、《面向对象程序设计语言C++》

**课程项目设计总结报告**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 项目名称 | 图形面积周长计算器 | | | | |
| 学号 |  |  |  |  |  |
| 邮件 |  | |  |  | |
| 完成时间 | 2017-1-5 | | | | |

**摘要：**

本项目是用c++语言编写，实现了基本图形周长面积的计算，和绘制图形面积的计算。采用各个图形面积周长计算公式计算（如扇形，任意直线图形面积计算），图形绘制后采用坐标系比例与图形所包含的像素点的个数计算面积。与同类软件相比而言计算包含的图形较少，绘制图形面积计算的绘制种类较少并且只是2D绘制，但是本软件图形界面相对其他软件相对优美，且界面简洁易懂，面积周长计算上基本能满足需求。

**一、项目概述**

**1、项目简介**

图形周长面积计算软件，给用户提供了方便简洁的计算界面，可以为用户提供图形计算的选着，计算相对的图形计算，同时还可以选着绘制图形，计算用户绘制的图形面积；软件拥有良好的界面交互，为用户的使用带来便捷（如概念模型的绘制，当前选择的提示等）。

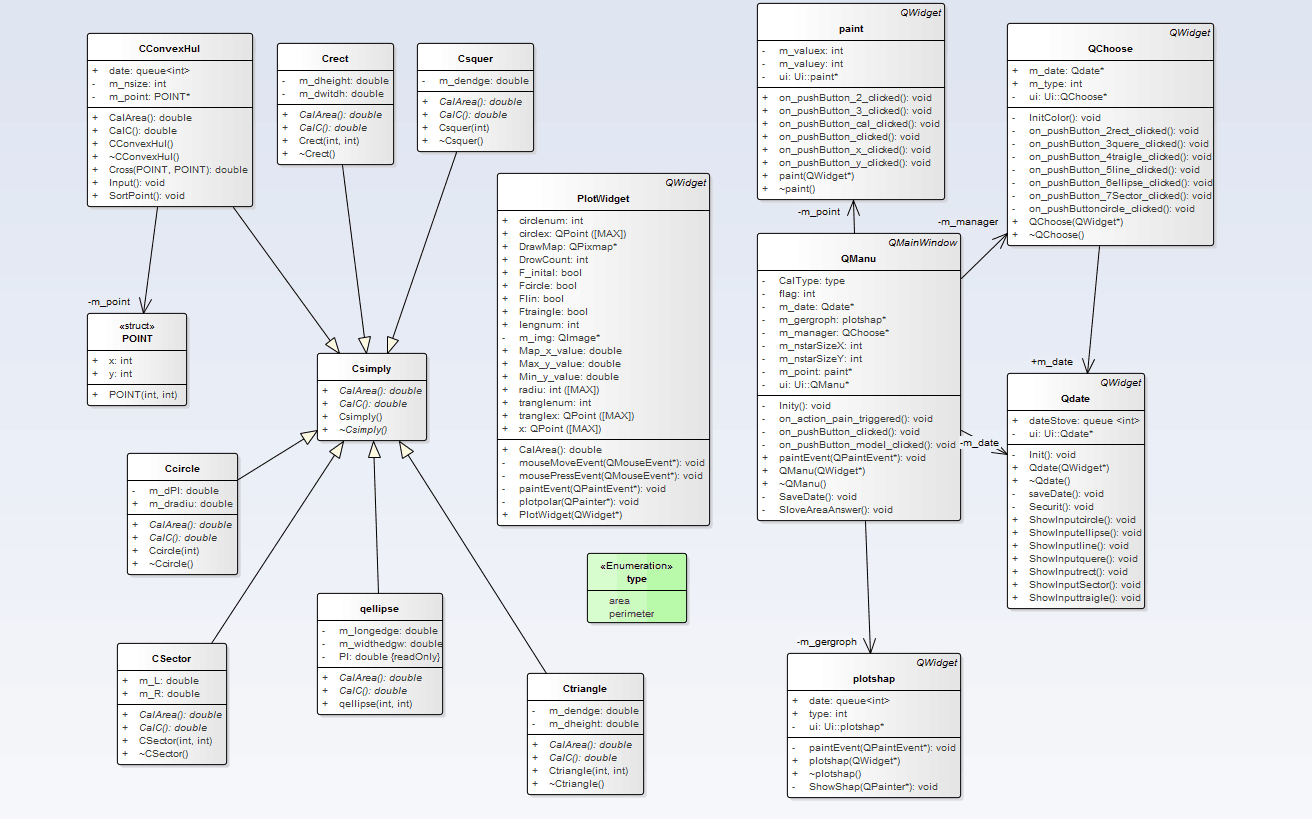
1. 功能要求

|  |  |  |
| --- | --- | --- |
| **编号** | **功能模块** | **功能描述（功能描述中要有数据，比如多少种飞机、动物；存储到什么文件。）** |
| FUN\_01 | 计算图形选择模块 | 根据用户图形选择对应计算图形，并在主界面上转换输入界面 |
| FUN\_02 | 计算模式转换 | 点击转换模式（面积计算模式—周长计算模式）转换，计算出不同的模式值。 |
| Fun\_03 | 概念模型展示 | 输入需要的数据点击确定后，会根据数据展示出对应的模型（及变化的模型） |
| Fun\_04 | 数据输入 | 对应不同图形有不同的数据输入界面 |
| Fun\_05 | 坐标轴转换 | 根据设置转变坐标轴的值，以改变每个像素点代表的大小 |
| Fun\_06 | 图形绘制选择 | 可以选择需要绘制图像类型 |
| Fun\_07 | 图形绘制 | 在界面上绘制图形 |
| Fun\_08 | 绘制图形面积计算 | 根据坐标轴值和用户绘制的图形点击确定后计算图形面积 |

**二、总体设计**

1、类关系图,可由VS直接导出总体的类关系图；

给出结构关系图，并简要说明，结构的优缺点，**从面向对象封装、继承及多态应用角度讨论。**

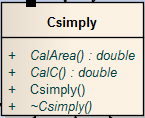


Csimply作为基类派生出椭圆，圆……等，QManu类及菜单由Qdate(输入数据保存模块)，plotshap（概念模型模块），Qchoose(图形选择模块)，paint(绘制图形模块)组和而成。PltWidget是自己写的一个插件（如文本框可以插入页面）是一个坐标系可以在上面绘制图形，被插入到了paint中。图形有一个基类便于调用保存等，页面采用组和方式比较直观，但manu类过于庞大功能权限过大。

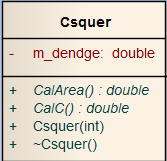
2、设计类图

每个类的类图应完整包含对应的类名、属性名称与数据类型、函数名及其对应返回值、参数。请使用ArgoUML、StarUML或自己百度找UML工具完成。比如下面这个类图：

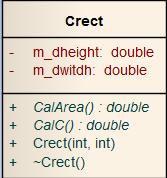
1. 普通图形的基类，派生出其他图形类（包含面积周长计算函数）。



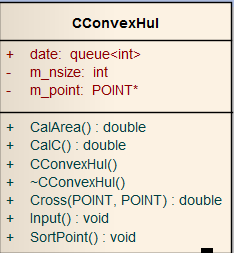
2. 由Csimply派生出的正方形类。



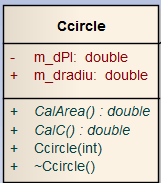
3. 由Csimply派生出的长方形类。



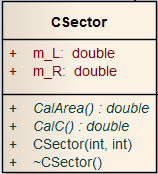
4.由Csimply派生出的任意直线构成的图形类。



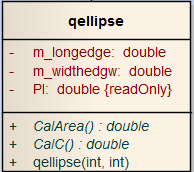
5.由Csimply派生出的圆形类。



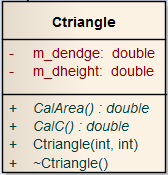
6.由Csimply派生出的类。



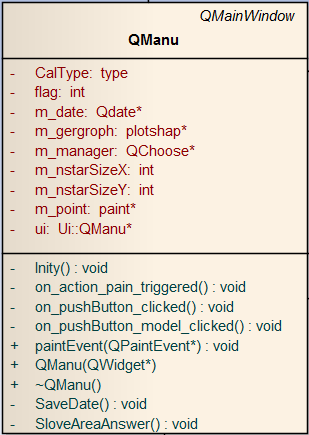
7.由Csimply派生出的椭圆类。



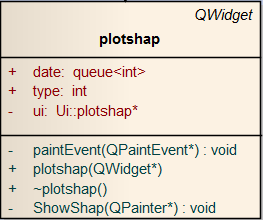
8. 由Csimply派生出的三角形类；



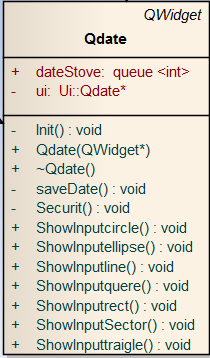
9.界面QManu类,既生成程序的界面。



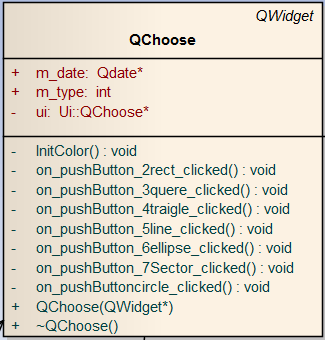
10.程序界面中的概念模型绘制类；



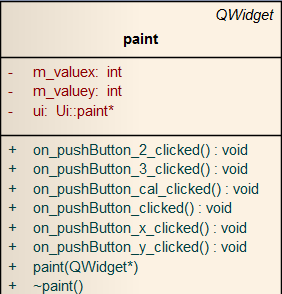
11.输入数据管理类，既保存界面的输入数据。数据类。同时控制输入界面的变换。



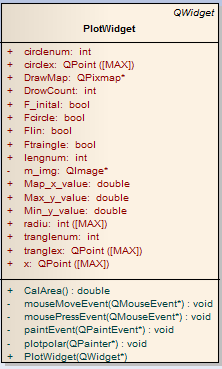
12.界面中用户选择计算图形类，既当前用户选择了那个类。



13.画图类，界面点击画图后生成的界面，进行图形的绘制。



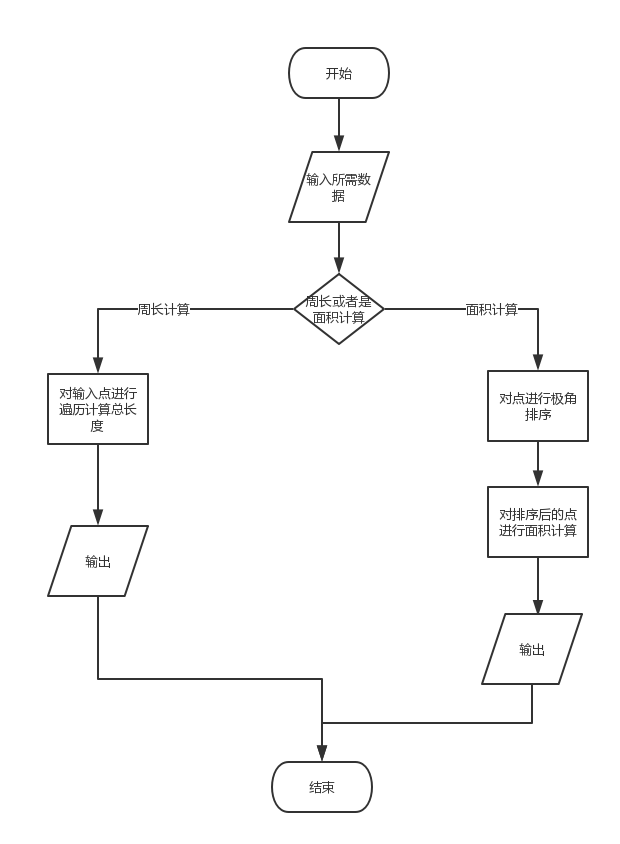
14.自己做的一个插件（坐标系） 插入paint类；在paint 的UI中添加成基本控件。



**三、核心算法**

**1. 不规则图形之任意直线图形周长面积计算。**

1. 选取p0作为y坐标最小的点，如果y坐标相等，选取x坐标最小的点
2. 对剩余的点相对与p0点的极角进行排序。
3. 排序后分为三角形对各个三角形去求面积。



CConvexHul类下的三个函数。

void CConvexHul::SortPoint()

{

int i,j;

POINT temp(m\_point[0].x,m\_point[0].y); //选取作为基点

int index = 0;

for (i = 0; i < m\_nsize; i++)

{

if (m\_point[i].y < temp.y || (m\_point[i].y == temp.y&&m\_point[i].x < temp.x))

{

index = i;

temp.x = m\_point[i].x;

temp.y = m\_point[i].y;

}

}

m\_point[index] = m\_point[0];

m\_point[0] = temp;

for (i = 1; i < m\_nsize; i++)

{

for (j = 1; j < m\_nsize - 1 - i; j++)

{

if ((m\_point[j].y - temp.y) && (m\_point[j + 1].y - temp.y))

{

if ((m\_point[j].x - temp.x) / (m\_point[j].y - temp.y) < (m\_point[j + 1].x - temp.x) / (m\_point[j + 1].y - temp.y))

{

POINT replace;

replace = m\_point[j];

m\_point[j] = m\_point[j + 1];

m\_point[j + 1] = temp;

}

}

}

}

}

/\*

\*计算图形面积

\*/

double CConvexHul::*CalArea*()

{

SortPoint();

int i;

double area = 0;

for (i = 1; i < m\_nsize-1; i++)

{

POINT temp1(m\_point[i].x - m\_point[0].x, m\_point[i].y - m\_point[0].y), temp2(m\_point[i + 1].x - m\_point[0].x, m\_point[i + 1].y - m\_point[0].y);

area += Cross(temp1, temp2);

}

return area / 2;

}

/\*

\* 计算图形周长

\*/

double CConvexHul::*CalC*()

{

double sum=0;

for(int i=0;i<m\_nsize;i++)

{

sum+=sqrt((m\_point[i].x-m\_point[i].x)\*(m\_point[i].x-m\_point[i].x)+

(m\_point[i].y-m\_point[i].y)\*(m\_point[i].y-m\_point[i].y));

}

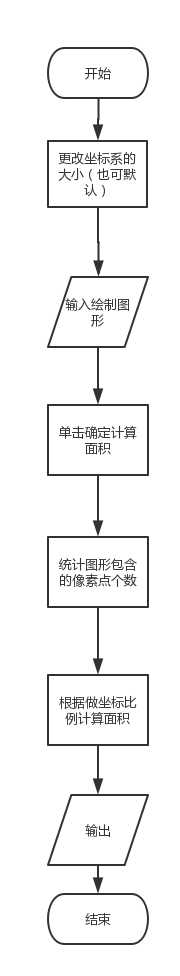
return sum;

}

1. **对绘制图形的面积计算。**

在图形在坐标系下绘制完成后，点击确认按钮就开始本函数。

实现是通过统计闭合图形内的像素点个数，根据坐标系得到每个像素点的值最后得到面积。

 /\*

\* 计算绘制图形包含多少像素点；

\*/

double PlotWidget::CalArea()

{

QRgb value=qRgb(255,255,255);

int i,j;

double sum=0;

//统计像素点。

for(i=0;i<width();i++)

{

int maxup=0;

int maxdown=0;

for(j=0;j<height();j++)

{

if( m\_img->pixel(i,j)!=value)

{

maxup=j;

break;

}

}

for(j=height()-10;j>=0;j--)

{

if( m\_img->pixel(i,j)!=value)

{

maxdown=j;

break;

}

}

sum+=maxdown-maxup;

}

return sum;

}

/\*根据坐标系值大小计算面积\*/

void paint::on\_pushButton\_cal\_clicked()

{

double right\_pos = width() - right\_margine;

double bottom\_pos = height()- bottom\_margine;

double area=ui->plotview->CalArea();

area=(double)((m\_valuex\*m\_valuey)/((right\_pos-left\_margine)\*(bottom\_pos-top\_margine)))\*area; //计算公式，得到单个像素点表示的值乘以所包含的像素点个数。

QString q=QString("%1").arg(area,0,'g',9);

ui->label\_Area->setText(q);

}

**五、程序实现**

1、开发平台与编译运行要求

|  |  |  |  |
| --- | --- | --- | --- |
| 硬件配置 | 操作系统及版本 | 开发平台及版本 | 第三方支持库及版本 |
|  | Windows10系统 | Qt |  |

如果有特殊编译运行要求，因给出具体说明。

1. 类似上面核心算法讨论一样，分节对核心功能（1）、（2）、（3）.......的实现截图与简要功能分析。

**1. 不规则图形之任意直线图形周长面积计算。**

功能分析：实现不规则图形面积的计算。

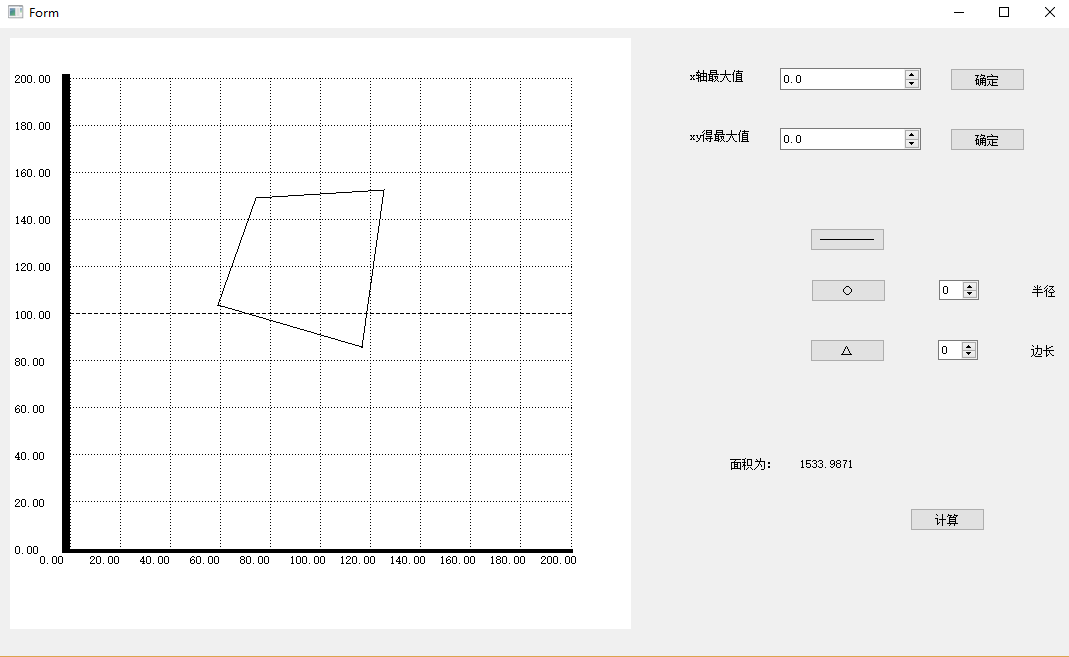
实现截图：



1. **对绘制图形的面积计算。**

功能分析：对绘制图形面积的计算。

实现截图：



3、功能完成度

对照2.2节 功能要求表，完成下表。有新增功能，在完成度上写新增。

|  |  |  |  |
| --- | --- | --- | --- |
| **编号** | **功能模块** | **完成情况描述** | **完成度** |
| FUN\_01 | 计算图形选择模块 | 根据用户图形选择对应计算图形，并在主界面上转换输入界面 | 100% |
| FUN\_02 | 计算模式转换 | 点击转换模式（面积计算模式—周长计算模式）转换，计算出不同的模式值。 | 100% |
| Fun\_03 | 概念模型展示 | 输入需要的数据点击确定后，会根据数据展示出对应的模型（及变化的模型 | 90%(由直线构成的图形只是简单一条直线) |
| Fun\_04 | 数据输入 | 对应不同图形有不同的数据输入界面 | 100% |
| Fun\_05 | 坐标轴转换 | 根据设置转变坐标轴的值， | 100% |
| Fun\_06 | 图形绘制选择 | 可以选择需要绘制图像类型 | 100% |
| Fun\_07 | 图形绘制 | 能在界面上绘制图形 | 100% |
| Fun\_08 | 绘制图形面积计算 | 能根据坐标轴值和用户绘制的图形点击确定后计算图形面积 | 100% |

**六、总结**

自己程序的优缺点，与现有同类软件的比较。

本软件界面优美，操作简介，有图形绘制计算也有输入值计算，包含面广。相对缺点是输入值的判定是否为合法数据（如绘制的图形是闭合图形）没有做判定。

**参考资料**

1. **CAD中的不规则图形面积计算**
2. **参考软件。**

**《计算不规则多边形面积》参考设置及其模式。**