Министерство образования и науки Украины

Пояснительная записка

к курсовому проекту на тему "Разработка графического редактора"

по курсу "Объектно-ориентированное программирование"

2005

**Содержание**

Введение

Постановка задачи

Описание проектного решения

Программная реализация

Руководство пользователя

Тестирование программы

Критерии качества разработки ПС

Заключение

Библиографический список

**Введение**

В записке приведено описание программы, разработанной в соответствии с заданием на курсовое проектирование по теме "Разработка графического редактора" по дисциплине "Объектно-ориентированное программирование". Программа предназначена для создания и редактирования графических файлов при помощи различных инструментов рисования.

При объектно-ориентированном программировании необходимо придерживаться следующих принципов:

- Действие в объектно-ориентированном программировании инициализируется посредством сообщения объекту. Сообщение содержит запрос на некоторое действие. В качестве реакции на сообщение получатель запускает некоторый метод, чтобы удовлетворить принятый запрос.

- Иерархия и классификация, т.е. все объекты одного класса используют одни методы в ответ на одинаковые сообщения.

Наследование – способность создавать новые типы повторно,

используя описание уже существующих типов

- Полиморфизм имеется в виду, что объекты реагируют на одно и тоже сообщение строго специфическим для них образом.

- Инкапсуляция т.е рассматривая объекты как высший уровень абстракции данных и деление его на части, получаем, что отношение частей к целому и взаимоотношения между частями становятся понятнее тогда, когда все содержится вместе как единое целое.

Разработка редактора производилась в Delphi 5, т.к. данная среда в полной мере отражает все достоинства объектно-ориентированного подхода, позволяет создавать новые классы и использовать библиотеку уже созданных стандартных классов Delphi 5. Среда удобна для разработки. Интерфейсы, создаваемые Delphi, наглядны и просты для использования.

Результаты тестирования подтверждают, что программа правильно выполняет все операции создания, редактирования, сохранения и пересохранения графических файлов.

Требованием к аппаратному и программному обеспечению является наличие любого IBM-совместимого компьютера с накопителем на гибких и жестких магнитных дисках и достаточным количеством оперативной памяти. На компьютере должна быть установлена операционная система, совместимая с Windows 3.x.

**1 Постановка задачи**

Цель настоящей курсовой работы – проектирование и программная реализация системы, которая в соответствии с заданным вариантом реализует процессы создания и взаимодействия группы объектов.

Архитектура системы определяет ее разбиение на модули, задает контекст, в рамках которого принимаются проектные решения на следующих этапах разработки. Приняв решения о структуре системы в целом, разработчик системы производит ее разбиение на относительно независимые в реализации части (модули).

По варианту задания на базе объектного подхода необходимо спроектировать и разработать модель системы, работающей с графикой по заданному плану. Постановка задачи подразумевает разработку многофункционального графического редактора со стандартными функциями удаления (стирания) фигур, рисования, вставки готовых образов, наложения фигур, растяжения (сжатия) и др. Система должна быть выдержана в едином стиле и снабжена справочной подсистемой с указанием возможностей.

**2 Описание проектного решения**

Целью данной работы является разработка простейшего графического редактора с помощью объектно-ориентированного языка программирования, с использованием его графических функций и методов.

Реализованный графический редактор позволяет осуществлять работу с как уже имеющимися изображениями, так и создавать собственные с помощью различных возможностей таких как:

создание векторных объектов (круга, квадрата, скруглённого квадрата, ромба) и их различные модификации (растягивание) в эллипс, прямоугольник и прямоугольник со скруглёнными вершинами соответственно. Возможно изменение цвета фигур, толщины линии их образа;

создание рисунков с помощью линий (карандаша), кисти различной толщины, спрея (объект, который создаёт эффект напыления пикселей на определённую небольшую область рисунка);

вывод текста на редактируемый рисунок;

стирание любой нарисованной картинки;

а также реализовано копирование, вставка а также сохранение картинки в формате \*.bmp.

Объектно-ориентированный анализ и проектирование нельзя определить ни как проектирование сверху вниз, ни как проектирование снизу вверх; его можно скорее назвать «возвратным проектированием», что подразумевает ступенчатый процесс разработки системы с постепенной модификацией различных, но, тем не менее, согласованных между собой логических и физических представлений о системе в целом.

Созданная система в процессе деятельности может быть представлена рядом состояний, которые осуществляют те или иные действия. В системе можно выделить некоторое начальное состояние и конечное, завершающее работу состояние. Анализ системы следует начать с жизненного цикла - рисунок 2.1.

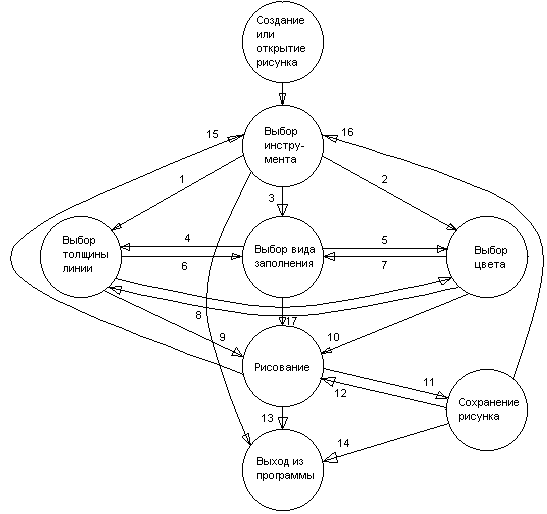


Рисунок 2.1 - Жизненный цикл системы

На рисунке 2.1 приняты следующие обозначения:

1 – выбран инструмент линия, карандаш, геометрическая фигура или текст;

2 - выбран инструмент цвета;

3 - выбран инструмент геометрическая фигура;

4,5,6,7,8,9 - выбран инструмент линия, карандаш, геометрическая фигура или текст;

10 - выбрана определенная толщина линии;

11 - некоторый этап рисования завершен, необходимо сохранить рисунок;

12 - рисунок сохранен, необходимо продолжить рисование;

13 - редактирование или рисование завершено, выход из программы;

14 - отредактированный или нарисованный рисунок сохранён, завершение работы.

15 - выбор другого инструмента, редактирование;

16 - рисунок сохранён, редактирование;

17 - выбран определённый тип заполнения;

Из рисунка 2.1 видно, что начальным этапом является создание нового или открытие уже существующего рисунка. Когда рисунок создан, можно осуществлять его редактирование, сохранение под новым именем либо в уже созданном файле. Для редактирования разработан ряд инструментов, осуществляющих различные функции:

карандаш;

ластик;

заливка;

распылитель;

рисование прямоугольников;

рисование скругленных прямоугольников;

рисование окружностей;

рисование линий;

рисование пирамиды.

рисование ромбов;

ввод текста;

Существуют также инструмент выбора цвета инструмента и цвета заливки.

В процессе редактирования происходит взаимодействие объектов, отвечающих за стиль рисования тех или иных фигур.

Рисунок 2.2 - Информационная модель взаимодействия объектов Ф и З

***2.Кнопка фигура (Ф)***

* имя

***1.Кнопка заполнение(З)***

* имя

R1

1 : M

Рисунок 2.3 - Информационная модель взаимодействия объектов Ф, З, F, G

***4. Picture.Graphic(G)***

* Ширина
* Высота

***3. MainForm (F)***

* Заголовок
* Ширина
* Высота

R2

1 : 1

***1.Кнопка заполнение(З)***

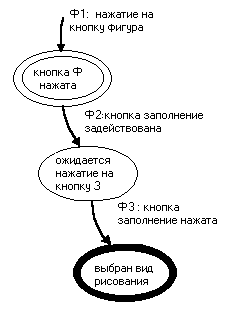
* имя

***2. Кнопка фигура (Ф)***

* имя

Взаимодействие объектов кнопка “фигура” и кнопка “заполнение” показано на диаграмме переходов состояний (рисунок 2.4).

Рисунок 2.4 - Модели состояний для кнопок “фигура” и “заполнение”



Так как большинство действий в системе производится в соответствии с событиями от мыши, то основные входные данные для системы – это текущие координаты и сами события MouseDown, MouseUp, MouseMove.

Диаграмма потоков данных приведена на рисунке 2.5.

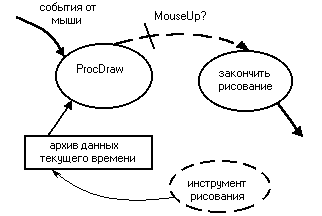


Рисунок 2.5 - Диаграмма потоков данных

Так как не было необходимости в проекте создавать специальные классы для реализации объектов, то в иерархии классов будут приведены основные классы среды Delphi, используемые для

реализации основных возможностей проекта и их описание.

Иерархия используемых графических классов изображена на рисунке 2.6.

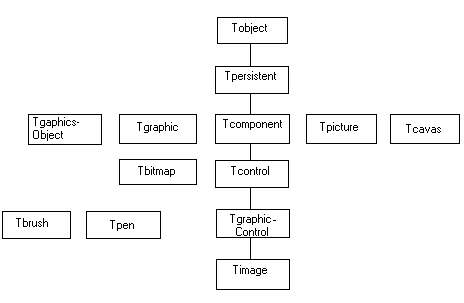


Рисунок 2.6 - Иерархия классов

**3. Программная реализация**

Согласно условию задачи, выполнены следующие пункты:

Используя основные классы среды Delphi, были реализованы основные возможности редактора для создания рисунков;

2) Редактирование различных графических изображений;

Запоминание картинок в файлах формата \*.bmp.;

Реализован удобный пользовательский интерфейс, который каждый пользователь может корректировать по своему желанию.

При решении задачи построения графического редактора был использован объектно-ориентированный подход. Его преимущество перед структурным подходом заключается в следующем:

Объект моделирования и действия над объектом разрабатываются совместно, что требует более тщательного анализа поведения и характеристик объекта, тем самым, повышая качество моделирования.

Объект моделирования и методы работы с ним объединяются в одной структуре – классе, связывая характеристики объекта с действиями над ними (инкапсуляция), что дает возможность рассматривать объект не как хранилище некоторых свойств, а как функционально полную единицу.

Достигается определенный уровень абстракции, что помогает при разработке сложных программ, т.к. позволяет идти от простого к сложному, от общего к частному, а также приводит к такому мощному средству создания классов, как наследование.

Возможность позднего связывания приводит к более гибкой манипуляции данными, т.к. позволяет использовать одни и те же действия при работе с объектами, которые отличаются некоторыми характеристиками, но имеют общие свойства (полиморфизм).

Данная программа написана в среде Delphi 5 в соответствии с основными принципами объектно-ориентированного программирования; объектный стиль программирования фиксирует поведение реального мира таким способом, при котором детали его реализации скрыты.

При проектировании сложной программной системы необходимо составлять ее из небольших подсистем, каждую из которых можно отладить независимо от других. В этом случае при разработке любого уровня системы нам необходимо будет одновременно держать в уме информацию лишь о немногих ее частях. Существуют два способа декомпозиции сложной программной системы:

- алгоритмическая декомпозиция - разделение алгоритмов по методу «сверху вниз» и соответствует структурному проектированию, где каждый модуль системы выполняет один из важных этапов общего процесса. Результат структурного проектирования можно представить в виде дерева подпрограмм.

- объектно-ориентированная декомпозиция - в качестве критерия декомпозиции системы используется принадлежность ее элементов к различным абстракциям данной предметной области.

Программа состоит из основного модуля Unit1, осуществляющего синхронизацию процесса обработки данных и взаимодействие конкретных экземпляров конкретных объектов; подключаемого модуля Unit2, в котором содержится класс Form2, с помощью которого задается размер рисунка; подключаемого модуля Unit3, в котором содержится класс Form3, с помощью которого выводится окно “О программе…”.

Модуль Unit1 просто дополненять и модифицировать. В частности, не представляет никаких затруднений добавить новые виды инструментов, добавив их в тип TDrawingTool и задав для них соответствующие действия в процедуре DrawShape.

DrawShape является наиболее функцмональной процедурой и управляет работой практически всей системы. События происходят в зависимости от выбранного инструмента( это отражается в переменной Drawingtool). События OnMouseDown, OnMouseMove, OnMouseUp вызывают процедуру и далее она производит все действия. При нажатии кнопки мыши(OnMouseDown) начинается прорисовка графического объекта, при движении мыши(OnMouseMove) объект прорисовывается в соответствии с изменением координат указателя мыши, когда кнопку отпускают(OnMouseUp) прорисовка заканчивается.

Процедура DrawShape описана в разделе public основного класса TForm1, что позволяет использовать ее в любом модуле программы

Описание метода TForm1.DrawShape:

procedure TForm1.DrawShape(TopLeft, BottomRight: TPoint; AMode: TPenMode);

var i:integer;

begin

with image1.Canvas do

begin

Pen.Mode :=amode;

case DrawingTool of

dtHandle: //КАРАНДАШ

begin

Pen.Mode :=pmcopy;

image1.Canvas.Pixels[topleft.x,topleft.y]:=colordialog1.Color;

image1.Canvas.LineTo(BottomRight.X, BottomRight.Y);

end;

dtLine: //РИСОВАНИЕ ЛИНИИ

begin

image1.Canvas.MoveTo(TopLeft.X, TopLeft.Y);

image1.Canvas.LineTo(BottomRight.X, BottomRight.Y);

end;

dtPolyline: //ПИРАМИДА

begin

image1.Canvas.Polyline([origin, movept, point(spinedit1.Value, spinedit2.Value)]);

end;

dtSpray: //РАСПЫЛИТЕЛЬ

begin

Pen.Mode :=pmcopy;

for i:=1 to 20 do

begin

image1.Canvas.Pixels[BottomRight.X+random(20),

BottomRight.Y+random(20)]:=colordialog1.Color;

end;

end;

dtFonty: //ВЫВОД ТЕКСТА

begin

image1.Canvas.TextOut(bottomright.x,bottomright.y,edit1.Text);

end;

dtRomb: //РОМБ

begin

image1.Canvas.MoveTo(topleft.x+((bottomright.x-topleft.x) div 2),topleft.y);

image1.Canvas.LineTo(bottomright.x, bottomright.y-((bottomright.y-topleft.y) div 2));

image1.Canvas.MoveTo(topleft.x+((bottomright.x-topleft.x) div 2),topleft.y);

image1.Canvas.LineTo(topleft.x, topleft.y+((bottomright.y-topleft.y) div 2));

image1.Canvas.LineTo(topleft.x+((bottomright.x-topleft.x) div 2),bottomright.y);

image1.Canvas.LineTo(bottomright.x, bottomright.y-((bottomright.y-topleft.y) div 2));

end;

dtRectangle: //РИСОВАНИЕ ПРЯМОУГОЛЬНИКА

image1.Canvas.Rectangle(TopLeft.X, TopLeft.Y, BottomRight.X,

BottomRight.Y);

dtEllipse: //РИСОВАНИЕ ЭЛЛИПСА

image1.Canvas.Ellipse(Topleft.X, TopLeft.Y, BottomRight.X,

BottomRight.Y);

dtGum: //ЛАСТИК

begin

Pen.Mode :=pmcopy;

pen.Color:=clwhite;

pen.Width:=10;

image1.Canvas.Pixels[topleft.x,topleft.y]:= clwhite;

image1.Canvas.LineTo(BottomRight.X, BottomRight.Y);

end;

dtBottle: //ЗАЛИВКА

image1.Canvas.FloodFill(topleft.x,topleft.y,image1.Canvas.Pixels[topleft.x,topleft.y],

fssurface);

dtRoundRect: //РИСОВАНИЕ СКРУГЛЁННОГО ПРЯМОУГОЛЬНИКА

image1.Canvas.RoundRect(TopLeft.X, TopLeft.Y, BottomRight.X, BottomRight.Y, (TopLeft.X - BottomRight.X) div 2,

(TopLeft.Y - BottomRight.Y) div 2);

end;

end;

end;

Полный текст всех модулей приведен в Приложении А.

Были использованы методы класса Tpicture:

LoadFromFile - загрузка изображения из файла

SaveToFile - сохранение изображения в файле,

Эти методы по умолчанию выводят диалоговые окна с помощью которых легко открывать и сохранять графические файлы.

При создании рисунка в программе используется объект Bitmap: Tbitmap, который создается конструктором Bitmap.Create, а затем уничтожается при помощи процедуры Bitmap.Free. Bitmap.Free автоматически вызывает деструктор объекта, если объект инициализирован, а если объект не создан, то деструктор не вызывается, что обеспечивает корректную работу программы.

При написании программы были соблюдены основные рекомендации стандартного оформления текстов программ. Соответствие стандартам заключается в следующем:

Составление идентификаторов производится так, чтобы по ним было понятно их основное предназначение (пример: и т.д.).

Используются «форматирующие пробелы» для повышения удобочитаемости программы.

Программа разбивается на несколько модулей, каждый из которых имеет свое функциональное назначение, при этом реализуется повторная используемость.

Использование объектно-ориентированного подхода обеспечивает расширяемость.

Устойчивость к неправильным входным данным – невозможность задания неправильных данных.

**4 Руководство пользователя**

Программа поставляется в виде исполняемого ЕХЕ файла, запускается под управлением ОС Windows 9X, не требует большого количества системных ресурсов. Имеет удобный пользовательский интерфейс и достаточно широкие возможности для рисования, просмотра и редактирования изображений.

Кнопки выбора инструментов снабжены пиктограммами, иллюстрирующими данный графический инструмент. Помимо этого, при нажатии на кнопку, т. е. при выборе какого-либо инструмента изменяется вид курсора. Редактор снабжен системой всплывающих подсказок.

В заголовке главного окна содержится информация о редактируемом рисунке:

имя рисунка,

путь,

размер рисунка в пикселях.

Главное меню содержит подпункты:

- создать,

открыть,

сохранить,

сохранить как,

выход.

Каждый подпункт снабжен стандартной комбинацией “горячих” клавиш, которые вызывают то или иное действие:

создать Ctrl+N,

открыть Ctrl+O,

сохранить Shift+F12,

сохранить как Ctrl+S,

выход F10.

Пункт главного меню правка содержит команды для работы с буфером:

вырезать Shift+Del,

копировать Ctrl+Insert,

вставить Shift+Incert.

Подпункт “О программе…” - содержит справку о данной программе.

При создании рисунка на экране появляется вспомогательное окно для установления требуемых размеров. При открытии или сохранении файла в стандартном диалоговом окне пользователь выбирает нужные ему данные. Для инструментов прямоугольник, прямоугольник со скругленными углами, эллипс, ромб можно определить вид заполнения внутреннего пространства, толщину контура, цвет. Для инструментов карандаш, линия, пирамида можно осуществлять выбор толщины линии, цвет. Для инструмента распылитель можно выбирать цвет. Выбор цвета сводится к выбору цвета из стандартной шестнадцатицветной палитры, либо к выбору любого из оттенков 16-битового цвета. Редактор имеет инструменты выбора цвета контура и цвета внутреннего заполнения.

В редакторе можно с большими удобствами просматривать и пересохранять существующие изображения. Эта фунция вызывается при нажатии кнопки “Просмотр и выбор”, которая делает видимой панель на которой размещены элементы “Выбор диска”, “Выбор директории”, “Выбор типа файла”, “Выбор файла”.

Основные элементы панели инструментов выполняют следующие действия:

карандаш – рисование с учётом выбранного цвета, и толщины линии;

ластик - старание любого вида рисунка

заливка – заливка выбранной области, с учётом выбранного цвета;

спрей – распыление точек в определённой области, с учётом выбранного цвета;

окружность – рисование различных окружностей и эллипсов, с учётом выбранного цвета, вида заполнения;

прямоугольник – рисование квадратов и прямоугольников, с учётом выбранного цвета, вида заполнения;

скруглённый прямоугольник – рисование квадратов и прямоугольников со скруглёнными вершинами, с учётом выбранного цвета, вида заполнения;

ромб – рисование ромбов, с учётом выбранного цвета, вида заполнения;

пирамида – рисование пирамиды, с вершиной задаваемой координатами Х и Y в правом нижнем углу редактора, с учётом выбранного цвета;

вывод текста – вывод на рисунок текста выбранного шрифта и цвета, в выбранную точку;

**5 Тестирование программы**

Самое реалистичное тестирование программы проходило при её отладке, при тестировании проверялись функциональные возможности подсистем и отдельных инструментов. Затем осуществлялся сбор всех инструментов и производилось системное тестирование. С целью тестирования в данном редакторе создавались графические изображения при помощи многократного использования различных инструментов. Проверялась работа редактора при изменении цвета, толщины и вида линий, а также при изменении стиля заполнения различных фигур.

В редакторе можно произвести коррекцию уже созданных файлов, содержащих растровые изображения.

Тестирование показало, что программа должным образом реагирует на события от мыши, обрабатывая их определенным способом и производя заданные действия.

**6. Критерии качества разработки программной системы**

Критерии качества разработки программной системы:

соответствие заданию и выбранным ограничениям;

эффективное использование объектного подхода;

стиль программирования;

стилизация;

дизайн.

Поскольку по сути варианты заданий на курсовое проектирование представляют собой лишь канву тех или иных описаний поведения одиночных объектов или их групп, то данная работа соответствует выбранным ограничениям, и отражает суть задания, несколько расширяя его.

ООП-подход лишен недостатков структурного программирования, которое не позволяет выделять абстракции, не имеет достаточных средств для организации параллелизма, следовательно, затрудняет создание достаточно сложных систем. ООП-подход, кроме того, экономит мышление разработчика – не нужно держать в памяти большой объем информации о структуре программы.

Объектно-ориентированное программирование базируется на наследовании, инкапсуляции и полиморфизме. Реализуя заданный вариант, удалось обойтись минимальным количеством классов и объектов и их вложенности.

Созданный редактор имеет понятный и простой в использовании интерфейс, стандартное для Windows оформление.

**Заключение**

В ходе выполнения курсового проекта закреплены понятие объектно-ориентированного программирования, объектно-ориентированного анализа и объектно-ориентированного подхода, навыки по программированию в объектно ориентированной среде.

Программа, описанная в курсовом проекте разработана в соответствии с постановкой задачи на курсовое проектирование по теме “Разработка графического редактора” по дисциплине “Объектно-ориентированное программирование”.

Интерфейс созданной программы удобен, прост, наглядно отображает ее возможности. Редактор снабжен системой всплывающих подсказок. Главное меню редактора содержит команды работы с файлами, команды занесения рисунка в буфер и вывода рисунка из буфера.

Тестирование подтвердило, что программа корректно выполняет обработку данных и демонстрацию результатов.

Всё это свидетельствует о работоспособности программы и позволяет сделать вывод о ее пригодности для создания и редактирования графических изображений.

**Библиографический список**

1. Доронина Ю.В. Конспект курса лекций по дисциплине “Объектно-ориентированное программирование”.

2. Фаронов В.В. Delphi 6. Учебный курс. – М.: Издатель Молгачева С.В., 2001. – 672 с., ил.

3. Методические указания к курсовой работе по дисциплине “Объектно-ориентированное программирование” для студентов специальности 7.080.401 – “Компьютеризированные системы обработки информации и управления” /Сост. Ю.В. Доронина, А.Н. Синицын, 2001 (электронная версия).

4. Дарахвелидзе П.Г., Марков Е.П. Delphi 4 – Среда визуального программирования. БХВ – Санкт-Петербург, 1999. – 816 с.

5. Озеров В. – Советы по Delphi (электронная книга).