МИНОБРНАУКИ РОССИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ

«ЛИПЕЦКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет автоматизации и информатики

Кафедра автоматизированных систем управления

Индивидуальное домашнее задание по объектно-ориентированному программированию

“Разработка графического редактора на языке С++ с использованием механизмов ООП”

Обучающийся \_\_\_\_\_\_\_\_\_ К.А. Власов

Группа \_\_\_\_\_\_\_\_\_ ПМ-21-1

Руководитель \_\_\_\_\_\_\_\_\_ В.В.Кургасов

Липецк 2022

**Цель работы**

Закрепить навыки использования механизмов ООП на примере реализации графического редактора.

###### Задание кафедры

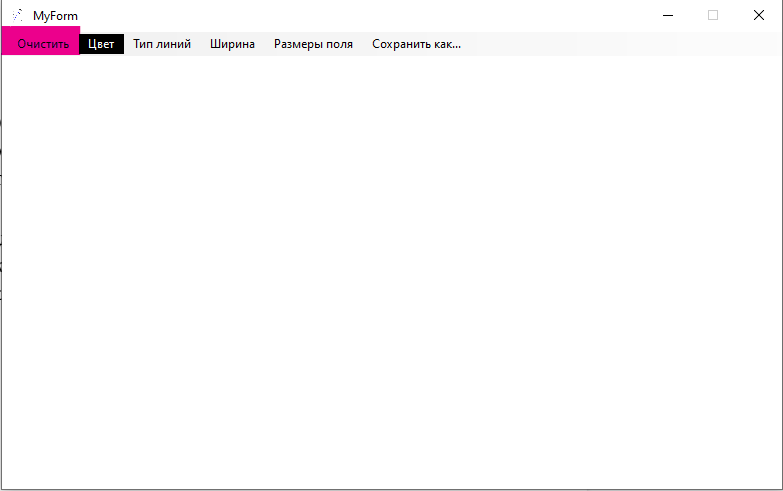
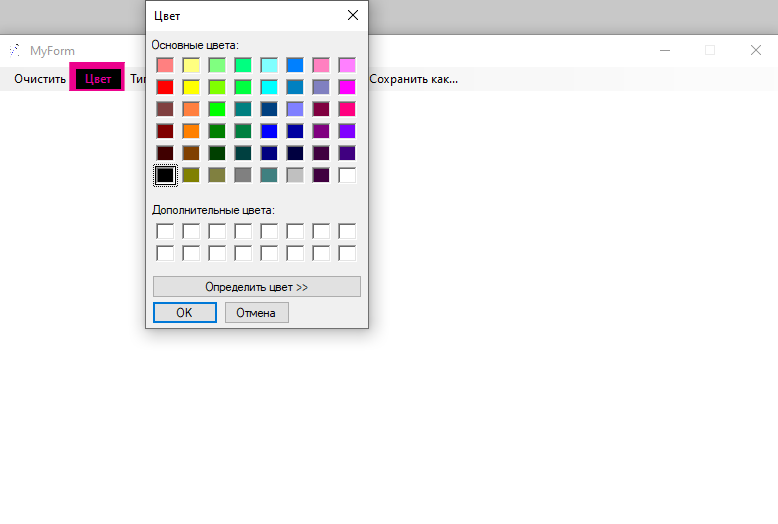
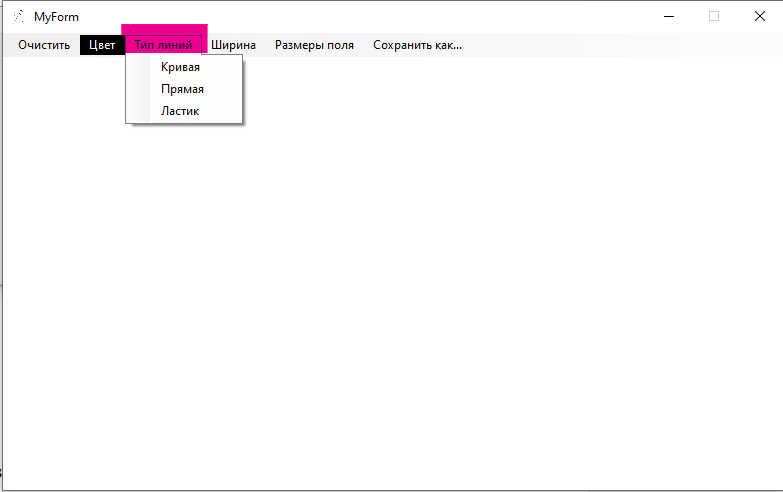
Реализовать на языке C++ редактор графической схемы (табл. 1). В ходе выполнения работы обязательно применение объектно-ориентированных возможностей языка С++: наследования и динамического полиморфизма. Каждый тип элемента схемы должен быть представлен в программе в виде отдельного класса, который унаследован от базового класса "графический элемент" (имеющего чисто виртуальную функцию прорисовки). Также необходим один класс "поле рисования", который содержит все графические элементы и отвечает за вызов функций прорисовки. Хранение графических элементов осуществляется с использованием контейнеров стандартной библиотеки C++.

| Вариант | Редактор |
| --- | --- |
| 3 | Векторный графический редактор |

Табл. 1. - Задачи для реализации механизма событий

## 

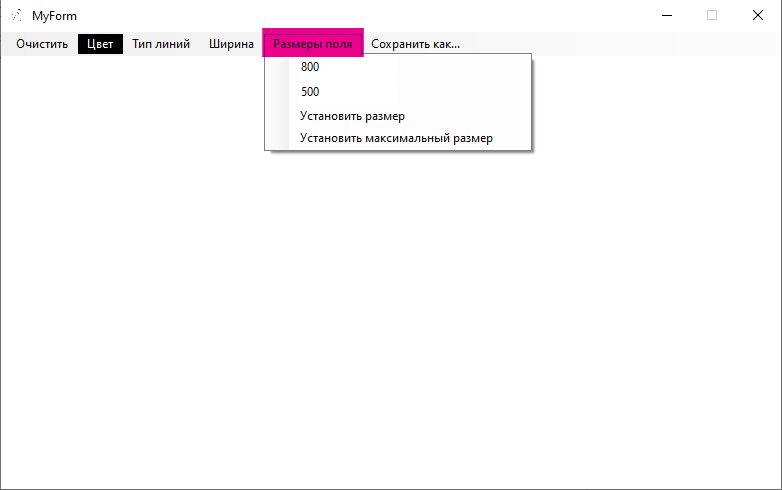
## Руководство пользователя

1. По умолчанию режим рисования “прямая”.
2. Чтобы очистить всё поле - нажмите кнопку “очистить”
3. Чтобы изменить цвет, нажмите на кнопку “цвет”.
   1. Затем выберите желаемый цвет и нажмите на кнопку “OK”.
4. Если хотите изменить тип линий, то нажмите на кнопку “Тип линий”.
5. Если хотите изменить ширину, то нажмите на кнопку “Ширина”.

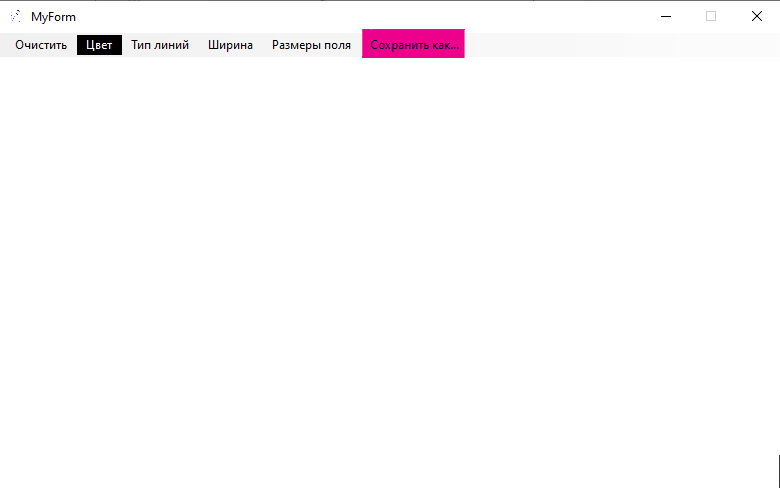


* 1. Затем установите значение шрифта от 1 до 25 и нажмита на кнопку “Установить значение”.

1. Если хотите изменить размер страницы, то нажмите на кнопку “Изменить размеры поля”.



* 1. Затем напишите в первой строчке желаемый размер ширины от 400, во второй желаемый размер высоты от 400 и нажмите на кнопку “Установить размер”.
  2. Если вы хотите установить максимальный размер для вашего монитора, то нажмите на кнопку “установить максимальный размер”.

1. Если вы хотите сохранить изображение, то нажмите на кнопку “Сохранить как,,,”, после чего выберите желаемый путь для файла.

## 

## Вывод

Закрепил навыки использования механизмов ООП на примере реализации графического редактора.

## 

## Контрольные вопросы

#### Зачем нужен перегруженный оператор присваивания?

Для того, чтобы объекту определенного типа можно было бы присвоить значение другого объекта, что избавляет программиста от необходимости постоянного прямого перевода.

Например есть класс Matrix, внутри которого есть двухмерный массив, так почему бы не сделать оператор присваивания для двухмерного массива или vector<vector<Type>>, что позволит удобно работать с объектом типа Matrix в случае, если необходимо будет приравнивание такого объекта к двухмерному массиву/vector<vector<Type>>

#### Зачем нужен механизм наследования?

Основное назначение механизма наследования — повторное использование кодов, так как большинство используемых типов данных являются вариантами друг друга, и писать для каждого свой класс нецелесообразно.

#### Зачем используются модификаторы при наследовании классов? Какие это модификаторы?

При объявлении порождаемого класса модификатор доступа может принимать значения *public, private, protected* либо отсутствовать, по умолчанию используется значение private.

* private: закрытый или приватный компонент класса или структуры. Приватный компонент доступен только в рамках своего класса или структуры
* protected: такой компонент класса доступен из любого места в своем классе или в производных классах. При этом производные классы могут располагаться в других сборках.
* public: публичный, общедоступный компонент класса или структуры. Такой компонент доступен из любого места в коде, а также из других программ и сборок.

#### Зачем нужен механизм полиморфизма?

Полиморфизм — это способность объекта использовать методы производного класса, который не существует на момент создания базового

Полиморфизм позволяет писать более абстрактные программы и повысить коэффициент повторного использования кода.

Кратко смысл полиморфизма можно выразить фразой: «Один интерфейс, множество реализаций».

#### Что понимается под динамическим полиморфизмом?

Динамический полиморфизм использует виртуальные функции для достижения полиморфизма во время выполнения, что означает, что когда система компилируется, она не знает, какую функцию вызовет программа, и только при ее запуске можно определить, какая функция перейдет к следующей.

#### Что такое интерфейс класса?

Интерфейсным классом называется абстрактный класс, не имеющий данных и состоящий в основном из чисто виртуальных функций. Такой класс может иметь обычные виртуальные функции (не чисто виртуальные), например деструктор. Также могут быть статические функции-члены.

#### Зачем нужен чисто виртуальный метод? Как он выглядит?

Чисто виртуальная функция является функцией, которая объявляется в базовом классе, но не имеет в нем определения. Поскольку она не имеет определения, то есть тела в этом базовом классе, то всякий производный класс обязан иметь свою собственную версию определения. Для объявления чисто виртуальной функции используется следующая общая форма:

### virtual тип имя\_функции(список\_параметров) = 0;

или

### virtual void calculate() = 0;

При введении чисто виртуальной функции в производном классе обязательно необходимо опре­делить свою собственную реализацию этой функции. Если класс не будет содержать определения этой функции, то компилятор выдаст ошибку.

Класс с таким методом нельзя создать

#### Какой класс называется абстрактным?

Абстрактные классы - это классы, которые содержат или наследуют без переопределения хотя бы одну чистую виртуальную функцию. Абстрактный класс определяет интерфейс для переопределения производными классами.

Важной особенностью абстрактных классов является то, что не существует ни одного объекта данного класса. Вместо этого абстрактный класс служит в качестве базового для других производных классов.

Причина, по которой абстрактный класс не может быть ис­пользован для объявления объекта, заключается в том, что одна или несколько его функций-членов не имеют определения. Тем не менее, даже если базовый класс является абстрактным, все равно можно объявлять указатели или ссылки на него, с помощью которых затем поддерживает­ся полиморфизм времени исполнения.