Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего

образования

НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ИРКУТСКИЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

|  |
| --- |
| Институт информационных технологий и анализа данных |
| наименование института |

ОТЧЕТ

к лабораторной работе по дисциплине:

|  |
| --- |
| **Технологии разработки программных комплексов** |
| Тестирование |

наименование темы

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Выполнил студент группы | ИСТб-21-1 |  |  |  | Соколов Э.Д. |
|  | шифр группы |  | подпись |  | Фамилия И.О. |
| Проверил |  |  |  |  | Бахвалова З.А. |
|  | должность |  | подпись |  | Фамилия И.О. |

Иркутск 2024 г.

**Содержание**

[1 Постановка задачи 3](#_Toc161907890)

[2 Требования 3](#_Toc161907891)

[2.1 Функциональные требования 3](#_Toc161907892)

[2.2 Нефункциональные требования 3](#_Toc161907893)

[2.3 Требования к пользовательскому интерфейсу 3](#_Toc161907894)

[3 Структурный контроль 4](#_Toc161907895)

[4 Стратегия «белого ящика» 7](#_Toc161907911)

[5 Стратегия «чёрного ящика» 10](#_Toc161907912)

[6 Тестирование пользовательского интерфейса 15](#_Toc161907913)

1 Постановка задачи

Дан равносторонний шестиугольник, внутри которого вписана шестиконечная звезда, вершины которой опираются на стороны шестиугольника. Требуется рассчитать площадь заштрихованной области, изображённой на рисунке 1.1.

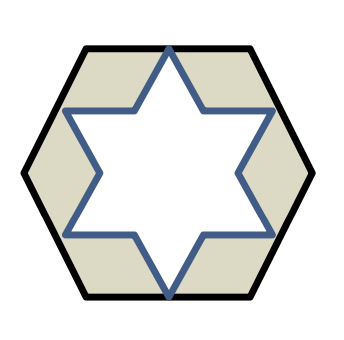


Рисунок 1.1 – Рисунок шестиугольника со вписанной шестиконечной звездой

2 Требования

Требуется создать приложения на языке JavaScript, которое будет рассчитывать площадь заштрихованной области.

2.1 Функциональные требования

Программа должна:

* строить геометрические фигуры и выводить их на экран,
* считать площадь заштрихованной области фигуры,
* изменять размеры фигуры в зависимости от введенных размеров
* Масштабировать полотно для отрисовки в зависимости от размеров окна

2.2 Нефункциональные требования

Программа должна быть реализована на платформе языка JavaScript с использованием Bootstrap для создания пользовательского интерфейса и Canvas для отрисовки фигур;

* 1. Требования к пользовательскому интерфейсу
* Для ввода значения размера фигур используется поле ввода напротив надписи: "Размер фигур (10-300), пиксели".
* Для запуска отрисовки фигур используется кнопка "Отрисовать".
* Результаты отрисовки выводятся автоматически под кнопкой "Отрисовать".
* Площадь красной зоны шестиугольника выводится под полем ввода размера фигур.
* Площадь синей зоны звезды выводится под площадью красной зоны.
* При изменении размера окна браузера автоматически пересчитывается размер холста и отрисовываются фигуры.
* На данный момент нет необходимости в переключении на другой тип расчета или вводе дополнительных параметров.

3 Структурный контроль

Таблица 1 – Тестирование программы с использованием структурного контроля

|  |  |  |  |
| --- | --- | --- | --- |
| **№ вопроса** | **Строки, подлежащие проверке** | **Результат проверки** | **Вывод** |
| 1.1 | 30-51 файла index.html | Все переменные инициализированы корректно | Все переменные инициализированы |
| 1.2 | 53-71 файла index.html | Все функции отрисовки фигур реализованы правильно | Отрисовка фигур работает корректно |
| 1.3 | 86-97 файла index.html | Проверка корректности ввода данных для размера фигур реализована верно | Доступны только допустимые значения для ввода размера фигур |
| 2.1 | 69-70 файла index.html | Программа выводит результаты расчета площади красной и синей зон с требуемой точностью | Расчеты соответствуют заданным требованиям точности |
| 3.1 | 78-81 файла index.html | Программа завершается только по нажатию на кнопку "Отрисовать" | Программа завершается по требованию пользователя |
| 4.1 | 15-18 файла index.html | Порядок ввода данных соответствует указанным требованиям | Порядок ввода данных отвечает требованиям |
| 4.2 | Нет | Нет неиспользуемых подпрограмм | Все подпрограммы используются корректно |
| 4.3 | Нет | Глобальные и локальные переменные имеют различные имена | Имена переменных уникальны и не пересекаются |

**Вывод:** В целом, структурный контроль – это, полезный метод тестирования программного обеспечения, который может помочь программисту создать качественный код и значительно сократить количество ошибок, но необходимо помнить, что это только один из способов тестирования и он не является абсолютным решением для обнаружения ошибок.

Плюсы структурного контроля как метода тестирования:

* Эффективное обнаружение ошибок: Использование метода структурного контроля может помочь обнаружить ошибки, связанные с некорректным использованием операторов и блоков кода.
* Повышение качества кода: Результатом использования метода структурного контроля может быть улучшение качества кода, так как этот подход помогает идентифицировать некорректно написанные операторы, циклы и блоки.
* Повышение понимания кода: Использование структурного контроля может помочь в понимании логики программного кода и улучшить его читаемость.

Минусы структурного контроля как метода тестирования:

* Ограниченная эффективность: Использование метода структурного контроля не гарантирует абсолютную надежность кода и может не помочь выявить определенные типы ошибок.
* Неэффективное использование времени: Использование метода структурного контроля может потребовать большого количества времени и ресурсов, которые могут быть необходимы для более важных задач.

4 Стратегия «белого ящика»

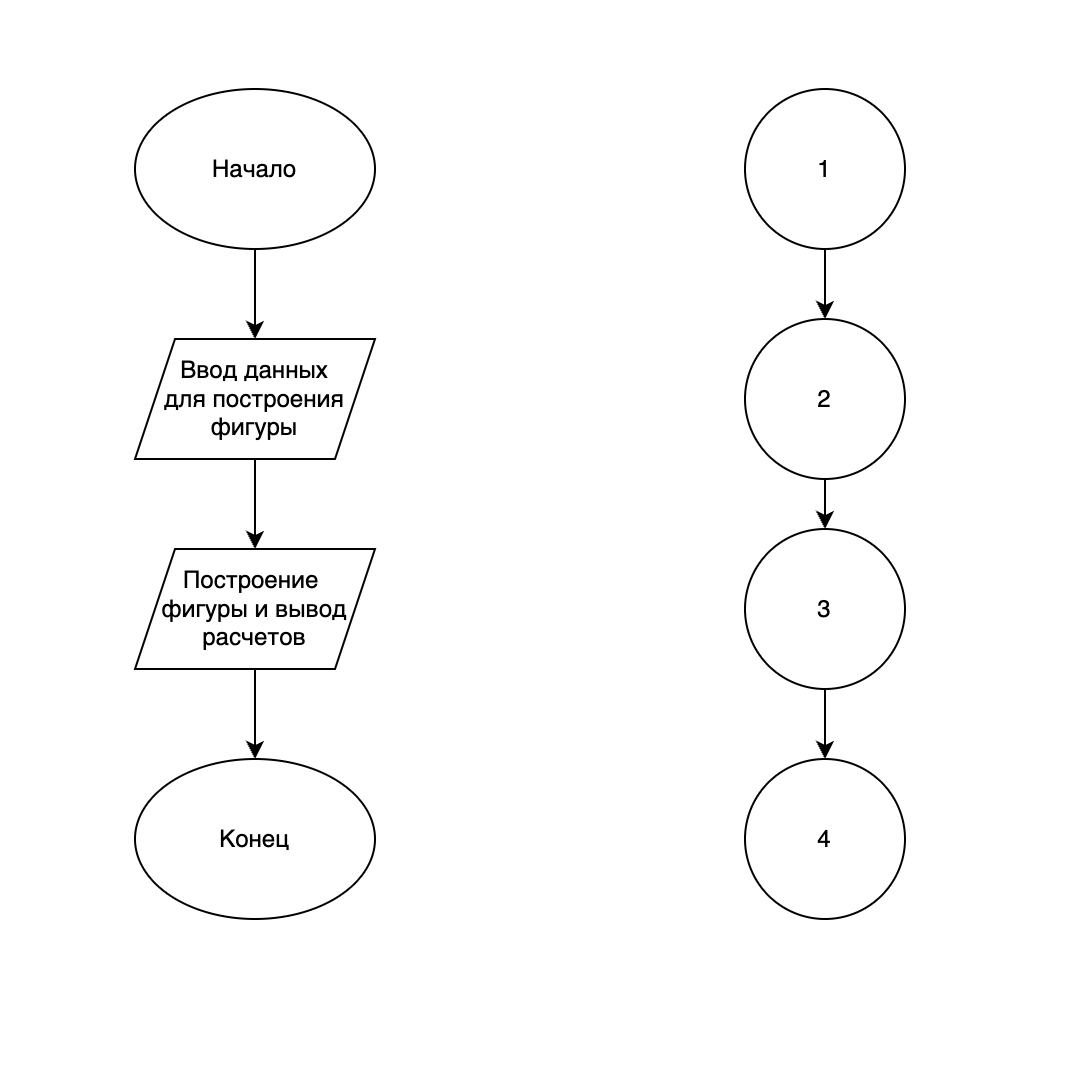


Рисунок 4.1 – Блок-схема решений при тестировании и её граф передачи управления

Таблица 4.1 – Тесты программы с использованием стратегии белого ящика

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Покрытие** | **Назначение теста** | **№ теста** | **Значение исходных данных** | **Ожидаемый результат** |
| Покрытие решений | Проверка оператора условия на принятие значения «истина», блока № 2 | 1 | RombLen = 50  Ang=45 | Путь 1 2 3 4 |
| Проверка оператора условия на принятие значения «ложь», блока № 2 | 2 | RombLen = 0  Ang=0 | Путь 1 2 (2) |
| Покрытие условий | Проверка пути, при значении «истина» блока №2 | 3 | RombLen = 50  Ang=45 | Путь 1 2 3 4 |
| Проверка пути, при значении «ложь» блока №2 | 4 | RombLen = 0  Ang=0 | Путь 1 2 (2) |
| Покрытие решений/условий | Проверка корректности выполнении блока №4 | 5 | RombLen = 50  Ang=45 | Путь 1 2 3 4 |
| Проверка корректности выполнения блока №3 | 6 | RombLen = 0  Ang=0 | Путь 1 2 (2) |
| Комбинаторное покрытие | Проверка принятия значения «истина» блоком №2 и блоком №4 корректного выполнения блока №6 | 7 | RombLen = 50  Ang=45 | Пусть 1 2 3 4 |

Таблица 4.2 – Тестирование с использованием стратегии белого ящика

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **№ теста** | **Входные данные** | **Ожидаемый результат** | **Результат** | **Вывод по покрытию** |
| 1 | RombLen = 0  Ang=0 | Путь 1 2 (2) | Путь 1 2 (2) | Операторы условий работают |
| 2 | RombLen = 50  Ang=45 | Пусть 1 2 3 4 | Пусть 1 2 3 4 |

**Вывод:** Подводя итоги тестирования программы стратегией белого ящика, в данной случае все методы покрывают все возможные варианты условий, связано это с небольшим количеством и не сложной структурой условий. Данное тестирование позволяет проверить логику и внутреннюю структуру программы. Однако даже у такой стратегии есть недостатки, в том, что даже в программе такого уровня, количество тестов сравнительно большое. Также эта стратегия не дает проверку соответствует программа полным требованиям.

5 Стратегия «чёрного ящика»

Таблица 5.1 – Классы эквивалентности

|  |  |  |
| --- | --- | --- |
| Входные условия | Правильные классы эквивалентности (№) | Неправильные классы эквивалентности (№) |
| Длина стороны ромба | 2<= RombLen  RombLen – число (1) | Строка символов (1),  Пустое значение (2),  Меньше 2 (3), |
| Основной угол ромба | 1<= Ang  Ang – число (1) | Строка символов (1),  Пустое значение (2),  Меньше 1 (3), |

Таблица 5.2 – Тест программы с использованием стратегии чёрного ящика

|  |  |  |  |
| --- | --- | --- | --- |
| № теста | Назначение теста | Значение исходных данных | Ожидаемый результат |
| 1 | Покрытие правильного класса эквивалентности №1 | RombLen = 50  Ang=45 | S = 786,02 |
| 2 | Покрытие неправильного класса эквивалентности №1 | RombLen = «dfb»  Ang=45  RombLen = 50  Ang=«dbf»  RombLen = «dbf»  Ang=«dfbd» | Невозможно ввести |
| 3 | Покрытие неправильного класса эквивалентности №2 | RombLen = «»  Ang=45  RombLen = 50  Ang=«»  RombLen =«»  Ang=«» | Невозможно ввести |
| 5 | Покрытие неправильного класса эквивалентности №3 | RombLen = -50  Ang=45 RombLen = 50  Ang=-45  RombLen = -50  Ang=-45 | Невозможно ввести |

Таблица 5.2.1 - Тестирование программы с использованием стратегии чёрного ящика

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № теста | Значение исходных данных | Ожидаемый результат | Результат | Вывод |
| 1 | RombLen = 50  Ang=45 | S = 786,02 | S = 786,02 | Правильный класс эквивалентности № 1 покрыт |
| 2 | RombLen = «dfb»  Ang=45 | Невозможно ввести | Невозможно ввести | Неправильный класс эквивалентности № 1 покрыт |
| 3 | RombLen = 50  Ang=«dbf» | Невозможно ввести | Невозможно ввести | Неправильный класс эквивалентности № 1 покрыт |
| 4 | RombLen = «dbf»  Ang=«dfbd» | Невозможно ввести | Невозможно ввести | Неправильный класс эквивалентности № 1 покрыт |
| 5 | RombLen = «»  Ang=45 | Невозможно ввести | Невозможно ввести | Неправильный класс эквивалентности № 2 покрыт |
| 6 | RombLen = 50  Ang=«» | Невозможно ввести | Невозможно ввести | Неправильный класс эквивалентности № 2 покрыт |
| 7 | RombLen =«»  Ang=«» | Невозможно ввести | Невозможно ввести | Неправильный класс эквивалентности № 2 покрыт |
| 8 | RombLen = -50  Ang=45 | Невозможно ввести | Невозможно ввести | Неправильный класс эквивалентности № 3 покрыт |
| 9 | RombLen = 50  Ang=-45 | Невозможно ввести | Невозможно ввести | Неправильный класс эквивалентности № 3 покрыт |
| 10 | RombLen = -50  Ang=-45 | Невозможно ввести | Невозможно ввести | Неправильный класс эквивалентности № 3 покрыт |

Таблица 5.3 – Тест граничных условий

|  |  |  |  |
| --- | --- | --- | --- |
| № теста | Назначение теста | Значение исходных данных | Ожидаемый результат |
| 1 | Проверить левую границу условия ввода длины стороны ромба | RombLen = 1 | Отсутствие построения |
| 2 | RombLen = 2 | Перестроение фигуры |
| 3 | Проверить левую границу условия ввода значения опорного угла | Ang =0 | Отсутствие построения |
| 4 | Ang = 1 | Перестроение фигуры |

Таблица 5.3.1 - Тестирование граничных условий

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № теста | Значение исходных данных | Ожидаемый результат | Результат | Вывод |
| 1 | RombLen = 1 | Отсутствие построения | Отсутствие построения | Левая граница условия ввода длины стороны ромба четко определена |
| 2 | RombLen = 3 | Перестроение фигуры | Перестроение фигуры |
| 3 | Ang = 1 | Перестроение фигуры | Перестроение фигуры | Левая граница условия ввода значения опорного угла четко определена |
| 4 | Ang = 0 | Отсутствие построения | Отсутствие построения |

Таблица 5.4 – Тесты ожидаемых ошибок

|  |  |  |  |
| --- | --- | --- | --- |
| № теста | Назначение | Исходные данные | Ожидаемый Результат |
| 7 | Тестирование расчёта площади при вводе длины стороны ромба = “пятьё” | RombLen = “пятьё” | Невозможно ввести |
| 8 | Тестирование расчёта площади при вводе длины стороны ромба = -21 | RombLen = -21 | Невозможно ввести проверьте ввод данных” |
| 9 | Тестирование расчёта площади при вводе длины стороны ромба = “ ” | RombLen = “ ” | Невозможно ввести |
| 10 | Тестирование расчёта площади при вводе длины стороны ромба = 65.74 | RombLen = 65.74 | Невозможно ввести |
| 11 | Тестирование расчёта площади при вводе длины стороны ромба меньше 1 | RombLen < 1 | Невозможно ввести |

Таблица 5.5 – Тестирование ожидаемых ошибок

|  |  |  |  |
| --- | --- | --- | --- |
| № теста | Назначение | Исходные данные | полученный Результат |
| 7 | Тестирование расчёта площади при вводе длины стороны ромба = “пятьё” | RombLen = “пятьё” | Невозможно ввести |
| 8 | Тестирование расчёта площади при вводе длины стороны ромба = -21 | RombLen = -21 | Невозможно ввести |
| 9 | Тестирование расчёта площади при вводе длины стороны ромба = “ ” | RombLen = “ ” | Невозможно ввести |
| 10 | Тестирование расчёта площади при вводе длины стороны ромба = 65.74 | RombLen = 65.74 | Невозможно ввести |
| 11 | Тестирование расчёта площади при вводе длины стороны ромба меньше 1 | RombLen < 1 | Невозможно ввести |

**Вывод:** Подводя итоги тестирования программы стратегией черного ящика, можно отметить возможность получения понимания поведения программы при использовании ее пользователем, а также узнать скрытые несоответствия с требованиями. Данная стратегия позволяет сразу понять соответствует ли она функциональным требованиям. Недостатками является возможность пропуска границ и переходов, которые не очевидны из спецификации.

6 Тестирование пользовательского интерфейса

Таблица 6.1 – Тестирование пользовательского интерфейса программы

|  |  |  |  |
| --- | --- | --- | --- |
| **№**  **теста** | **Действие** | **Реакция системы** | **Результат** |
|  | Ввод в поле ввода напротив надписи «Длина стороны (макс. 300)» | Формирования рисунка фигуры на основании введенных данных,  Расчет площади заштрихованной области | Изменение рисунка построенной фигуры,  Напротив, надписи: «Sp-Sk» изменение числа |
|  | Ввод в поле ввода напротив надписи «Опорный угол ромба(макс. 179)» | Формирования рисунка фигуры на основании введенных данных,  Расчет площади заштрихованной области | Изменение рисунка построенной фигуры,  Напротив, надписи: «Sp-Sk» изменение числа |
|  | Измените размеров окна варианта расчета | Формирования рисунка фигуры на основании введенных данных,  Расчет площади заштрихованной области | Изменение рисунка построенной фигуры,  Напротив, надписи: «Sp-Sk» изменение числа |

Таблица 6.2 – Тесты функциональности пользовательского интерфейса программы

|  |  |  |  |
| --- | --- | --- | --- |
| **№**  **теста** | **Функция** | **Действие** | **Реакция системы** |
| 1 | Построение фигуры | Введите значение в поле ввода длина радиуса окружности = 10 | В поле для отображения фигуры, появятся ромб с длиной стороны 10, с вписанной окружностью |
| 2 | Расчёт площади построенной фигуры | Введите значение в поле ввода длина радиуса окружности = 10 | В поле отображения площади, изменяется значение площади на 31.44 |
| 3 | Изменение типа вывода | Нажатие на кнопку теория | В поле для отображения фигуры, появится справочная информация по предметной области |

Таблица 6.2.1 – Тестирование функциональности пользовательского интерфейса программы

|  |  |  |  |
| --- | --- | --- | --- |
| **№**  **теста** | Входные данные | Ожидаемый результат | **Результат** |
| 1 | Введите значение в поле ввода длина радиуса окружности = 10 | Верно | Верно |
| 2 | Введите значение в поле ввода длина радиуса окружности = 10 | Верно | Верно |
| 3 | Нажатие на кнопку теория | Верно | Не верно |

**7 Листинг кода**

**unit UnitRC;**

**interface**

**uses**

**Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,**

**Dialogs, ComCtrls, StdCtrls, ExtCtrls, jpeg;**

**type**

**TForm1 = class(TForm)**

**RadioGroup1: TRadioGroup;**

**GroupBoxRomb: TGroupBox;**

**Label1: TLabel;**

**RombLine: TEdit;**

**Label2: TLabel;**

**RombAngleR: TEdit;**

**GroupBoxCirc: TGroupBox;**

**Label3: TLabel;**

**OkrRad: TEdit;**

**Label4: TLabel;**

**RombAngleC: TEdit;**

**UpDownRombL: TUpDown;**

**UpDownRombAnR: TUpDown;**

**UpDownRadOkr: TUpDown;**

**UpDownRombAnC: TUpDown;**

**DrawImage: TImage;**

**Image1: TImage;**

**Button1: TButton;**

**Image2: TImage;**

**GroupBoxCalc: TGroupBox;**

**Label5: TLabel;**

**Label6: TLabel;**

**Label7: TLabel;**

**Label8: TLabel;**

**Label9: TLabel;**

**Label10: TLabel;**

**Label11: TLabel;**

**Label12: TLabel;**

**Label13: TLabel;**

**Label14: TLabel;**

**procedure RadioGroup1Click(Sender: TObject);**

**procedure Image1Click(Sender: TObject);**

**procedure DrawImageClick(Sender: TObject);**

**procedure Button1Click(Sender: TObject);**

**procedure UpDownRadOkrChanging(Sender: TObject;**

**var AllowChange: Boolean);**

**procedure RombLineChange(Sender: TObject);**

**procedure RombAngleRKeyPress(Sender: TObject; var Key: Char);**

**procedure RombLineKeyPress(Sender: TObject; var Key: Char);**

**procedure RombAngleCKeyPress(Sender: TObject; var Key: Char);**

**procedure OkrRadKeyPress(Sender: TObject; var Key: Char);**

**private**

**{ Private declarations }**

**public**

**{ Public declarations }**

**end;**

**var**

**Form1: TForm1;**

**procedure RedrawAll();**

**implementation**

**{$R \*.dfm}**

**procedure RedrawAll();**

**var ang:double;**

**k: double; // Pi / 180;**

**gx,gy:integer;**

**xc,yc:double;**

**ix,iy:integer;**

**xci,yci:integer;**

**RombLen:double;**

**Cr:double;**

**Scr:double;**

**Sromb:double;**

**Sraz:double;**

**begin**

**Form1.DrawImage.Canvas.FillRect(Form1.DrawImage.Canvas.ClipRect);**

**k:= Pi / 180;**

**Form1.DrawImage.Canvas.Pen.Color:=clRed;**

**gx:=Form1.DrawImage.Width;**

**gy:=Form1.DrawImage.Height;**

**if Form1.RadioGroup1.ItemIndex=0 then**

**begin**

**RombLen:= strtofloat(Form1.RombLine.Text);**

**ang:=k\*strtofloat(Form1.RombAngleR.Text);**

**end**

**else**

**begin**

**ang:=k\*strtofloat(Form1.RombAngleC.Text);**

**RombLen:= (strtofloat(Form1.OkrRad.Text)/(sin(ang/2)))/(cos(ang/2));**

**end;**

**xc:=RombLen\*sin(ang/2);**

**yc:=RombLen\*cos(ang/2);**

**xci:=trunc(xc);**

**yci:=trunc(yc);**

**ix:= trunc(gx/2);**

**iy:= trunc(gy-(gy-(yc+yc))/2);**

**Form1.DrawImage.Canvas.MoveTo(ix,iy);**

**Form1.DrawImage.Canvas.LineTo(ix-xci,iy-yci);**

**Form1.DrawImage.Canvas.LineTo(ix,iy-(yci+yci));**

**Form1.DrawImage.Canvas.LineTo(ix+xci,iy-yci);**

**Form1.DrawImage.Canvas.LineTo(ix,iy);**

**Cr:=((xc+xc)\*(yc+yc))/(4\*RombLen);**

**Form1.DrawImage.Canvas.Pen.Color:=clBlue;**

**Form1.DrawImage.Canvas.Ellipse(ix-trunc(Cr),iy-yci-trunc(Cr),ix+trunc(Cr),iy-yci+trunc(Cr));**

**Scr:=Cr\*Cr\*Pi;**

**Sromb:=((xc+xc)\*(yc+yc))/2;**

**Sraz:=Sromb-Scr;**

**Form1.Label8.Caption:=floattostrf(Sromb,ffFixed,5,2);**

**Form1.Label9.Caption:=floattostrf(Scr,ffFixed,5,2);**

**Form1.Label10.Caption:=floattostrf(Sraz,ffFixed,5,2);**

**Form1.Label12.Caption:=floattostrf(Cr,ffFixed,5,2);**

**Form1.Label13.Caption:=floattostrf(RombLen,ffFixed,5,2);**

**end;**

**procedure TForm1.RadioGroup1Click(Sender: TObject);**

**begin**

**If RadioGroup1.ItemIndex=0 then**

**begin**

**GroupBoxRomb.Visible:=true;**

**GroupBoxCirc.Visible:=false;**

**end**

**else**

**begin**

**GroupBoxRomb.Visible:=false;**

**GroupBoxCirc.Visible:=true;**

**end;**

**RedrawAll();**

**end;**

**procedure TForm1.Image1Click(Sender: TObject);**

**begin**

**Image1.Visible:=false;**

**DrawImage.Visible:=true;**

**end;**

**procedure TForm1.DrawImageClick(Sender: TObject);**

**begin**

**DrawImage.Visible:=false;**

**Image1.Visible:=true;**

**end;**

**procedure TForm1.Button1Click(Sender: TObject);**

**begin**

**if Button1.Tag=0 then**

**begin**

**DrawImage.Visible:=false;**

**Image1.Visible:=false;**

**Image2.Visible:=true;**

**Button1.Tag:=255;**

**end**

**else**

**begin**

**Button1.Tag:=0;**

**DrawImage.Visible:=true;**

**Image1.Visible:=false;**

**Image2.Visible:=false;**

**end**

**end;**

**procedure TForm1.UpDownRadOkrChanging(Sender: TObject;**

**var AllowChange: Boolean);**

**begin**

**OkrRad.Text:=OkrRad.Text**

**end;**

**procedure TForm1.RombLineChange(Sender: TObject);**

**begin**

**if Length((Sender as Tedit).Text)=0 then exit;**

**if strtoint((Sender as Tedit).Text)=0 then exit;**

**RedrawAll();**

**end;**

**procedure TForm1.RombAngleRKeyPress(Sender: TObject; var Key: Char);**

**begin**

**if not (Key in ['0'..'9', #8])then Key:=#0;**

**end;**

**procedure TForm1.RombLineKeyPress(Sender: TObject; var Key: Char);**

**begin**

**if not (Key in ['0'..'9', #8])then Key:=#0;**

**end;**

**procedure TForm1.RombAngleCKeyPress(Sender: TObject; var Key: Char);**

**begin**

**if not (Key in ['0'..'9', #8])then Key:=#0;**

**end;**

**procedure TForm1.OkrRadKeyPress(Sender: TObject; var Key: Char);**

**begin**

**if not (Key in ['0'..'9', #8])then Key:=#0;**

**end;**

**end.**