

1 qt绘图的机制

2 绘图

3 更新绘图

1 qt绘图的机制

qt中的所有的图都是画出来的 例如我们的按钮 窗口 标签上的图

画图需要两个东西:

画家:

QPainter

画板

QPaintDevice(常见的控件 按钮 label 窗口...)

! PaintEngine 类是抽象类, 提供了 QPainter 如何在指定的平台上给指定的设备绘画的抽象接口, 对开发者而言, 一般不会用到。



2 绘图

```
1 #include "widget.h"
2 #include "ui_widget.h"
3 #include <QDebug>
4 #include <QPainter>
5 #include <QPen>
6 Widget::Widget(QWidget *parent) :
7     QWidget(parent),
8     ui(new Ui::Widget)
9 {
10     ui->setupUi(this);
11 }
12
```

```

13 Widget::~Widget()
14 {
15     delete ui;
16 }
17
18 void Widget::paintEvent(QPaintEvent *event)
19 {
20
21     //绘图一定需要在这个函数中绘图
22     //画家
23     QPainter p(this); //定义一个画家 并且制定了绘图设备
24     p.drawPixmap(0,0,this->width(),this->
height(),QPixmap("../Image/bk.jpg"));
25     QPen pen; // 定义一支笔
26     pen.setWidth(5);
27     pen.setColor(QColor(Qt::red)); //设置笔的颜色
28     //将笔给到画家
29     p.setPen(pen);
30     p.drawLine(50,50,300,300); //画线
31     p.drawEllipse(0,0,this->width(),this->height());
32 }
33

```

```

1 void Widget::paintEvent(QPaintEvent *e)
2 {
3     qDebug() << "in paintEvent" << "recet" << e->rect(); //得到需要重新绘制的区域
4     QPainter paint(this); //定义画家（请了一个画家来画画），画在主窗口上
5     //画家画图片，作为背景图片
6     paint.drawPixmap(0,0,this->width(),this->height(),
7     QPixmap("../Image/bk.jpg"));
8     QPen pen; //定义一只画笔
9     pen.setColor(QColor(255,0,255)); //设置画笔的颜色
10    pen.setWidth(5); //设置画笔的宽度
11    pen.setStyle(Qt::DashDotLine); //设置画笔线条的风格
12    paint.setPen(pen); //画家使用这只画笔
13    paint.drawLine(30,30,500,30); //画家画水平的线条
14    paint.drawLine(30,30,30,500); //画家画垂直的线条
15    pen.setColor(Qt::yellow); //设置画笔的颜色
16    pen.setStyle(Qt::SolidLine); //设置画笔线条的风格
17    pen.setWidth(3); //设置线条的宽度

```

```

18 paint.setPen(pen);//画家使用这只画笔
19 QBrush brush(Qt::blue);//定义一把蓝色的画刷
20 brush.setStyle(Qt::DiagCrossPattern);//设置画刷的风格
21 paint.setBrush(brush);//画家使用这把画刷
22 paint.drawRect(50,50,150,200);//画家画矩形
23 paint.setPen(QPen());
24 paint.setBrush(QBrush(QColor(255,187,255),Qt::Dense3Pattern));
25 paint.drawEllipse(280,60,180,250);//画家画椭圆
26 }

```

3 更新绘图

调用 this->update更新绘图

```

1 #include "widget.h"
2 #include "ui_widget.h"
3 #include <QPainter>
4 #include <QPixmap>
5 Widget::Widget(QWidget *parent) :
6     QWidget(parent),
7     ui(new Ui::Widget)
8 {
9     ui->setupUi(this);
10    i=0;
11    j=0;
12 }
13
14 Widget::~~Widget()
15 {
16     delete ui;
17 }
18 void Widget::paintEvent(QPaintEvent *event)
19 {
20
21     QPainter p(this);
22     p.drawPixmap( i,j,50,50,QPixmap("../Image/butterfly.png"));
23
24 }
25
26 void Widget::mousePressEvent(QMouseEvent *event)
27 {

```

```
28
29  i = event->x();
30  j = event->y();
31  //刷新绘图
32  this->update();
33  }
34
```