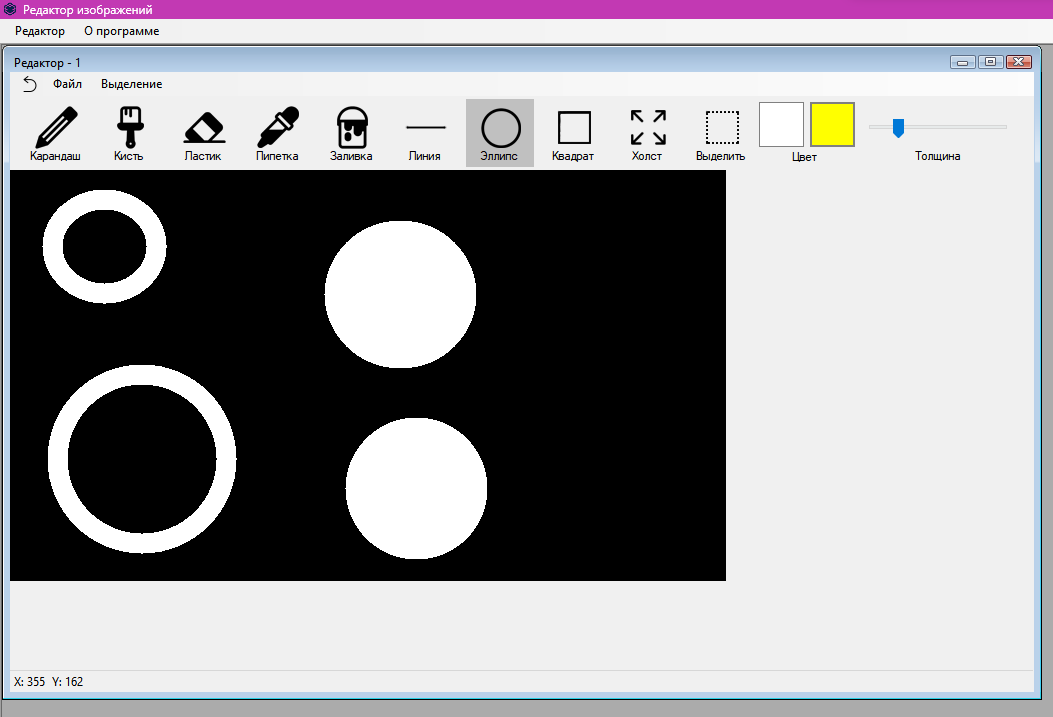


Рисунок 12. Рисование эллипсов

Для рисования прямоугольника используйте инструмент «Прямоугольник». Для рисования квадрата нажмите и удерживайте клавишу CTRL. Чтобы нарисовать полый прямоугольник используйте левую кнопку мыши. Чтобы нарисовать заполненный прямоугольник используйте правую кнопку мыши.

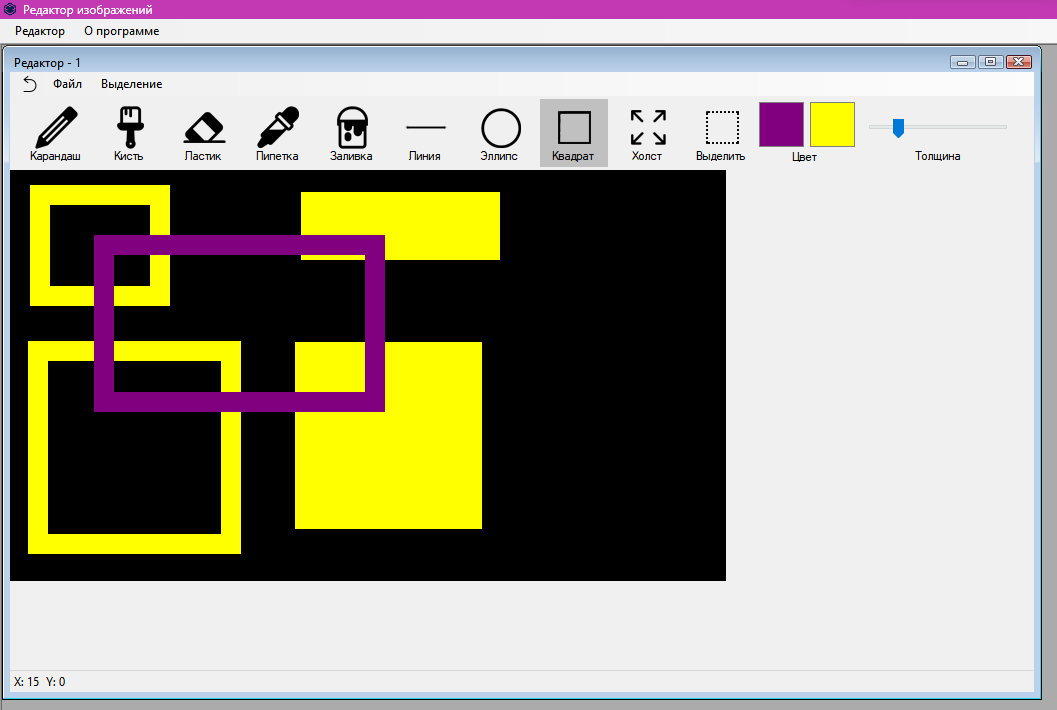


Рис. 13. Рисование прямоугольников

Для изменения размеров холста используйте инструмент «Холст». Чтобы изменить размер, удерживайте правый нижний угол изображения и растягивайте холст которые изображены на рисунке 7 номер 9.

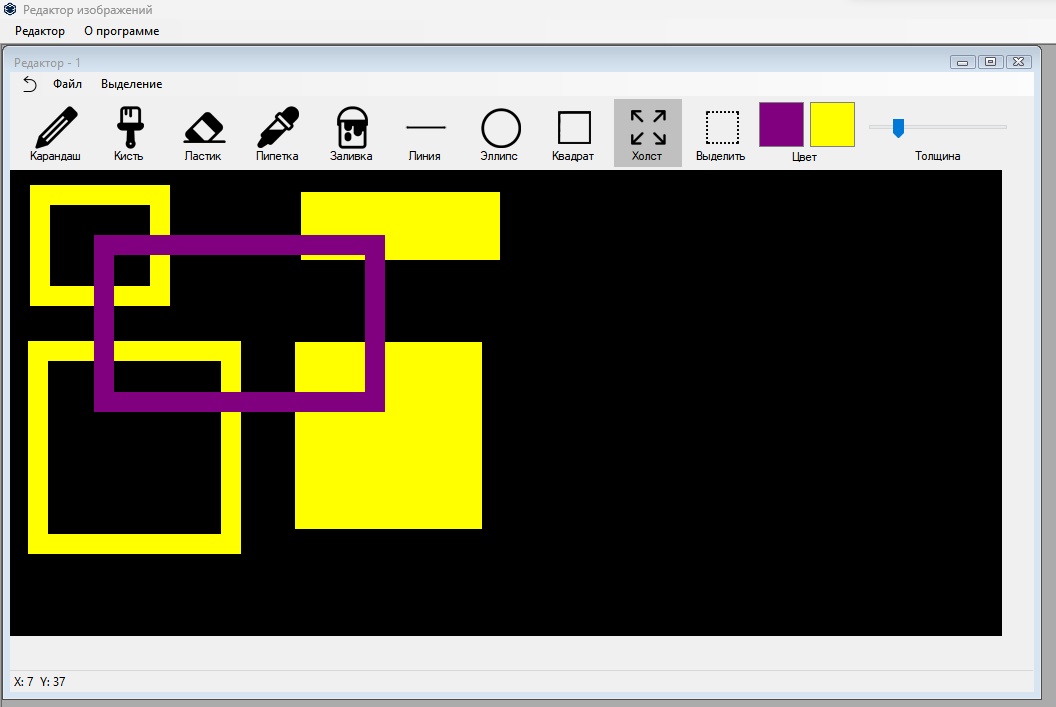


Рис. 14. Измененный размер изображения

Для выделения фрагментов изображения используйте инструмент «Выделение». Вокруг выделенной области отобразится прямоугольник. Выделенную зону можно копировать, вырезать в буфер обмена и удалять. Чтобы выделить весь холст, нажмите сочетание клавиш CTRL + A.

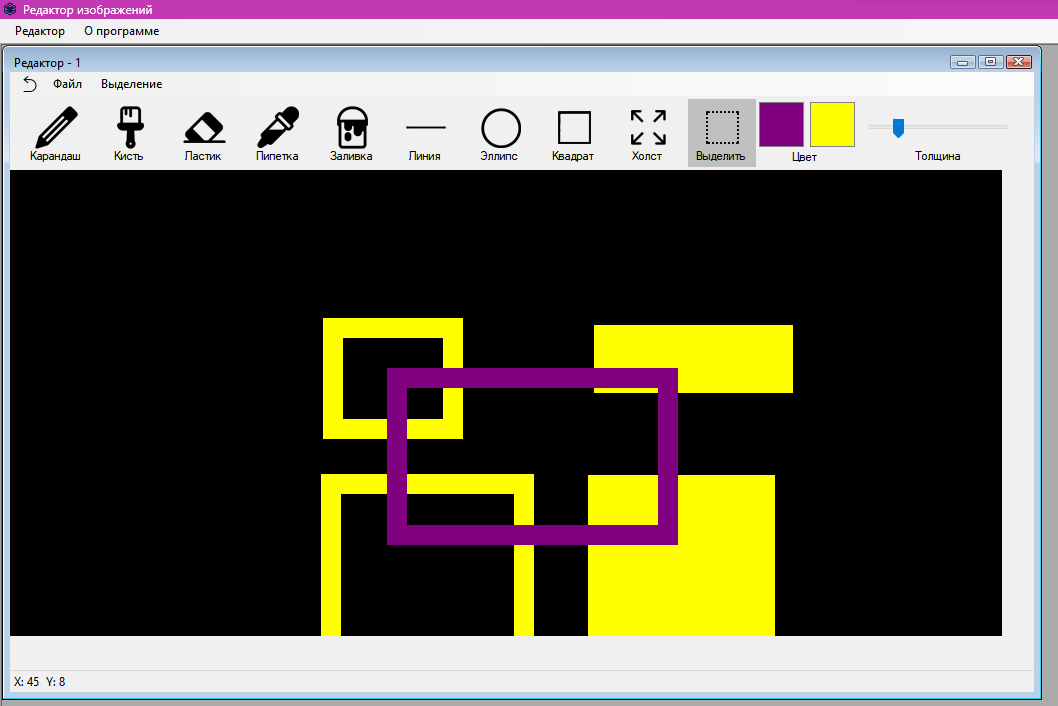


Рисунок 15. Работа с выделенным фрагментом

Для сохранения изображения перейдите Файл -> Сохранить либо нажмите сочетание клавиш CTRL + S.

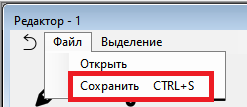


Рисунок 16. Работа с выделенным фрагментом

Общий вывод

В ходе контрольной работы я вспомнил основы программирования на C#, познакомился со структурами и ранее незнакомыми мне объектами и методами. Смог реализовать несколько простых программ. Нашел некоторые недостатки и баги программах, которые учел и постараюсь не допустить в будущем. Активно поработал с официальной документацией к языку C# и средой Visual Studio.