Федеральное агентство по образованию РФ

Дальневосточный государственный университет

Институт математики и компьютерных наук

Кафедра информатики

Векторный графический редактор

Практическая работа

студента Б8103-A группы

Константинова Остапа Владимировича

Научный руководитель:

Кленин Александр Сергеевич

Владивосток, 2013

# Содержание

[Содержание 2](#_Toc351332861)

[Аннотация 3](#_Toc351332862)

[Введение 3](#_Toc351332863)

[Описание предметной области 3](#_Toc351332864)

[Неформальная постановка задачи 3](#_Toc351332865)

[План работ 3](#_Toc351332866)

[Требования к окружению 4](#_Toc351332867)

[Требования к аппаратному обеспечению 4](#_Toc351332868)

[Требования к программному обеспечению 4](#_Toc351332869)

[Требования к пользователям 4](#_Toc351332870)

[Архитектура системы (Общие требования) 4](#_Toc351332871)

[Спецификация данных 4](#_Toc351332872)

[Описание формата или структуры данных 4](#_Toc351332873)

[Функциональные требования 6](#_Toc351332874)

[Требования к интерфейсу 6](#_Toc351332875)

[Проект 6](#_Toc351332876)

[Средства реализации 6](#_Toc351332877)

[Структуры данных 6](#_Toc351332878)

[Модули и алгоритмы 8](#_Toc351332879)

[Стандарт кодирования 8](#_Toc351332880)

[Проект интерфейса 8](#_Toc351332881)

[Заключение 10](#_Toc351332882)

[Список литературы 10](#_Toc351332883)

# Аннотация

В настоящем документе описаны основные этапы создания векторного графического редактора от постановки задачи до непосредственной реализации оной в соответствии с документацией [1],[2].

# Введение

Для создания и редактирования изображений в настоящее время в основном используется графические редакторы.

Целью практической работы является разработка программы «Графический редактор», которая позволяет пользователю просматривать и редактировать графические файлы.

Из поставленной цели вытекают следующие задачи, которые необходимо решить для разработки данного приложения:

* изучить методическую литературу по технологии разработки программных продуктов;
* изучить теоретические аспекты среды программирования Lazarus;
* подготовить дизайн-проект приложения;
* организовать удобную навигацию в программе.

## Описание предметной области

Программный продукт может быть использован для создания различных графических изображений, при помощи заданного набора примитивов, и позволяет преобразовывать полученное изображение. Программа разработана в учебных целях и не претендует на использование в качестве полноценного редактора.

## Неформальная постановка задачи

Программный продукт предназначен для работы с изображением и поэтому должен позволять выполнять следующие действия:

* Работать с графическими файлами;
* Создавать новые графические файлы;
* Рисовать основные геометрические примитивы;
* Иметь дополнительные возможности редактирования;
* Сохранять графические файлы.

## План работ

17.11.2012 – Добавить простые инструменты;

24.11.2012 – Добавить возможность регистрации инструментов;

01.12.2012 – Добавить возможность редактировать параметры фигур;

08.12.2012 – Додавить палитру;

15.12.2012 – Добавить масштабирование и прокрутку;

22.12.2012 – Добавить возможность редактирование формы фигур;

29.12.2012 – Добавить возможность сохранения и загрузки.

# Требования к окружению

## Требования к аппаратному обеспечению

|  |  |  |
| --- | --- | --- |
|  | **Минимальные** | **Рекомендуемые** |
| **Процессор** | 233 МГц | 300 МГц или выше |
| **Оперативная память** | 64 МБ | 128 МБ или выше |
| **Свободное место на жёстком диске** | 30MB или больше | |

## Требования к программному обеспечению

На компьютере пользователя должна быть установлена любая операционная система линейки Windows NT.

## Требования к пользователям

Пользователь должен иметь опыт использования прикладными программами на компьютере под управлением, требуемой для работы приложения, операционной системы.

# Архитектура системы (Общие требования)

Программа распространяется в качестве exe файла, и не требует каких - либо сторонних библиотек или же программ.

# Спецификация данных

Для сохранения изображения и его последующей загрузке используется формат файлов с расширением .lpf.

## Описание формата или структуры данных

Формат lpf:

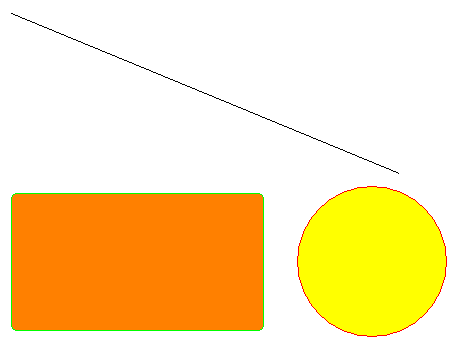


Рис. Графическое изображение, описанное в примере.

Созданные файлы хранятся в формате lpf. В примере ниже описана структура сохранения Рис. в формате векторного редактора.

Vector Editor by LifePack

Scale = 1

OffsetX = 0

OffsetY = 0

FilledAreaLeft = 0

FilledAreaTop = 0

FilledAreaRight = 751

FilledAreaBottom = 688

figure = TEllipse

PointsArr = |368/254|,|518/405|

PC\_PenCol = 255

PW\_PenWid = 0

PS\_PenSty = 0

SelectedFigure = DrawFigure

BC\_BrushCol = 65535

BS\_BrushSty = 0

end

figure = TRoundRect

PointsArr = |82/261|,|335/399|

PC\_PenCol = 65280

PW\_PenWid = 0

PS\_PenSty = 0

SelectedFigure = DrawFigure

BC\_BrushCol = 33023

BS\_BrushSty = 0

PW\_WidthEll = 10

PW\_HeightEll = 10

end

figure = TLine

PointsArr = |82/81|,|470/241|

PC\_PenCol = 0

PW\_PenWid = 0

PS\_PenSty = 0

SelectedFigure = DrawFigure

end

# Функциональные требования

Система должна позволять пользователю:

* Работать с графическими файлами;
* Создавать новые графические файлы;
* Рисовать основные геометрические примитивы;
* Сохранять графические файлы;
* Обрабатывать ошибки файлов;
* Редактировать свойства фигуры;
* Редактировать свойства нескольких фигур одновременно;
* Копировать, удалять, вставлять фигуры;
* Перемещать фигуры по холсту;
* Редактировать вершины фигур.

# Требования к интерфейсу

Система должна:

* обладать дружественным интерфейсом;
* иметь прозрачную архитектуру для пользователя, пользователю должна быть предоставлена возможность менять вид программы по своему усмотрению;
* Функциональные элементы меню должны находится в левом и верхнем углах экрана, т.к. там они легче всего воспринимаются;
* Пункты меню должны отражать основные моменты пользования программой.

# Проект

Продукт состоит из 13 модулей;

2-х форм;

1 одного проекта;

папки Images.

## Средства реализации

По требованию заказчика выбран язык Delphi, а работа проводилась в среде разработки Lazarus.

## Структуры данных

Основная диаграмма классов - модульUfigures.

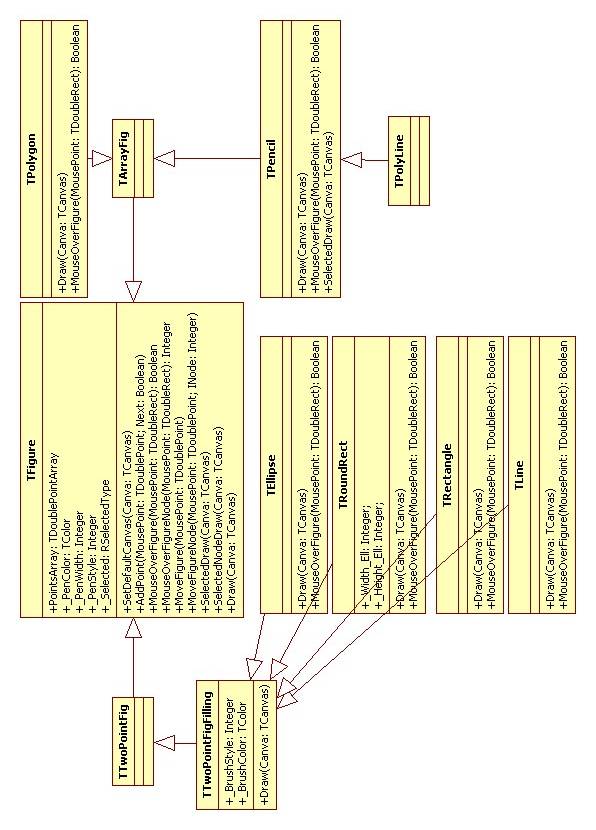


Рис. Диаграмма классов

## Модули и алгоритмы

Uabout – Форма “О программе”;

Uedialoglist – Форма “Окно об ошибках”;

Uedits – Модуль работы с редакторами свойств;

Uditsstatic– Модуль работы с редакторами свойств для статических объектов;

Uexceptions – Описывает типы ошибок;

Ufigures– Модуль описывает и реализовывает фигуры;

Ugraph– Модуль реализовывает работу с некоторыми переменными;

Umain– Главная форма;

Unotvisualtools– Не визуальные инструменты;

Urtti– Работа с RTTI;

Usaveloadtolpf– Загрузка и сохранение информации;

Utransformation– Трансформирование;

Uvisualtools– Визуальные инструменты.

## Стандарт кодирования

Использован стандарт кодирования, принятый сообществом Lazarus.

## Проект интерфейса

Основные элементы интерфейса:

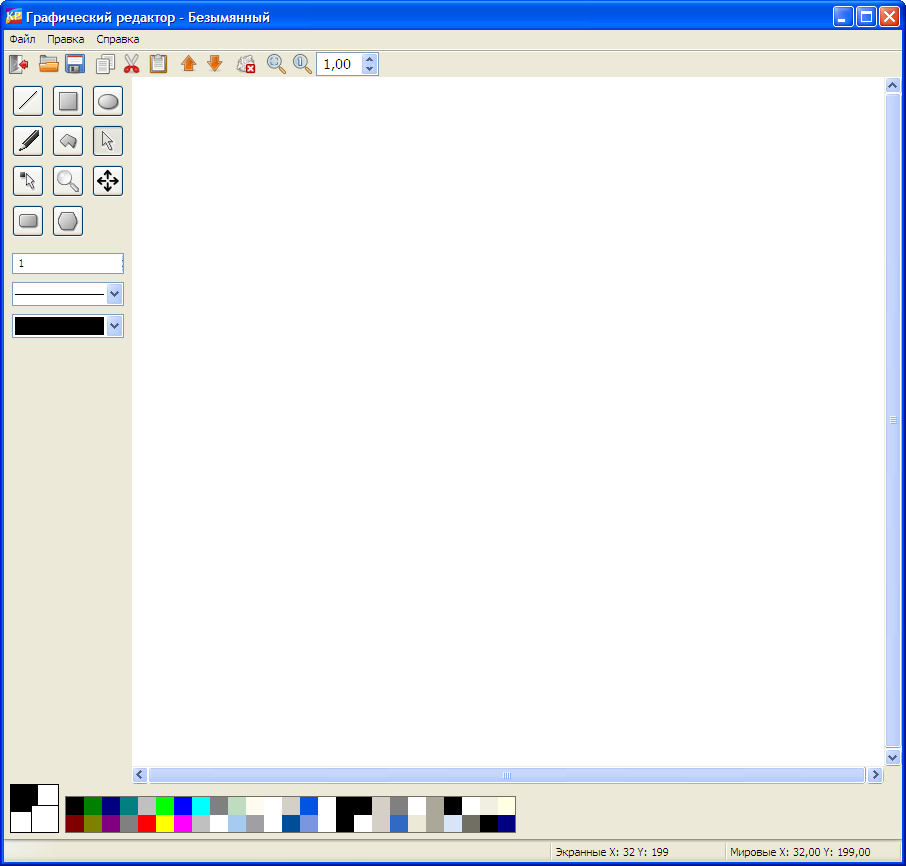


Рис. Основное окно программы



Рис. Набор инструментов

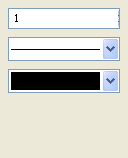


Рис. Редактор свойств

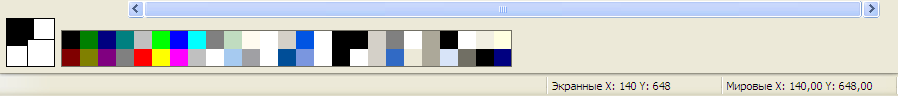


Рис. Редактор цветов

# Заключение

Таким образом, в процессе работы мною был сделан графический векторный редактор, за который я получил зачет, а также повысил свои навыки программирования.

# Список литературы

1. *Кленин А. С.* Методические указания по подготовке и защите отчётов, ДВГУ, 2009,   
    <http://imcs.dvgu.ru/ru/courses/repplan>
2. *Кленин А. С.* Технология программирования: программа курса, ДВГУ, 2003,   
    [http://imcs.dvgu.ru/ru/courses/progtech](http://imcs.dvgu.ru/courses/progtech)