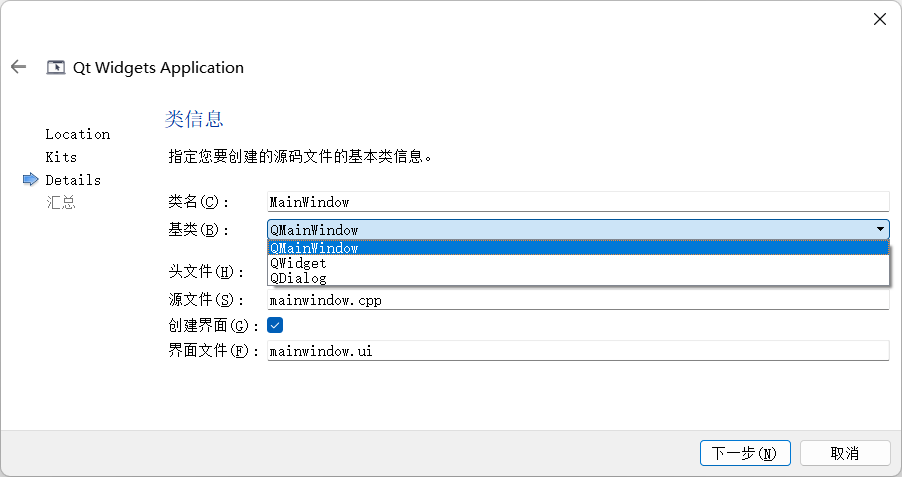
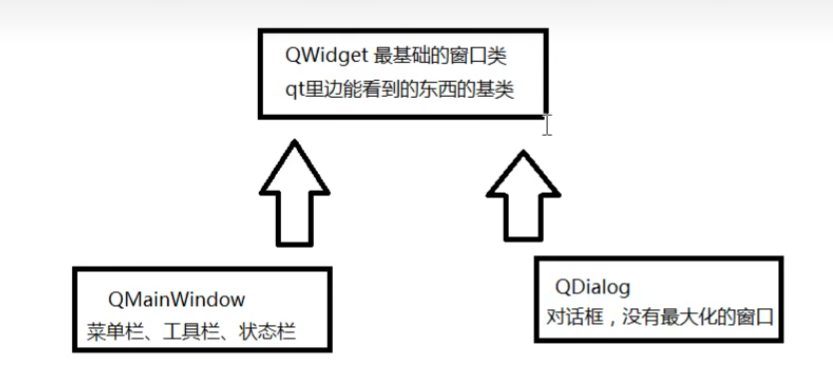
1.Ctrl+W快捷关闭文件

2.创建项目



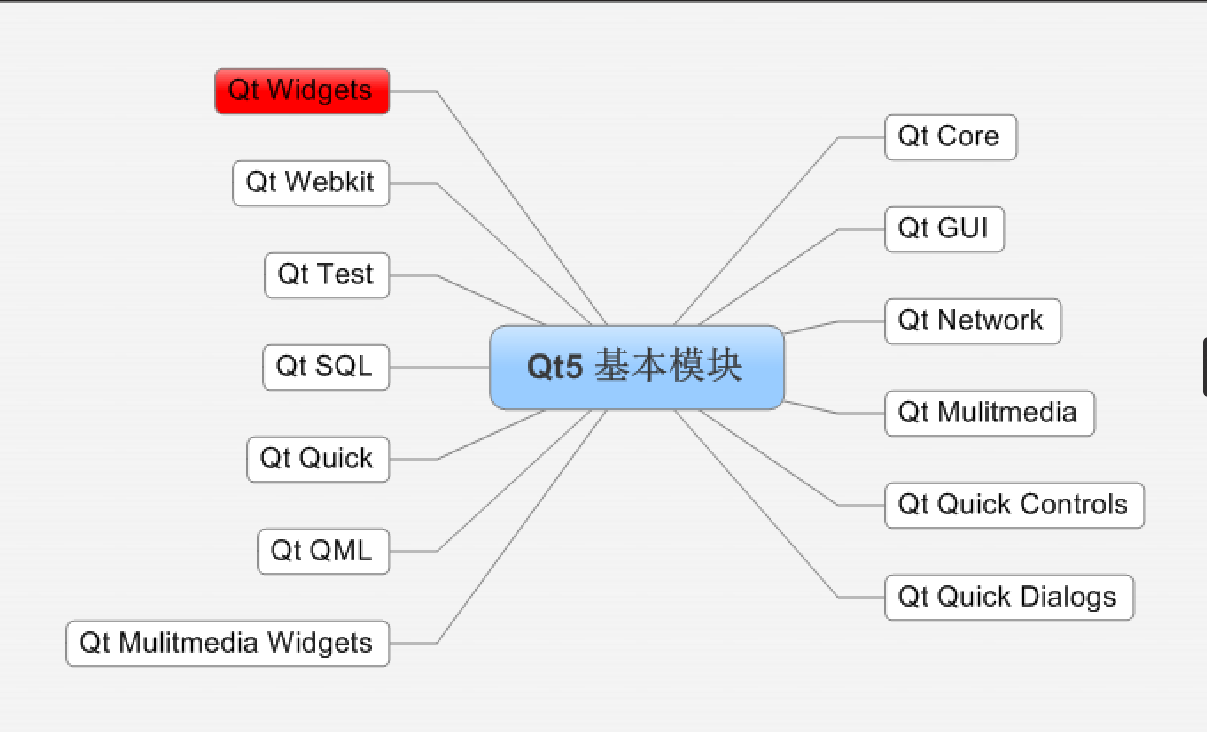
类名是指**继承了基类**的子类类名

需要先选择**基类**，三种选项之间的关系是：



其中，QMainWindow和QDialog都继承自QWidget。

3.



4.快捷键

**运行** Ctrl+R

**编译(构建)** Ctrl+B

**注释** Ctrl+/

**自动对齐** Ctrl+I

**添加定义** alt+回车

5.使用某个官方类时，需要先写入.pro文件的**工程模块变量QT**，再include到.cpp或.h中

Qt中的绝大部分类都继承自QObject，QObject是一个顶层类

6.在Qt中切换项目：右键——设为活动项目

7.每个窗口对象都有一个**Children列表**，父对象在析构时会自动释放子对象

8.信号和槽

信号：各种事件。如按钮被点击时会发出被点击的信号。

槽：响应信号的动作。

想让一个对象收到另一个对象发出的信号，需要**connect()函数**。

9.自定义信号和槽



**emit**用于发出信号

10.Qt中的字符串类型为QString

11.信号与槽扩展

一个信号可以连接多个槽，此时槽函数的调用顺序是**随机的**

一个槽可以连接多个信号

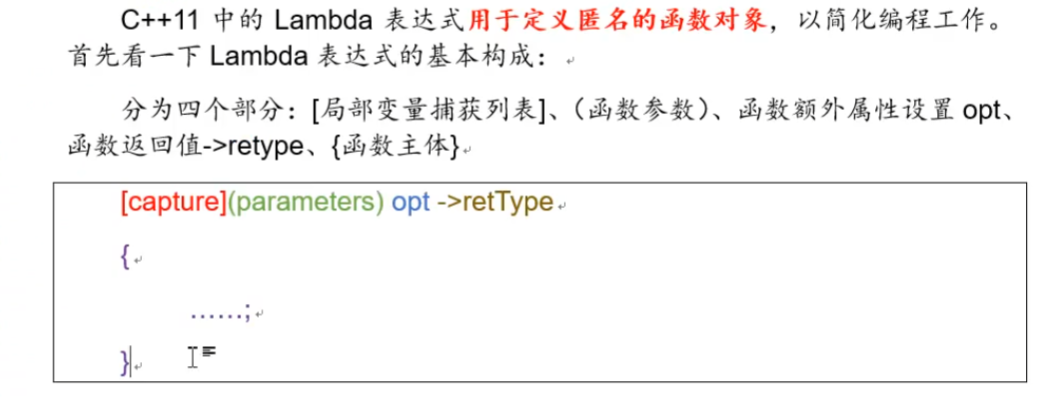
**信号可以连接信号**

信号可以用**disconnect()**断开连接，与connect()参数一样

信号和槽函数的参数类型必须对应，个数可以不同。

信号函数的参数个数>=槽函数参数个数。槽函数的参数会**接收**信号函数的参数。

12.lambda表达式



opt->retType是可选的，标明返回类型，省略时编译器会自动计算

调用的两种方法：

1. 将匿名函数赋值给函数指针，通过函数指针调用

**void(\*p)()=**

**[](){**

**qDebug()<<"Hello lambda";**

**};**

**p();**

1. 直接在定义时，匿名函数后加();

**[](){**

**qDebug()<<"Hello lambda";**

**}()**

未在[]中捕获变量时，不能调用函数外变量

捕获：

int a = 10;

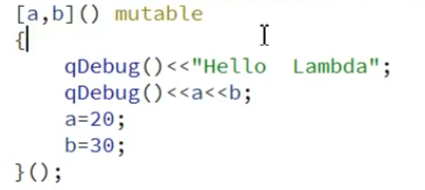
int b = 20;

[a, &b](){ //直接在[]中写变量名表示值传递，写&表示引用传递。

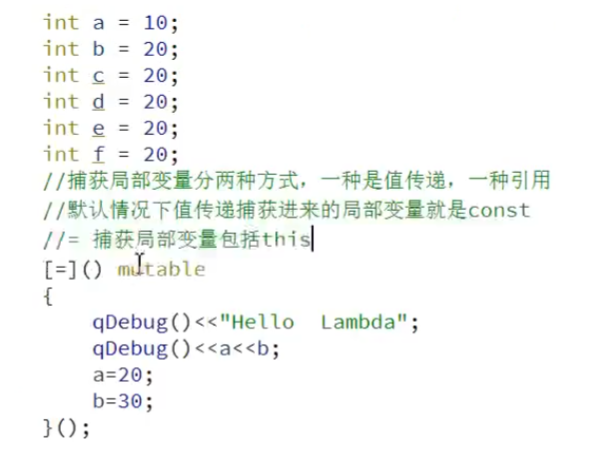
qDebug()<<a;

}

//默认情况下，值传递是const，不能修改。使之能修改的方法是在参数列表()后加mutable关键字：



在[]中使用=表示以值传递的方式捕获所有局部变量



全部引用传递：

&符号



[&, b]表示b值传递，其他引用传递

使用匿名函数：

**QPushButton \*btn = *new* QPushButton("按钮一", *this*);**

**connect(btn, &QPushButton::clicked,**

**[=](){**

**qDebug()<<"空门苍天下第一";**

**}); //无需传入信号接收者，直接定义lambda函数**

使用引用传递&时，实际上还是会传递参数的地址，因此在参数随构造函数消失时，参数的地址指向的值也会改变。

connect()不会随着构造函数的消失而消失

1. QMainWindow



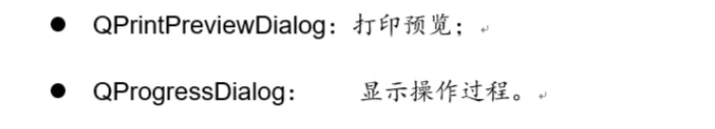
14.dialog对话框

没有最大化、最小化按钮的窗口，分为模态、非模态对话框

模态对话框：对话框未关闭前**不能**操作同一个进程的其他窗口

非模态对话框：对话框未关闭前**也能**操作同一个进程的其他窗口

15. Qt内置对话框



QMessagebox使用：在UI界面中选择转到槽

QMessageBox::information(*this* , "游戏结束", "才...才不会告诉你已经胜利了呢"); //不能点击同进程的其他窗口



1. 布局

布局分为静态布局和动态布局

静态布局：给出这个组件的坐标和长宽值，位置固定

动态布局：布局是动态的，会随着窗口的变化而变化

17.QStringList()<<"英雄"<<"简介"

使用QStringList()创建字符串列表，并使用<<添加元素

18. 容器：

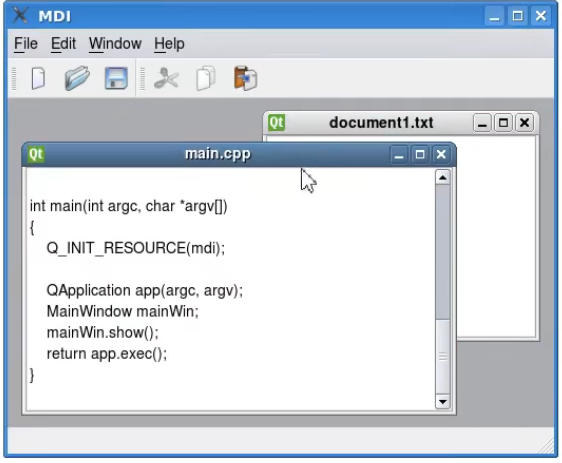
scroll Area 滚动条

Tool box 工具盒子，分页显示

Tab Widget 分栏显示

Stacked Widget 站桩，切换到某一页，在程序运行时不能切换

MDI:在一个程序中能分多个窗口：



Spin Box #计数器

Key Sequence Edit #快捷键

1. 事件
2. QString

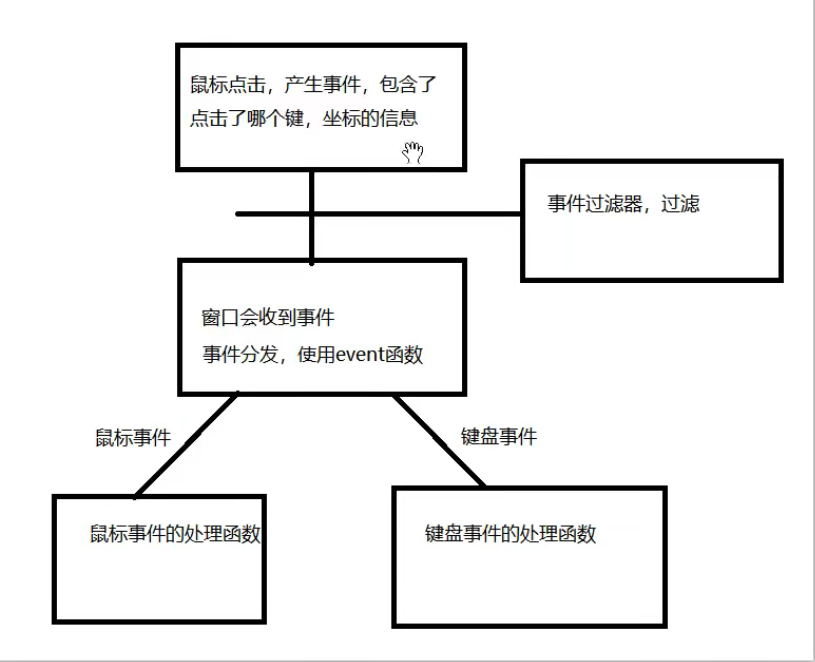
QString会自动将元素加上双引号

x,y都是int类型时：QString("[%1, %2]").arg(x).arg(y) #arg(x)是添加x取代1，arg(y)是添加y取代2。这样最后创建的对象是”x,y”

QString可以用html的语法

QString("<h1><center>[%1, %2]</center></h1>").arg(x).arg(y)

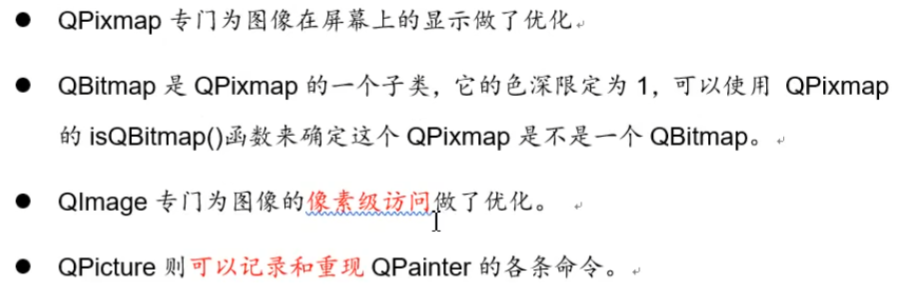
21.



22. 绘图事件

窗口需要重新显示的时候，就会收到一个绘图事件 paintEvent。

绘图设备是指继承了QPainterDevice的子类。Qt提供了5个这样的类：QPixmap、Qbitmap、QImage、QPicture、QWidget。



23.文件操作

QFileInfo info(file); //使用QFileInfo可以获取文件的属性

1. 延时函数QTimer::singleShot()

QTimer::singleShot(250, [=](){

*//翻动上下左右的硬币*

*//下一行*

*if*(row+1 < 4){*this* -> mCoins[row+1][col] -> flip();}

*if*(row -1 >=0){

*//上一行*

*this* -> mCoins[row-1][col] -> flip();

}

*if*(col - 1 >= 0){

*//左边*

*this* -> mCoins[row][col-1] -> flip();

}

*if*(col +1 <4){

*//右边*

*this* -> mCoins[row][col+1] -> flip();

}

});

1. 音效添加

QSound::play(“音效资源”);

1. DLL文件

DLL是动态链接库，可以用于存储编译后的可执行代码、函数、数据以及其他资源

当执行一个可执行文件（.exe）时，有时候会需要依赖一些动态链接库（DLL）文件。这些DLL文件包含了可执行文件所需的代码和函数库，以提供所需的功能和服务。

exe文件运行时，会现在当前exe文件所在的目录寻找DLL文件，找不到后会在系统变量PATH中寻找文件

27.打包发布



选择Release构建后会出现这个文件夹



28.

29.

30.

31.

32.

33.

34.

35.

36.

37.

38.

39.

40.

41.

42.

43.

44.

45.

46.

47.

48.

49.

50.

51.

52.

53.

54.

55.

56.

57.

58.

59.

60.

61.

62.

63.

64.

65.

66.

67.

68.

69.

70.

71.

72.

73.

74.

75.

76.

77.

78.

79.

80.

81.

82.

83.

84.

85.

86.

87.

88.

89.

90.

91.

92.

93.

94.

95.

96.

97.

98.

99.

100.

101.