1. **Qt进行绘图时,要重写绘图函数**

void *paintEvent*(QPaintEvent \*);

**2.定义绘图对象,在paintEvent函数中**

QPainter painter(this);

**3.将坐标系的该点变成原点坐标**

painter.translate(this->width()-ui->textEditRead->width()+0.5\*ui->SliderValue->width(),this->height()-1.8\*ui->SliderValue->height()-5);

**4.设置画笔颜色**

QPen pen(Qt::*black*,3,Qt::*SolidLine*,Qt::*RoundCap*,Qt::*RoundJoin*);

painter.setPen(pen);

painter.setBrush(QColor(200,200,200,125));//设置画刷,最后一位为透明度

**5. 画下面直线,画从(0,0)点到(60,0)的**

painter.drawLine(0,0,60,0);

**6.填充颜色，填充(0,5)到(30,20)到(60,5)这个三角形之间的颜色，并设置画刷为青色**

QPolygon polyGon;

polyGon.clear();

polyGon<<QPoint(0,5);

polyGon<<QPoint(30,20);

polyGon<<QPoint(60,5);

painter.setBrush(Qt::*cyan*);

painter.drawConvexPolygon(polyGon);

**7.在刻度线上写字**

painter.drawText(pointLeft,QString::number(num));

**8.设置渐变，在y坐标为sliderHigh\*30/valueSlider上设置渐变，比例为0,0.3,0.7,1.0，用矩形区域,0,0,60, sliderHigh来表达渐变**

//线性渐变

QlinearGradient linerGradient=QLinearGradient(0,0,0,1000);

linerGradient.setColorAt(0.0,Qt::*cyan*);

linerGradient.setColorAt(0.3,Qt::*green*);

linerGradient.setColorAt(0.7,Qt::*yellow*);

linerGradient.setColorAt(1.0,Qt::*red*);

QBrush qsh(linerGradient);

painter.setBrush(qsh);

QPen pen(Qt::*red*,3,Qt::*SolidLine*,Qt::*RoundCap*,Qt::*RoundJoin*);

painter.drawRect(0, 0, 60, sliderHigh);

//圆形渐变

QRadialGradient radiaGradient(200,200,150,150,100);

radiaGradient.setColorAt(0.0,Qt::*white*);

radiaGradient.setColorAt(0.2,color);

radiaGradient.setColorAt(1.0,Qt::*black*);

paint->setBrush(radiaGradient);

//锥形渐变

QConicalGradient conicalGradient(200,200,30);

conicalGradient.setColorAt(0.0,Qt::*white*);

conicalGradient.setColorAt(0.2,color);

conicalGradient.setColorAt(1.0,Qt::*black*);

paint->setBrush(conicalGradient);

**9.在函数的最后设置更新**

update(); //刷新

**10.加载图片，并且放大缩小图片**

QImage img; //图像

QMatrix matrix;

matrix.scale(0.5,0.5);

img.load(":/new/prefix1/RES/car.png");

img=img.transformed(matrix);

painter.drawImage(-60,-120,img);

**11.绘制扇形**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*设置参数\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

int valueX=-150; //矩形起始位置横坐标

int valueY=-150; //矩形起始位置纵坐标

int valueArea=300;//矩形的场合宽

int startAngle=60;//开始角度

int rangeAngle=60;//角度跨度

QPainter painterTail(ui->widgetPaint); painterTail.setPen(QColor(200,200,200,125));

QRect rectTailRight(valueX,valueY,valueArea,valueArea);//画一个矩形域

QPointF centerTailRight(0,0); //圆弧的圆心坐标

QPainterPath myPathTailRight; //设置路径

myPathTailRight.moveTo(centerTailRight);//将点移动到圆心

myPathTailRight.arcTo(rectTailRight,180+startAngle,rangeAngle);//从圆心画圆弧

painterTail.setBrush(QColor(200,200,200,125));//设置画刷为灰色

painterTail.drawPath(myPathTailRight); //画出图形

**12.绘制椭圆**

QPainter painterTotal(ui->widgetPaint); painterTotal.translate(pointX+0.45\*img.width()+10,pointY+0.34\*img.height()-10);//将该点变为直角坐标系的原点

QPainterPath myPathTotal; //设置路径

painterTotal.setPen(QColor(180,0,0,62.5)); //设置画笔 painterTotal.setBrush(QColor(180,0,0,125));//设置画刷

QPointF center(valueX+0.5\*valueArea+45,valueY+0.5\*valueArea+135);

myPathTotal.moveTo(center); //将点移动到圆心

myPathTotal.addEllipse(center,120,200); //从圆心画圆弧

painterTotal.drawPath(myPathTotal); //画出图形

**13.消锯齿**

painter.setRenderHint(QPainter::*Antialiasing*);//消锯齿