Technical Guide for the Area calculator version 1.0

Made by Savelij Borovskij

# Content

Content

[Content 1](#_Toc71535379)

[Development Environment 2](#_Toc71535380)

[Target Platform 2](#_Toc71535381)

[User Requirements 2](#_Toc71535382)

[Design of the programming solution (screen layout) 2](#_Toc71535383)

[Working program 4](#_Toc71535384)

[Code 4](#_Toc71535385)

[Test plan 9](#_Toc71535386)

[Test log 11](#_Toc71535387)

[Maintenance 12](#_Toc71535388)

## Development Environment

For the development environment I chose Visual Basic .NET programming language. As an environment I used window form programming, which is both object-oriented and procedural programming.

## Target Platform

The target platform for the Area Calculator program is Windows operating system.

## User Requirements

I am asked to develop a small object-oriented program that shows the key features of OOP programming. It is called Area Calculator. The program allows user to calculate areas of different types of shapes, using base, length, width, height, etc.; Once the required inputs are provided the program will calculate an area of the shape and present the result.

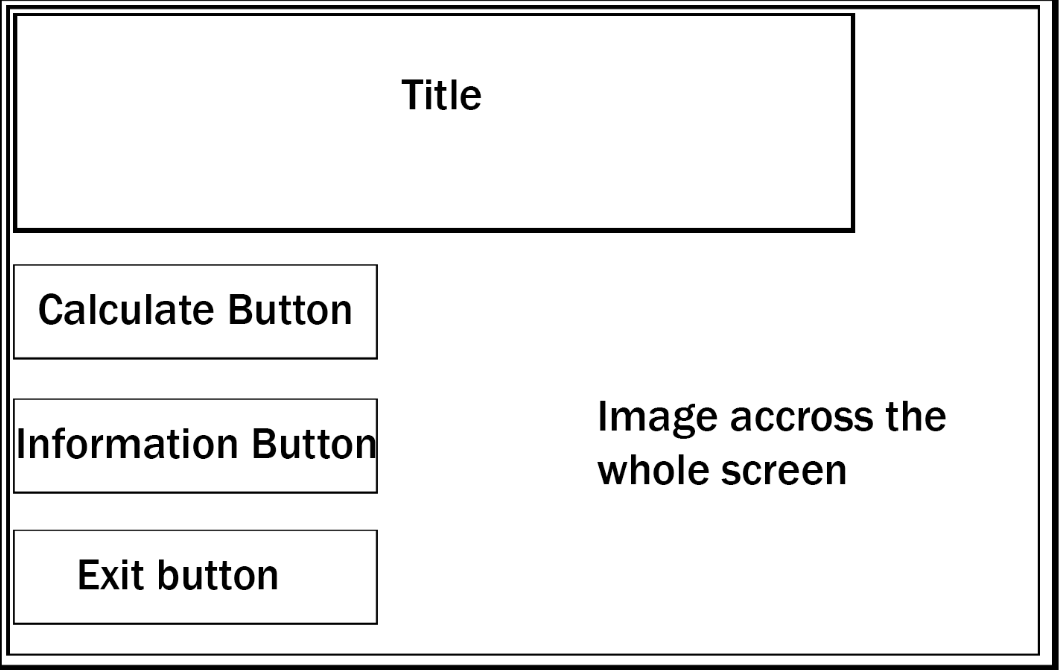
Additional features:

1) On-screen help

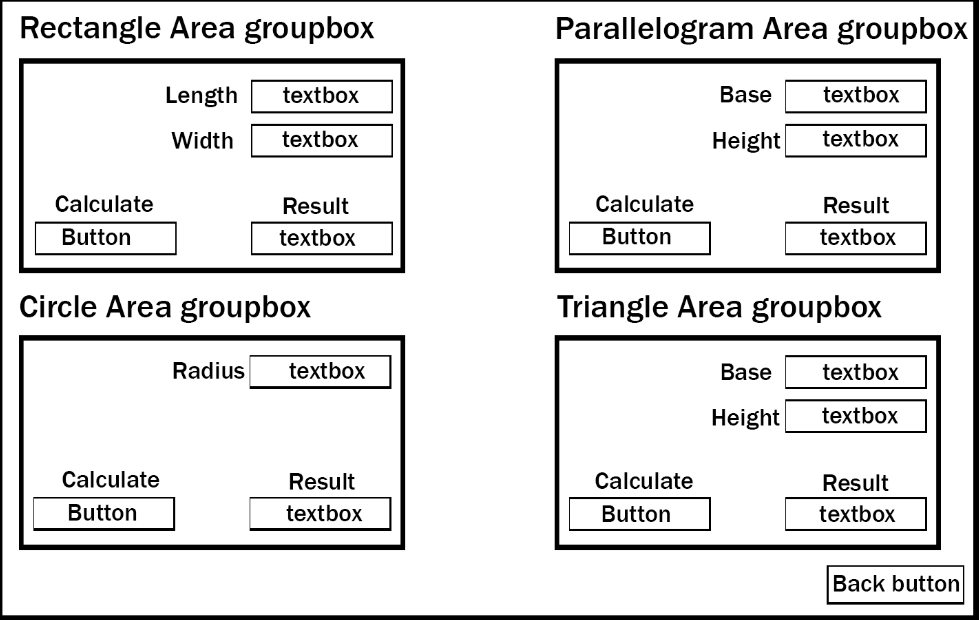
2) Information screen

## Design of the programming solution (screen layout)

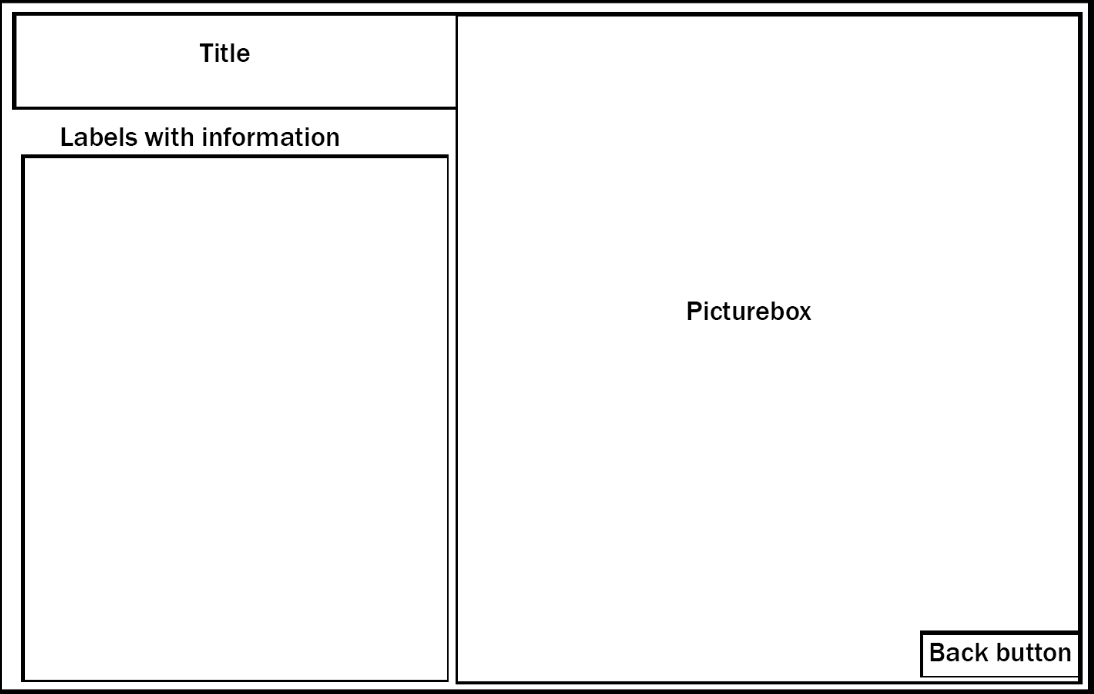
1) Main menu screen



2) Calculate screen

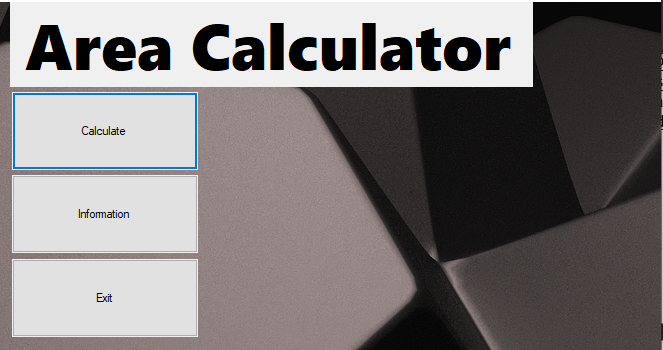


3) Information screen

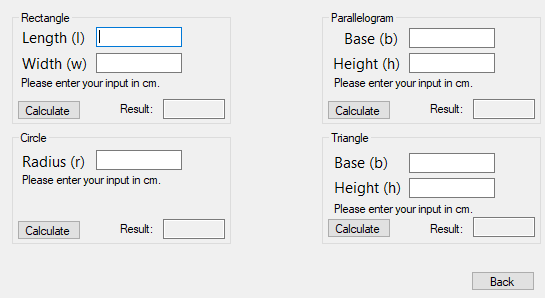


## Working program

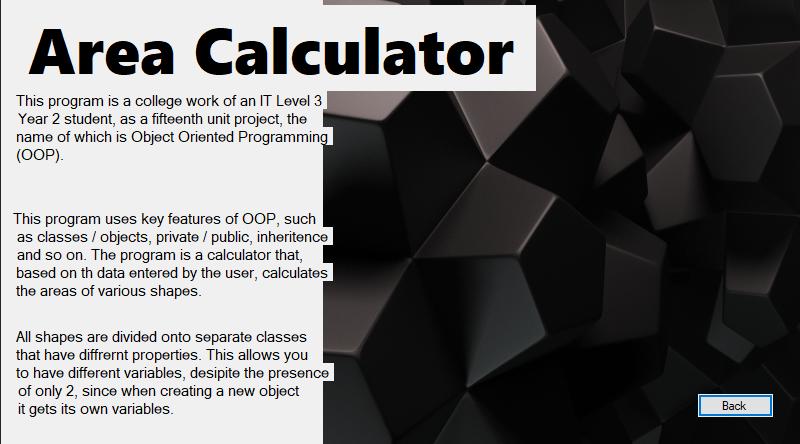
Main screen



Calculation screen



Information screen



## Code

frmMain

Public Class frmMain

Private Sub btnExit\_Click(sender As Object, e As EventArgs) Handles btnExit.Click

End 'closes program

End Sub

Private Sub btnCalculate\_Click(sender As Object, e As EventArgs) Handles btnCalculate.Click

frmCal.Show() 'shows another window

Me.Hide() 'hides current window

End Sub

Private Sub btnInfo\_Click(sender As Object, e As EventArgs) Handles btnInfo.Click

frmInfo.Show() 'shows another window

Me.Hide() 'hides current window

End Sub

End Class

frmCal

Public Class frmCal

'declaring new OBJECTS from their CLASSES

Dim Rec1 As clsRectangle = New clsRectangle

Dim Par1 As clsParallelogram = New clsParallelogram

Dim Tri1 As clsTriangle = New clsTriangle

Dim Cir1 As clsCircle = New clsCircle

Private Sub Button1\_Click(sender As Object, e As EventArgs) Handles btnBack.Click

frmMain.Show() 'shows another window

Me.Hide() 'hides current window

End Sub

'RECTANGLE

Private Sub txtRecL\_TextChanged(sender As Object, e As EventArgs) Handles txtRecL.TextChanged

Rec1.Var1 = Val(txtRecL.Text) 'first variable of this object will be equal to that textbox input

End Sub

Private Sub txtRecW\_TextChanged(sender As Object, e As EventArgs) Handles txtRecW.TextChanged

Rec1.Var2 = Val(txtRecW.Text) 'second variable of this object will be equal to that textbox input

End Sub

Private Sub btnRec\_Click(sender As Object, e As EventArgs) Handles btnRec.Click

Rec1.Area() 'calling Area sub from SHAPE class to calculate area

txtRecResult.Text = Rec1.Result 'displaying the result in the specific textbox

End Sub

'PARALLELOGRAM

Private Sub txtParB\_TextChanged(sender As Object, e As EventArgs) Handles txtParB.TextChanged

Par1.Var1 = Val(txtParB.Text) 'first variable of this object will be equal to that textbox input

End Sub

Private Sub txtParH\_TextChanged(sender As Object, e As EventArgs) Handles txtParH.TextChanged

Par1.Var2 = Val(txtParH.Text) 'second variable of this object will be equal to that textbox input

End Sub

Private Sub btnPar\_Click(sender As Object, e As EventArgs) Handles btnPar.Click

Par1.Area() 'calling Area sub from SHAPE class to calculate area

txtParResult.Text = Par1.Result 'displaying the result in the specific textbox

End Sub

'CIRCLE

Private Sub txtCirR\_TextChanged(sender As Object, e As EventArgs) Handles txtCirR.TextChanged

Cir1.Var1 = Val(txtCirR.Text) 'first variable of this object will be equal to that textbox input

End Sub

Private Sub btnCir\_Click(sender As Object, e As EventArgs) Handles btnCir.Click

Cir1.Area() 'calling another sub from Circle class to calculate area

txtCirResult.Text = Cir1.Result

End Sub

'TRIANGLE

Private Sub txtTrianA\_TextChanged(sender As Object, e As EventArgs) Handles txtTrianBase.TextChanged

Tri1.Var1 = Val(txtTrianBase.Text) 'first variable of this object will be equal to that textbox input

End Sub

Private Sub txtTrianB\_TextChanged(sender As Object, e As EventArgs) Handles txtTrianHeight.TextChanged

Tri1.Var2 = Val(txtTrianHeight.Text) 'second variable of this object will be equal to that textbox input

End Sub

Private Sub btnTrian\_Click(sender As Object, e As EventArgs) Handles btnTrian.Click

Tri1.Area() 'calling Area sub from SHAPE class to calculate area

txtTrianResult.Text = Tri1.Result 'displaying the result in the specific textbox

End Sub

End Class

frmInfo

Public Class frmInfo

Private Sub Button1\_Click(sender As Object, e As EventArgs) Handles btnBack.Click

Me.Hide()

frmMain.Show()

End Sub

End Class

clsShape

Public MustInherit Class ClsShape 'Keyword MustInherit indicates that this is an Abstract Class

'Setting values as private

Private \_Var1 As Integer = 0

Private \_Var2 As Integer = 0

Private \_Result As Integer = 0

'creating sub to calculate area

Public Overridable Sub Area() 'Making that Sub overridable so that whenever I need to override the formula inside I would be able to

Result = Var1 \* Var2

End Sub

Public Property Var1 As Integer 'creating a new function to interact with private values

Get 'read

Return \_Var1

End Get

Set(value As Integer) 'write

\_Var1 = value

End Set

End Property

Public Property Var2 As Integer 'creating a new function to interact with private values

Get 'read

Return \_Var2

End Get

Set(value As Integer)

\_Var2 = value 'write

End Set

End Property

Public Property Result 'creating a new function to interact with private values

Get 'read

Return \_Result

End Get

Set(value)

\_Result = value 'write

End Set

End Property

End Class

clsRectangle

Public Class clsRectangle 'creating class for rectangle shape

Inherits ClsShape 'Inherits abstract class clsShape

End Class

clsParallelogram

Public Class clsParallelogram 'creating class for parallelogram shape

Inherits ClsShape 'Inherits abstract class clsShape

End Class

clsTriangle

Public Class clsTriangle 'creating class for triangle shape

Inherits ClsShape 'Inherits abstract class clsShape

'Making new sub that will override Area sub from clsShape to add a new formula

Public Overrides Sub Area() 'adding new formula, because to calculate an area for the triangle we have to use a different one

Result = Var1 \* Var2 / 2

End Sub

End Class

clsCircle

Public Class clsCircle 'creating class for circle shape

Inherits ClsShape 'Inherits abstract class clsShape

'Making new sub that will override Area sub from clsShape to add a new formula

Public Overrides Sub Area() 'adding new formula, because to calculate an area for the circle we have to use a different one

Result = Var1 \* Math.PI 'formula Math.PI that stands for PI number value

End Sub

End Class

## Test plan

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. | What tested (Test Item) | Test data | Expected outcome | Test description |
| 1 | Length | 1 | Accept | Test if the program accepts the correct input |
| 2 | Length | -1 | Accept | Test if the program accepts the correct input |
| 3 | Length | A | Not accept | Test if the program does not accept incorrect input |
| 4 | Width | 2 | Accept | Test if the program accepts the correct input |
| 5 | Width | -2 | Accept | Test if the program accepts the correct input |
| 6 | Width | B | Not accept | Test if the program does not accept incorrect input |
| 7 | Radius | 3 | Accept | Test if the program accepts the correct input |
| 8 | Radius | -3 | Accept | Test if the program accepts the correct input |
| 9 | Radius | C | Accept | Test if the program does not accept incorrect input |
| 10 | Base | 4 | Accept | Test if the program accepts the correct input |
| 11 | Base | -4 | Accept | Test if the program accepts the correct input |
| 12 | Base | D | Not accept | Test if the program does not accept incorrect input |
| 13 | Height | 5 | Accept | Test if the program accepts the correct input |
| 14 | Height | -5 | Accept | Test if the program accepts the correct input |
| 15 | Height | E | Not accept | Test if the program does not accept incorrect input |
| 16 | Length | ‘’ | Not accept | Test if the program does not accept incorrect input |
| 17 | Width | %»;!»№ | Not accept | Test if the program does not accept incorrect input |
| 18 | Length; Width | -1, 25 | Accept | Test if the program accepts the correct input |
| 19 | Length, Width | Ab, 11 | Not accept | Test if the program does not accept incorrect input |
| 20 | Length, Width | sdo, qwe | Not accept | Test if the program does not accept incorrect input |

## Test log

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No. | What tested (Test Item) | Test data | Expected outcome | Actual outcome | Note |
| 1 | Length | 1 | Accept | Accept | Passed |
| 2 | Length | -1 | Accept | Accept | Passed |
| 3 | Length | A | Not accept | Accept | Not passed |
| 4 | Width | 2 | Accept | Accept | Passed |
| 5 | Width | -2 | Accept | Accept | Passed |
| 6 | Width | B | Not accept | Accept | Passed |
| 7 | Radius | 3 | Accept | Accept | Accept |
| 8 | Radius | -3 | Accept | Accept | Accept |
| 9 | Radius | C | Accept | Accept | Accept |
| 10 | Base | 4 | Accept | Accept | Accept |
| 11 | Base | -4 | Accept | Accept | Accept |
| 12 | Base | D | Not accept | Accept | Not passed |
| 13 | Height | 5 | Accept | Accept | Accept |
| 14 | Height | -5 | Accept | Accept | Accept |
| 15 | Height | E | Not accept | Accept | Not passed |
| 16 | Length | ‘’ | Not accept | Not accept | Passed |
| 17 | Width | %»;!»№ | Not accept | Accept | Not passed |
| 18 | Length; Width | -1, 25 | Accept | Accept | Passed |
| 19 | Length, Width | Ab, 11 | Not accept | Accept | Not passed |
| 20 | Base, Height | sdo, qwe | Not accept | Accept | Not passed |

## Maintenance

To service my program, I will need to conduct tests and debugs every month. To do this, feedback from users will be collected and errors will be checked and resolved. Also in the future, it is planned to improve the graphic design by supplementing the interface and adding various frames, lines, etc.