Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Российский государственный университет им. А.Н. Косыгина» (Технологии. Дизайн. Искусство)

Институт Мехатроники и информационных технологий

Кафедра Информационных технологий

КУРСОВАЯ РАБОТА

БАКАЛАВРА

по дисциплине: «Геометрическое моделирование»

на тему: Разработка графического редактора

Направление подготовки: 09.03.01 Инфоpматика и вычислительная техника

Профиль: Системы автоматизированного проектирования

Выполнила Шишерина М.А.,

студентка группы МВС-117 3 курса очной формы обучения

\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*подпись фамилия, имя, отчество*

\_\_\_\_\_\_\_\_\_\_\_\_ «\_\_\_» \_\_\_\_\_\_\_\_\_ 2020г.

*оценка дата сдачи*

Преподаватель \_\_\_\_\_\_\_\_\_\_ \_ /ст.пр. А.М. Козлов/\_\_\_\_\_

*подпись ученая степень, ученое звание, фамилия, инициалы*

Москва 2020

ОГЛАВЛЕНИЕ

[ВВЕДЕНИЕ 2](#_Toc39685291)

[1. Выбор алгоритмов и структур данных 4](#_Toc39685292)

[1.1 Структуры данных 4](#_Toc39685293)

[1.2 Реализация классов 6](#_Toc39685294)

[1.3 Реализация выделительной рамки 8](#_Toc39685295)

[1.3.1 Метод accessory\_point\_figure() для класса Segment 8](#_Toc39685296)

[1.3.2 Метод accessory\_point\_figure() для класса MyCircle 9](#_Toc39685297)

[1.3.3 Метод accessory\_point\_figure() для классов MyArc, MyPoliline, MyText 10](#_Toc39685298)

[1.4 Полезные функции 12](#_Toc39685299)

[1.4.1 Функция redrawing() 12](#_Toc39685300)

[1.4.2 Функция vectors\_cleanin() 13](#_Toc39685301)

[1.4.3 Функция figures\_choice() 13](#_Toc39685302)

[1.4.4 Функция remove\_selection () 14](#_Toc39685303)

[1.5 Удаление объектов 15](#_Toc39685304)

[2. Руководство пользователя 17](#_Toc39685305)

[2.1 Создание нового изображения 17](#_Toc39685306)

[2.2 Инструмент «Отрезок» 18](#_Toc39685307)

[2.3 Инструмент «Полилиния» 20](#_Toc39685308)

[2.4 Инструмент «Окружность» 21](#_Toc39685309)

[2.5 Инструмент «Дуга» 22](#_Toc39685310)

[2.6 Инструмент «Текст» 23](#_Toc39685311)

[2.7 Инструмент «Прямоугольная рамка» 25](#_Toc39685312)

[2.8 Инструмент «Стрелка» 28](#_Toc39685313)

[2.9 Сохранение изображения 29](#_Toc39685314)

[2.10 Загрузка существующего изображения 31](#_Toc39685315)

[2.11 Функция Close All 33](#_Toc39685316)

[СПИСОК ЛИТЕРАТУРЫ 34](#_Toc39685317)

# ВВЕДЕНИЕ

Графический редактор – программа, позволяющая создавать, просматривать, обрабатывать и редактировать цифровые изображения на компьютере.

Задачей данной курсовой работы является разработка графического редактора, реализующего следующие функции:

1. Построение объектов: отрезок, полилиния (состоящая из чередующихся отрезков и дуг, построенных по трем точкам), окружность, дуга.
2. Удаление выбранных объектов.
3. Вывод и редактирование текстов с возможностью выбора типа и размера шрифтов, угла поворота текста и места размещения на экране.
4. Выбор объектов с использованием прямоугольной рамки (выбранными считаются те объекты, которые полностью или частично попали внутрь прямоугольной области)
5. Организация диалога с помощью меню. Построение и удаление объектов должно быть представлено в виде режимов работы. Ввод объектов производить в режиме “тянущаяся линия”.
6. Сохранение и считывание результатов работы в файлах.

Язык программирования: С++

Среда разработки: Embarcadero C++ Builder

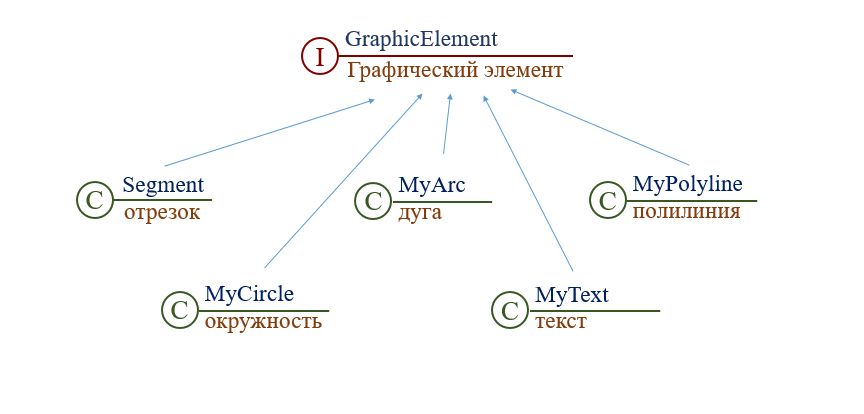
# Выбор алгоритмов и структур данных

## 1.1 Структуры данных

Объект в программировании – некоторая сущность в цифровом пространстве, обладающая определённым состоянием и поведением, имеющая определенные свойства (атрибуты) и операции над ними (методы). Как правило, при рассмотрении объектов выделяется то, что объекты принадлежат одному или нескольким классам, которые определяют поведение (являются моделью) объекта.

Объекты в данной программе:

* отрезки;
* полилинии;
* окружности;
* дуги;
* текст.

Все они представляю из себя графические элементы, поэтому для реализации был создан интерфейс GraphicElement с виртуальным методом create(), отвечающим за отрисовку объекта. GraphicElement наследуют классы Segment, MyArc, MyPolyline, MyCircle, MyText, которые реализуют его метод.

Для дальнейшей перерисовки, созданные объекты нужно где-то хранить. Эта задача решается с помощью контейнера vector.

Почему именно vector?

1. Не имеет фиксированного размера;
2. Позволяет легко добавлять элементы в конец;
3. К элементам вектора можно обращаться по индексу;
4. Имеет функцию очистки всего вектора;
5. Позволяет удалять элементы по индексу.

Таким образом, данный контейнер подходит для наших задач.

В программе имеется 5 векторов, каждый из которых хранит в себе экземпляры одного из представленных выше классов:

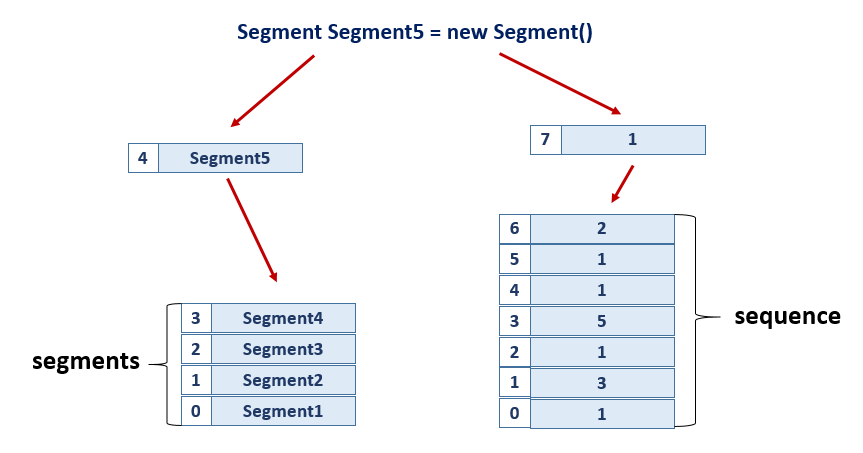
vector<Segment> segments; // отрезки

vector<MyArc> arcs; // дуги

vector<MyPolyline> polylines; // полилинии

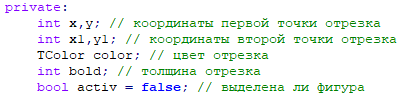
vector<MyCircle> circles; // окружности

vector<MyText> strings; // строки текста

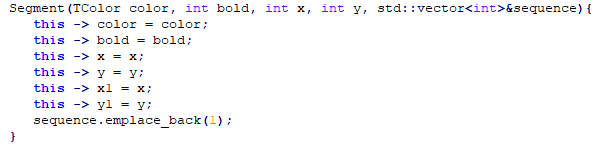
Чтобы запоминать последовательность отрисовки элементов добавлен еще один vector<int> sequence. При создании нового графического элемента в него добавляется число, соответствующее типу этого элемента: 1 – отрезок, 2 – дуга, 3 – полилиния, 4 – окружность, 5 – текст.

## 1.2 Реализация классов

Рассмотрим на примере классаSegment, который отвечает за создание отрезков.

Атрибуты:

Конструктор:

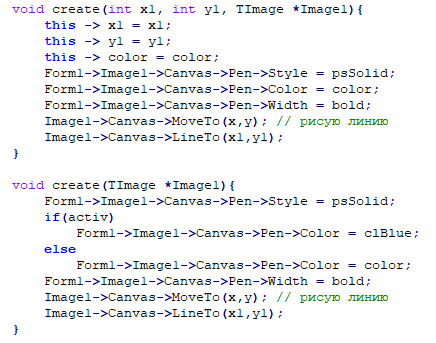


При создании экземпляра класса конструктор инициализирует его поля и добавляет в вектор sequence число, соответствующее данному классу.

Методы:

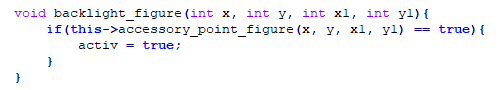
Класс содержит два метода create() предназначенные для разных ситуаций:

* первый метод используется, при изначальном создании объекта и дополнительно задает конечные координаты отрезка. Так же он используется для имитации режима «тянущаяся линия»;
* второй метод принимает только объект типа Image, на котором требуется нарисовать отрезок и строит его по заполненным ранее полям.

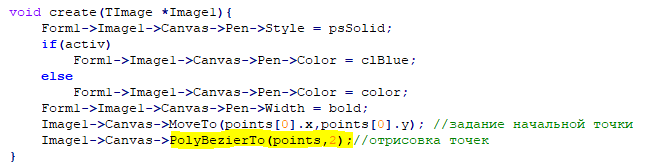


Метод accessory\_point\_figure() проверяет, принадлежит ли фигура

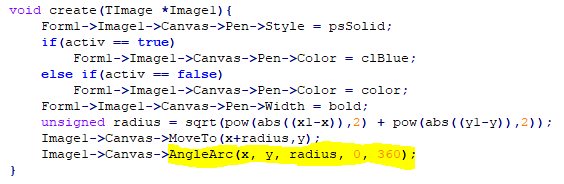
заданной рамке и возвращает значение типа bool. По результатам работы accessory\_point\_figure(), backlight\_figure() меняет значение атрибута active на true, если объект выделен прямоугольной рамкой.



Все остальные классы организованы аналогичным образом, за исключением некоторых особенностей отрисовки. Так, например, в классе MyArc для создания дуги используется метод Canvas PolyBezierTo(const array of TPoint Points), принимающий массив точек, по которым строится кривая Безье,

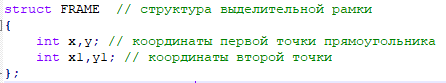


а в классе MyCircle используется метод ngleArc(int X, int Y, unsigned Radius, float StartAngle, float SweepAngle).



## 1.3 Реализация выделительной рамки

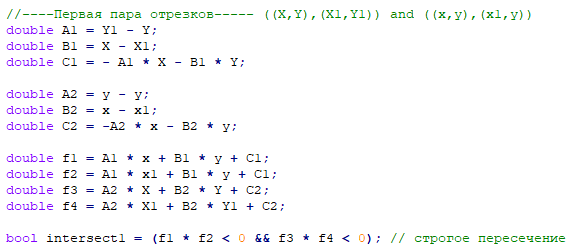
Выделительная прямоугольная рамка представлена в программе структурой:



Каждый класс содержит в себе метод accessory\_point\_figure(), проверяющий, принадлежит ли фигура заданной рамке и возвращает значение типа bool.

### 1.3.1 Метод accessory\_point\_figure() для класса Segment

1) Определяем, пересекает ли прямая каждую из сторон прямоугольника. Так как сторона прямоугольника тоже является отрезком, необходимо решить задачу о пересечении двух отрезков.

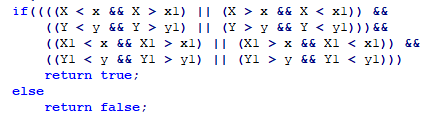


Где X, Y и X1, Y1 – координаты точек самого отрезка, а x, y и x1, y – координаты одной из сторон прямоугольника.

Повторяем алгоритм для всех сторон прямоугольной рамки.

2) Если отрезок пересекается хотя бы с одной из сторон прямоугольника, значит его часть входит внутрь рамки (вернуть true), если нет, то нужно проверить, не находится ли отрезок внутри рамки полностью.

3) Для этого проверяем находятся ли обе его точки или одна из них внутри рамки:



если да, то возвращаем true, если нет, то false.

### 1.3.2 Метод accessory\_point\_figure() для класса MyCircle

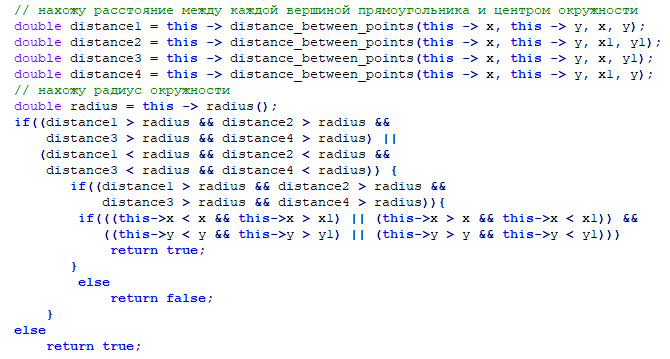
1) Если расстояние между центром окружности и, хотя бы, одной из вершин прямоугольника меньше радиуса – они пересекаются (true), иначе непонятно.

Для данной задачи нам понадобится дополнительный метод double distance\_between\_points(int x, int y, int x1, int y1), который находит расстояние от одной точки до другой по их координатам.



Изначально это формула радиуса, выраженная из уравнения окружности

Узнаем расстояние от центра окружности до каждой из вершин прямоугольника и сравниваем их с радиусом:

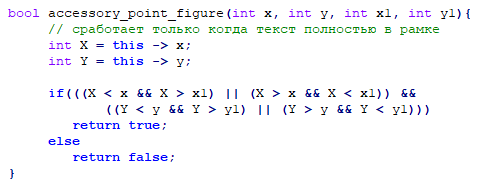


2) Если после выполнения первого пункта не ясно, пересекаются ли фигуры, необходимо проверить, находится ли окружность внутри прямоугольника. Для этого смотрим, где относительно рамки расположены ее точки.

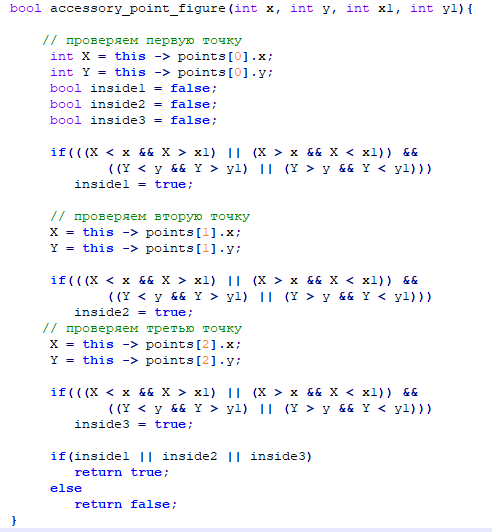
### 1.3.3 Метод accessory\_point\_figure() для классов MyArc, MyPoliline, MyText

Для данных классов принадлежность объекта прямоугольной области определяется принадлежностью его точек. Поэтому, в отличие от двух предыдущих классов, здесь может возникнуть ситуация, когда часть фигуры попадает в область прямоугольной рамки, но не выделяется.

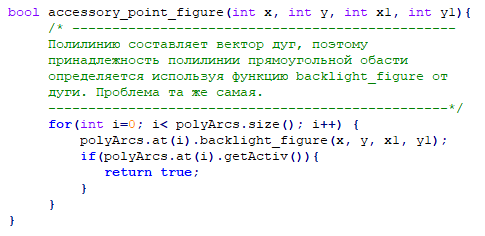
Текст



Полилиния



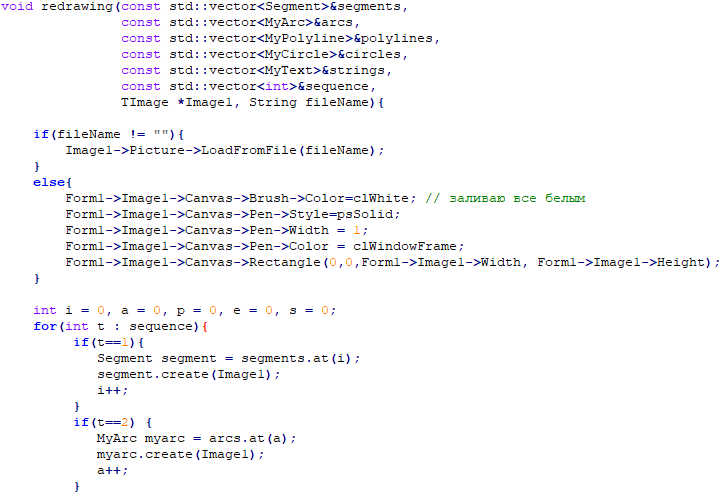
Дуга

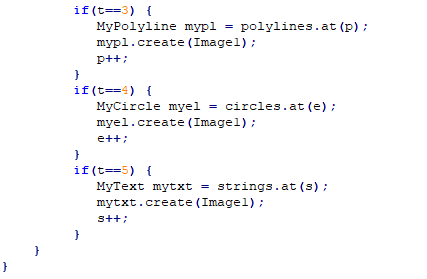


## 1.4 Полезные функции

### 1.4.1 Функция redrawing()

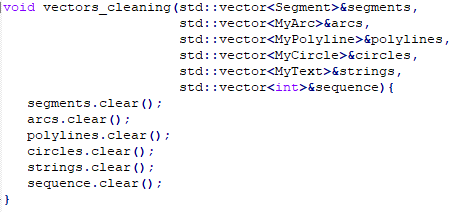
Принимает векторы, содержащие объекты классов-потомков интерфейса GraphicElement и перерисовывает все элементы в заданном вектором sequence порядке, вызывая у каждого метод create().





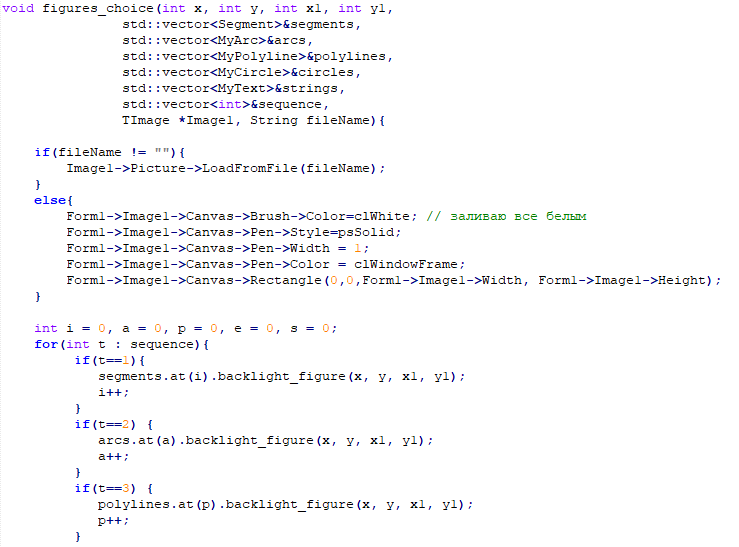
### 1.4.2 Функция vectors\_cleanin()

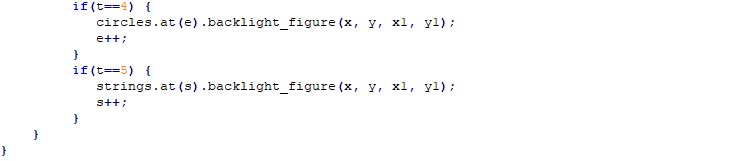
Очищает все векторы.



### 1.4.3 Функция figures\_choice()

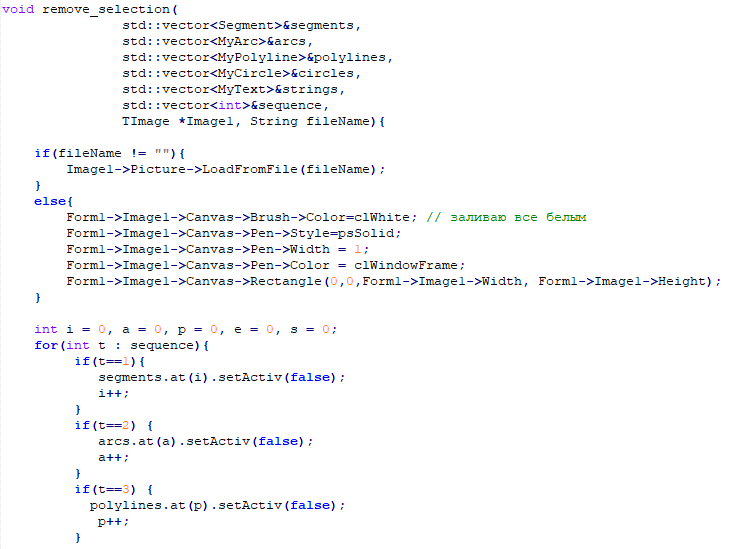
Проверяет все фигуры на принадлежность области прямоугольной рамки и выделяет их, если они в нее входят.

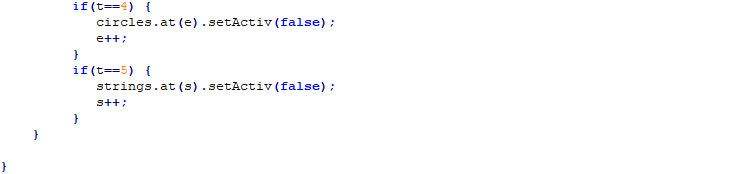




### 1.4.4 Функция remove\_selection ()

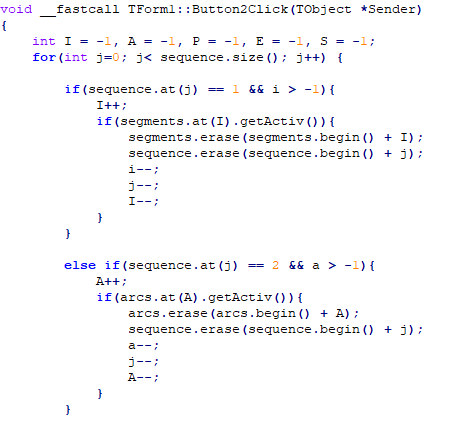
Снимает выделение со всех фигур.





## 1.5 Удаление объектов

При нажатии на кнопку «Удалить» происходит обход по всем векторам графических элементов и те, поле active у которых true удаляются из векторов. Происходит перерисовка изображения.

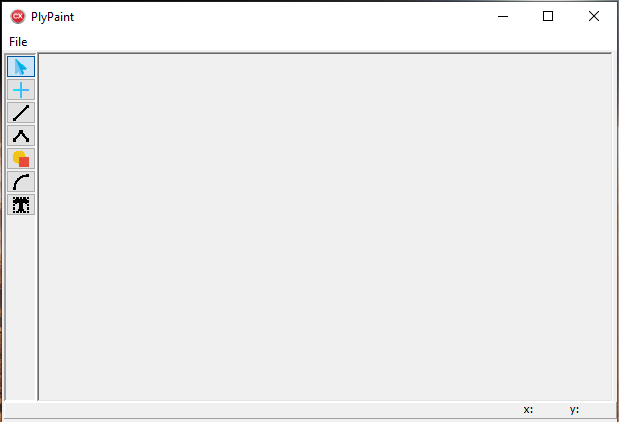






# Руководство пользователя

При запуске программы открывается ее окно



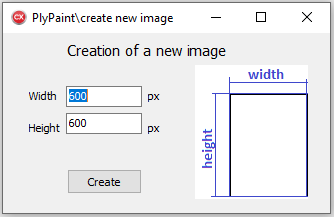
меню

Расположение указателя на рабочей области (px)

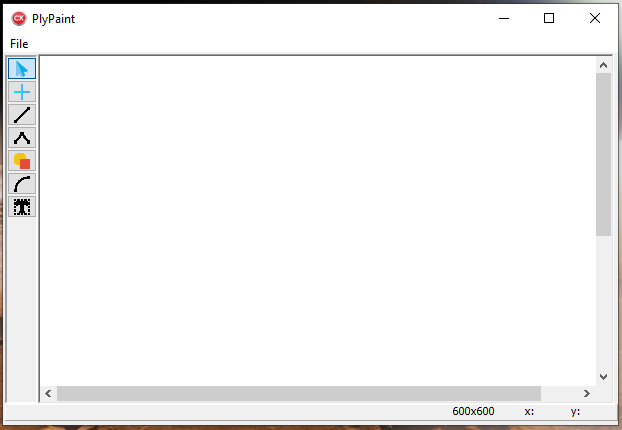
Панель инструментов

## 2.1 Создание нового изображения

Перед тем, как начать рисовать, необходимо создать новую рабочую область. Для этого перейдите в File -> New. Появится окно, позволяющее выбрать размеры рабочей области в пикселях.



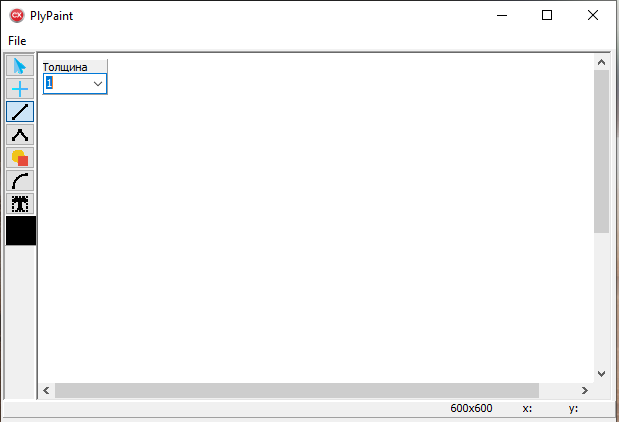
Для создания изображения нажмите на кнопку «Create».



Рабочая область

## 2.2 Инструмент «Отрезок»

Нажмите на кнопку «Отрезок» на панели инструментов.

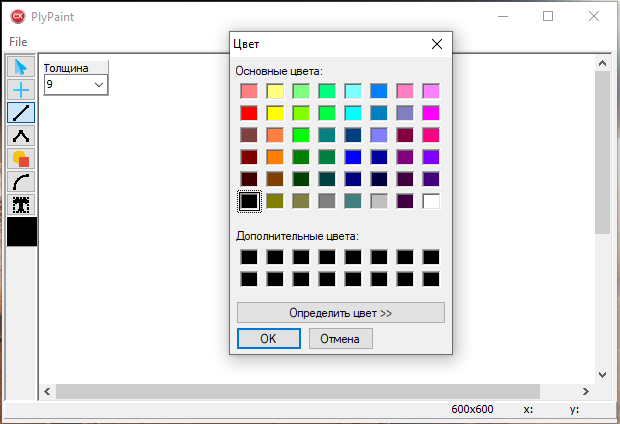


Толщина линии

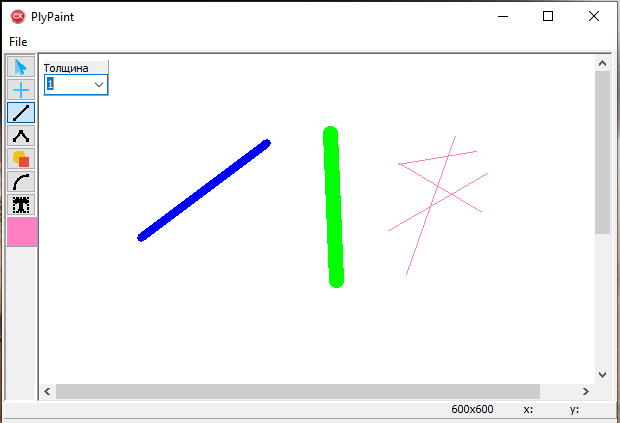
Выбор цвета

Инструмент «Отрезок»

Появятся две панели, позволяющие выбрать толщину и цвет линии.

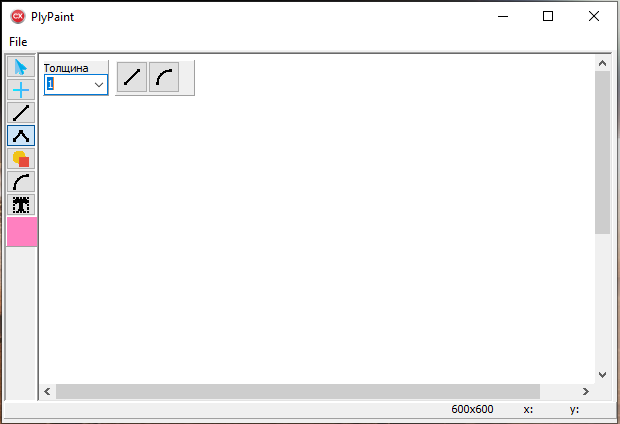


Первое нажатие левой кнопки мыши по рабочей области задает начало отрезка, второе – конец.



## 2.3 Инструмент «Полилиния»

Нажмите на кнопку «Полилиния» на панели инструментов.



Инструмент «Полилиния»

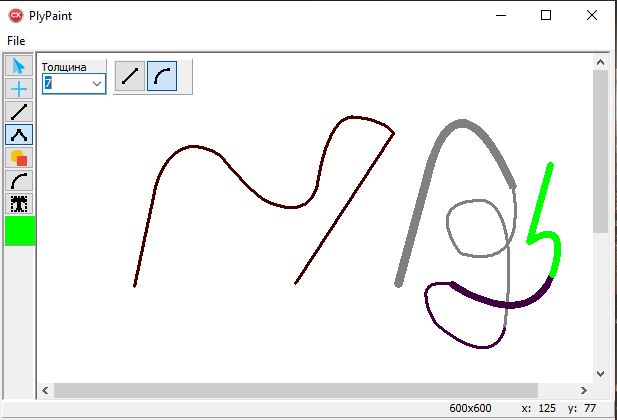
Тип сегмента

Толщина сегмента

Выбор цвета

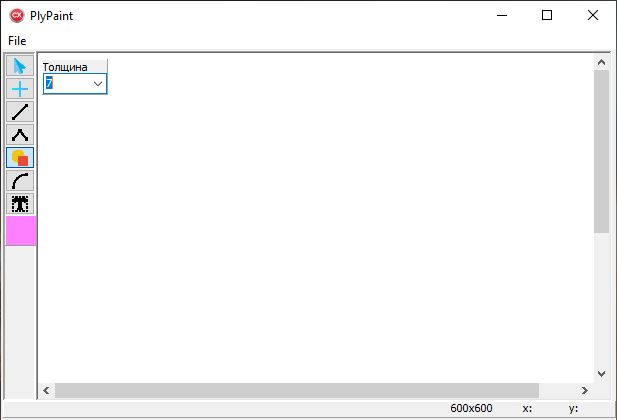
Инструмент «Полилиния» позволяет создавать объект, состоящий из сегментов разного типа. Программа предоставляет пользователю возможность создавать полилинию из отрезков и дуг, построенных по трем точкам.

Для рисования нажмите левой кнопкой мыши на рабочей области, для прекращения создания полилинии нажмите правой кнопкой мыши.



## 2.4 Инструмент «Окружность»

Нажмите на кнопку «Окружность» на панели инструментов.

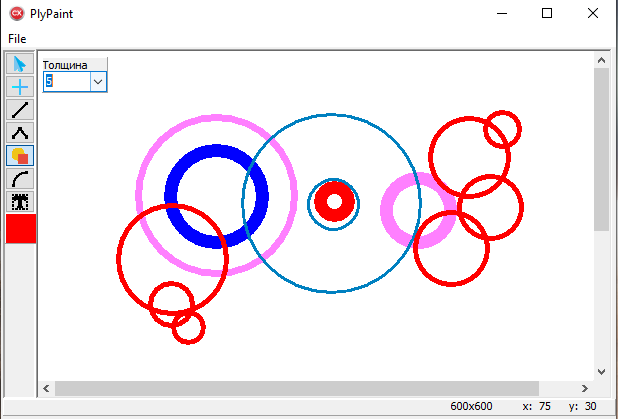


Инструмент «Окружность»

Выбор цвета

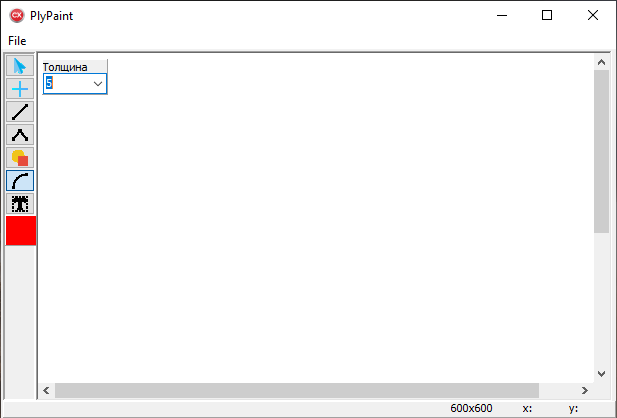
Толщина линии

Первое нажатие левой кнопки мыши задает центр окружности, второе – точку на самой окружности.



## 2.5 Инструмент «Дуга»

Нажмите на кнопку «Дуга» на панели инструментов.



Инструмент «Дуга»

Толщина линии

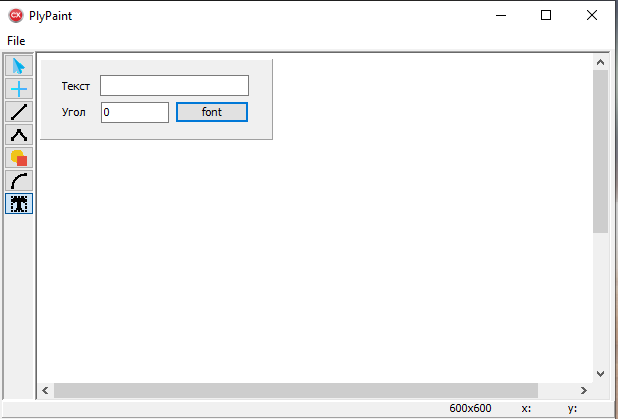
Выбор цвета

Данный инструмент строит кривую Безье по трем точкам, заданным пользователем.



## 2.6 Инструмент «Текст»

Нажмите на кнопку «Текст» на панели инструментов.



Поле ввода строки

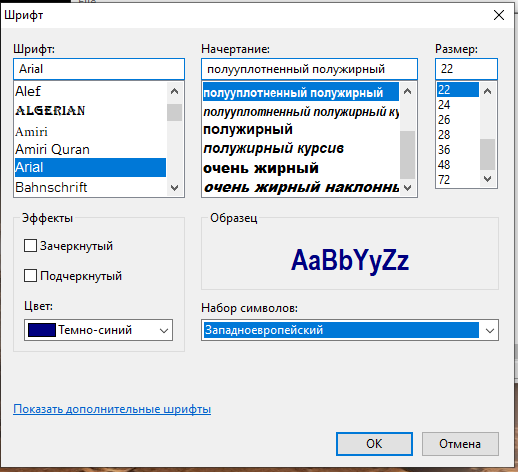
Поле ввода угла наклона текста

Кнопка выбора шрифта

Инструмент «Текст»

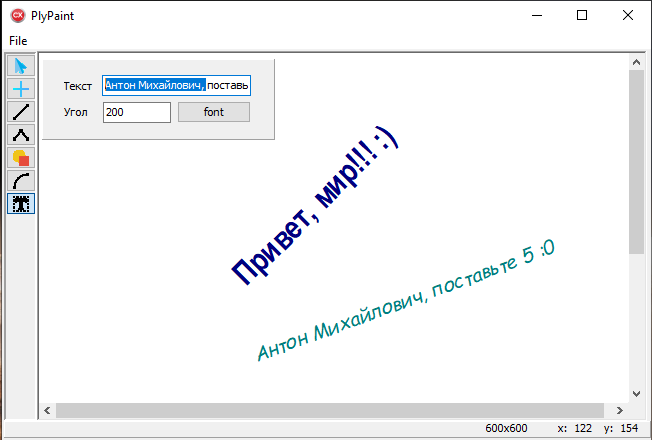
Панель редактирования текста, как и другие дополнительные панели для инструментов можно передвигать.

Кнопка «Font» открывает диалог выбора шрифта.



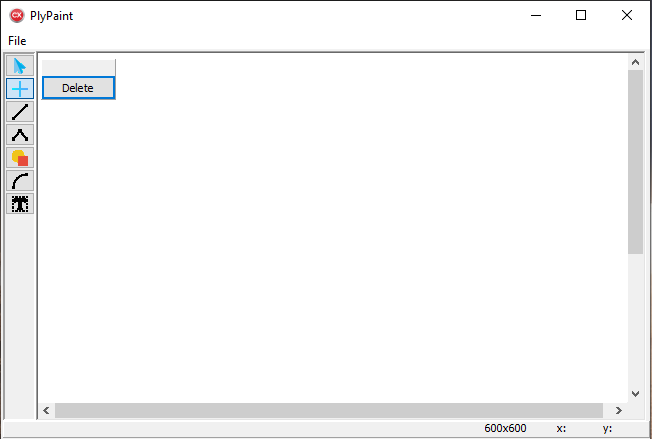
Можно задать угол поворота строки текста относительно оси x в десятых единицах градуса. Т.е. 900 = 90 градусов.

Для отображения текста на рабочей области выполните все необходимые настройки и нажмите на изображении левой кнопкой мыши.



## 2.7 Инструмент «Прямоугольная рамка»

Нажмите на кнопку «Прямоугольная рамка» на панели инструментов.

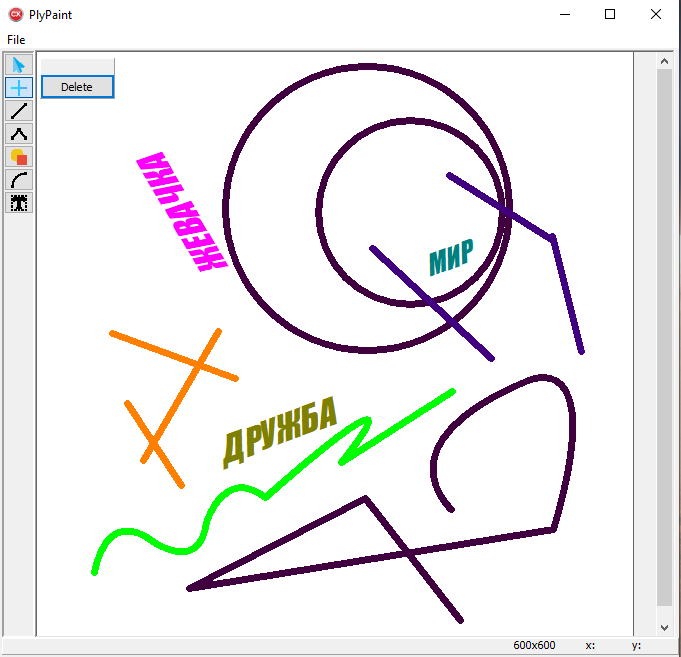


Кнопка удаления выделенных элементов

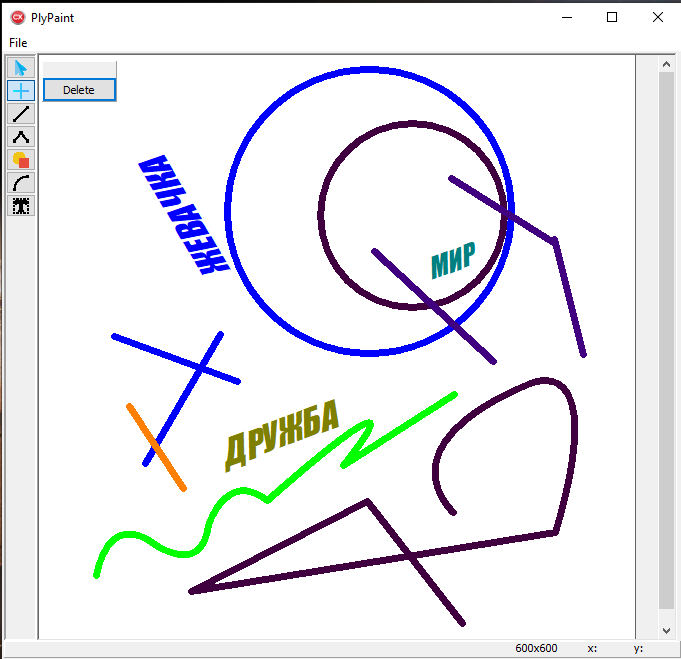
Инструмент «Прямоугольная рамка»

Этот инструмент позволяет выделять нарисованные объекты и удалять их. Выделенные объекты подсвечиваются синим цветом.

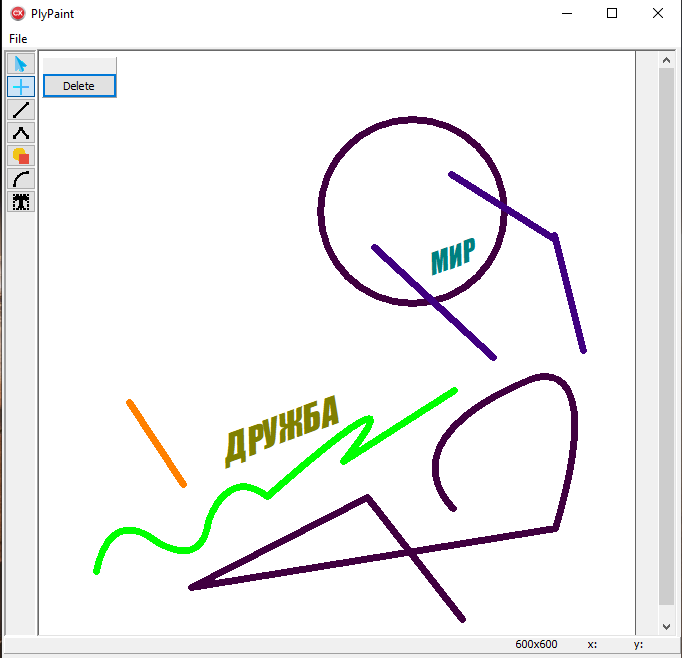
1. Нарисовали фигуры



1. Выделили фигуры

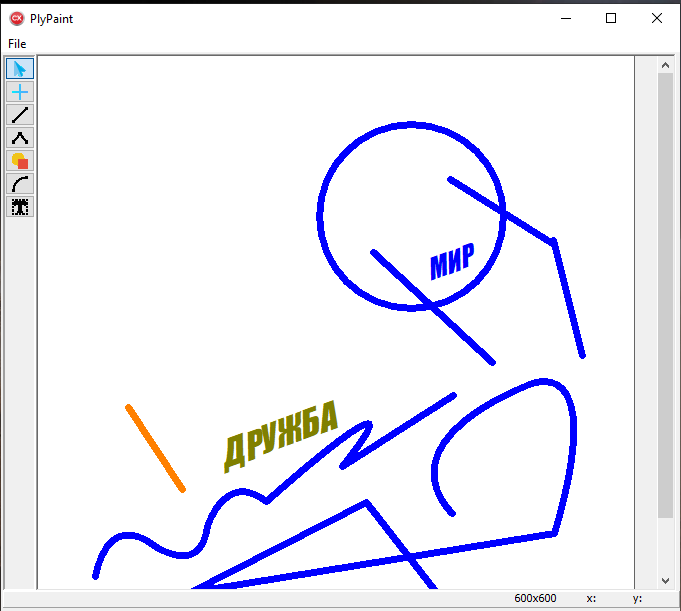


1. Удалили фигуры



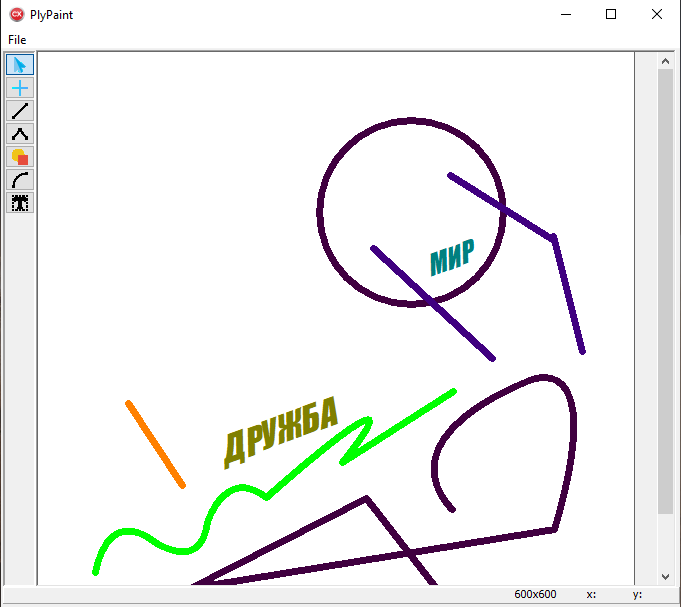
## 2.8 Инструмент «Стрелка»

Нажмите на кнопку «Стрелка» на панели инструментов.



Инструмент «Стрелка»

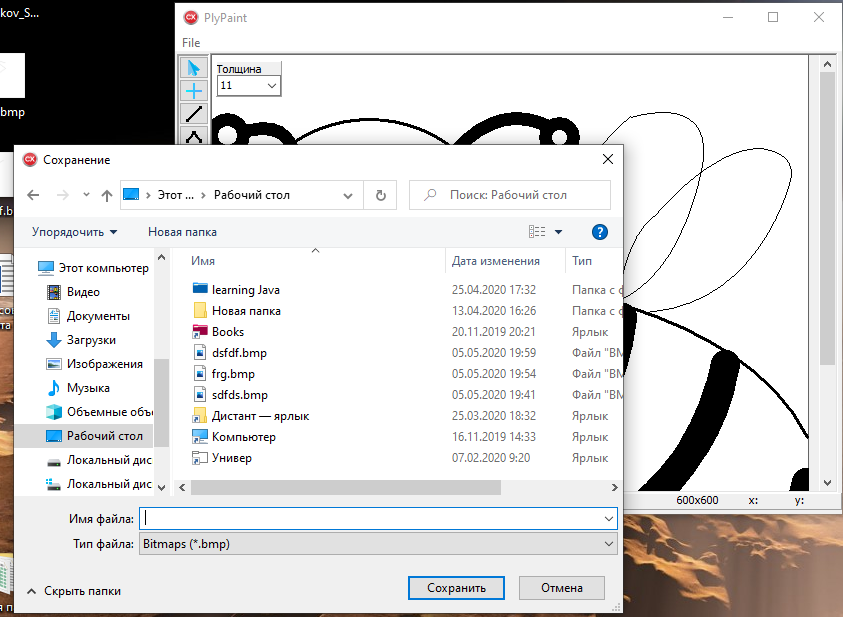
Нажмите этим инструментом на рабочей области, и он снимет выделение.



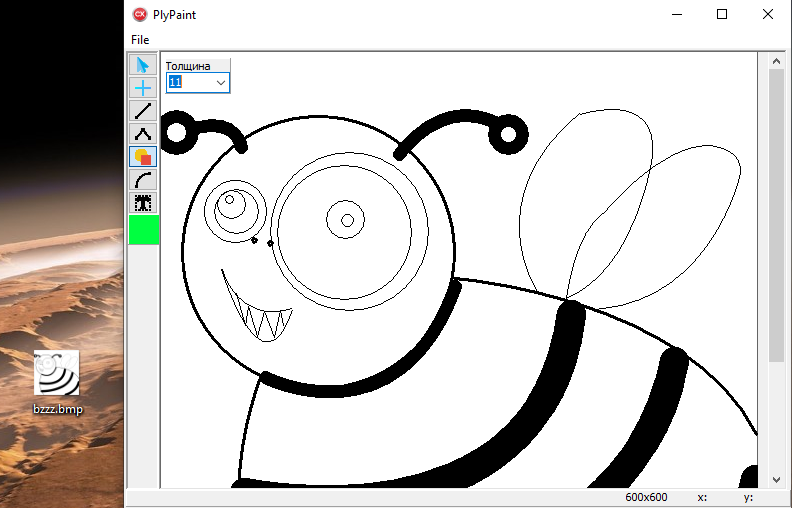
## 2.9 Сохранение изображения

Для этого перейдите в File -> Save As. Появится окно, позволяющее выбрать путь сохранения файла.

Изображения сохраняются в формате bmp.



Придумываем имя нашему изображению и сохраняем его.

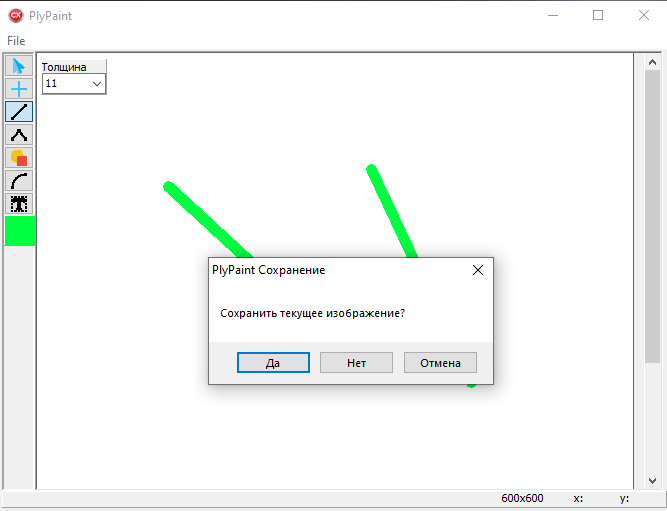


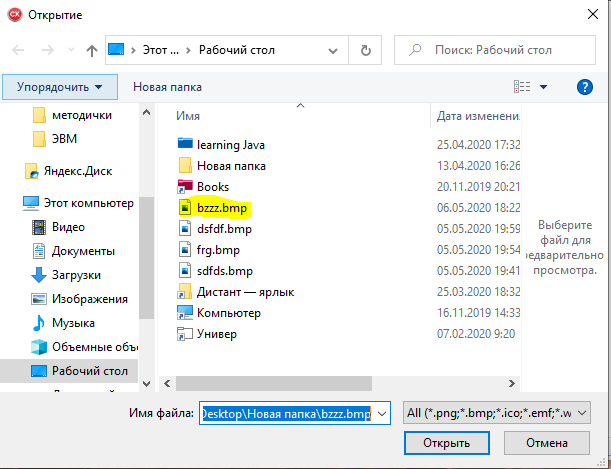
Функция File -> Save становится доступна, когда изображение уже сохраняли. Она позволяет сохранить новую версию картинки по тому же пути.

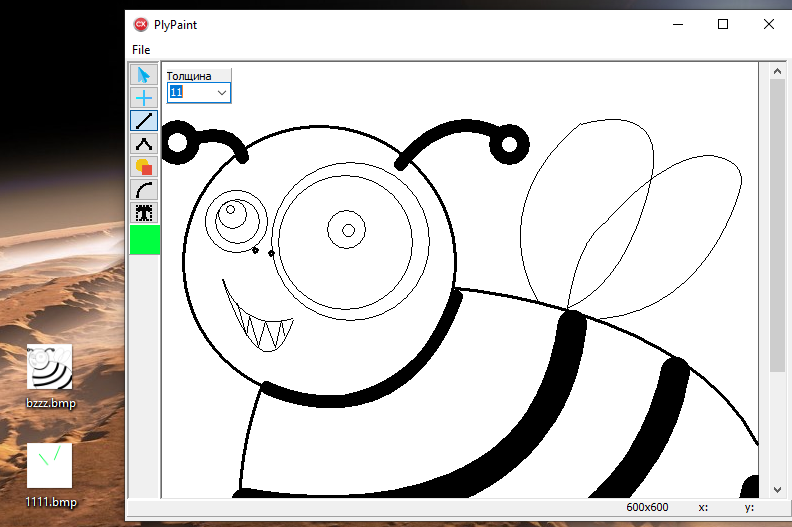
## 2.10 Загрузка существующего изображения

Для этого перейдите в File -> Open. Появится окно, позволяющее выбрать файл.

Если у Вас уже есть несохраненное изображение в программе, то будет предложено его сохранить перед открытием другого изображения.



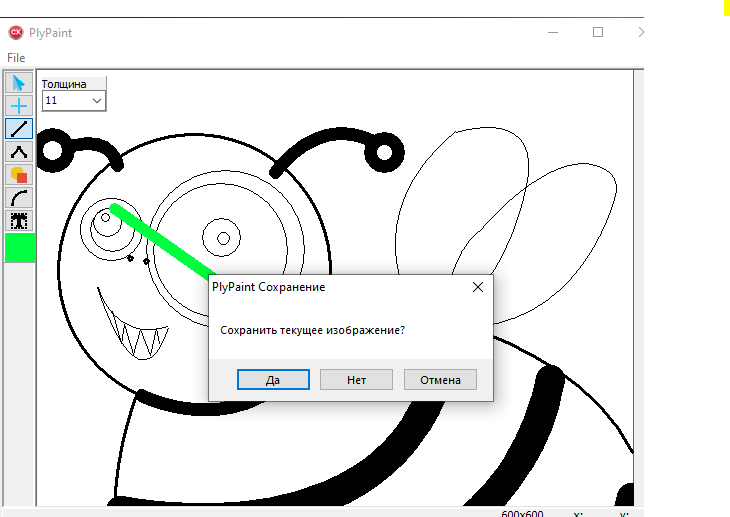


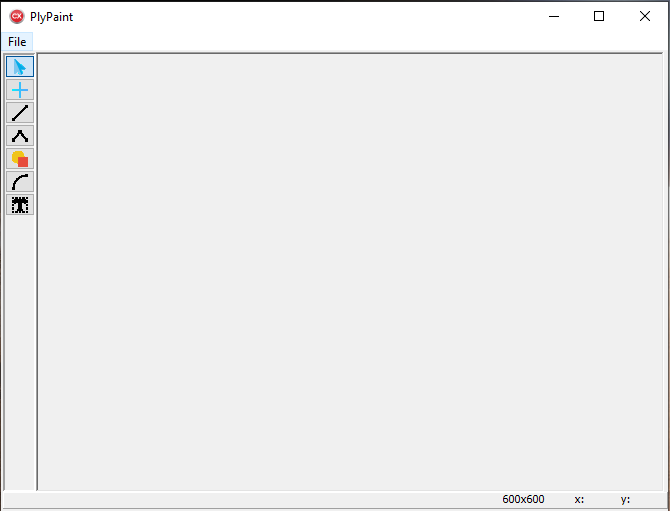


## 2.11 Функция Close All

Перейдите в File -> Close All.

Данная функция закрывает рабочую область. Если Вы работали над изображением и не сохранили его, то вам будет предложено это сделать.





# СПИСОК ЛИТЕРАТУРЫ

1. Архангельский А.Я. C++Builder 6. Справочное пособие. Книга 1. Язык C++. – М.: БиномПресс, 2002 г. – 544 с.
2. Архангельский А.Я. C++Builder 6. Справочное пособие. Книга 2. Классы и компоненты. – М.: Бином-Пресс, 2002 г. – 528 с.
3. Страуструп Бьярне. Программирование: Принципы и практика использования С++ : Пер. с англ. – М.: ООО «И. Д. Вильямс», 2011. – 1248с.