

Qt –視窗影像處理應用程式 增加幾何轉換處理功能

Yih-Chuan Lin

CSIE Windows Programming Class

National Formosa University



Enhanced Image Processor

- 學習目的
 - 熟練Qt MainWindow之視窗程式開發
 - 熟練Qt QImage、及QPixmap類別製作影像處理視窗程式
 - 學習如何影像幾何轉換處理
 - 學習整合多類別程式開發

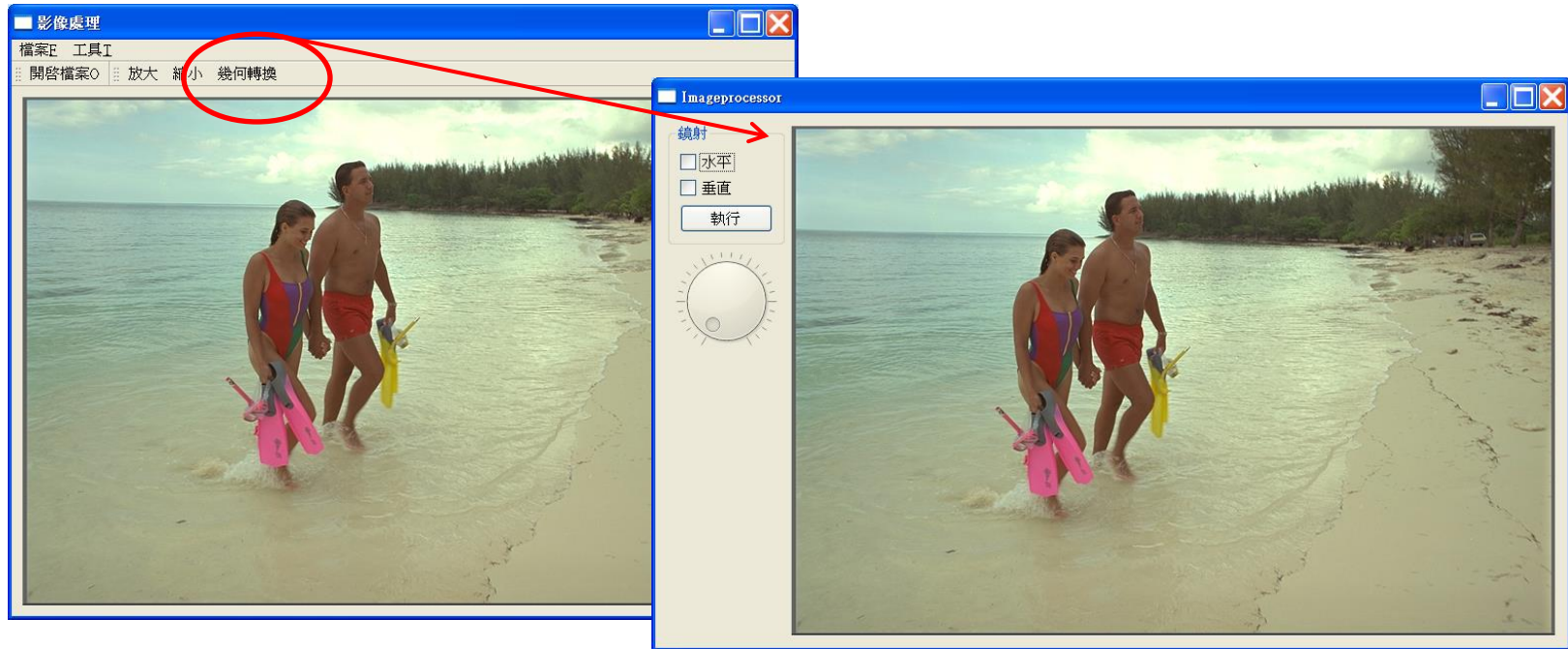




Code less.
Create more.
Deploy everywhere.

Enhanced Image Processor

以Qt 主視窗框架實作 視窗影像處理應用程式



相關類別：

QMainWindow, QMenuBar, QToolBar, QImage, QPixmap, QTransform.





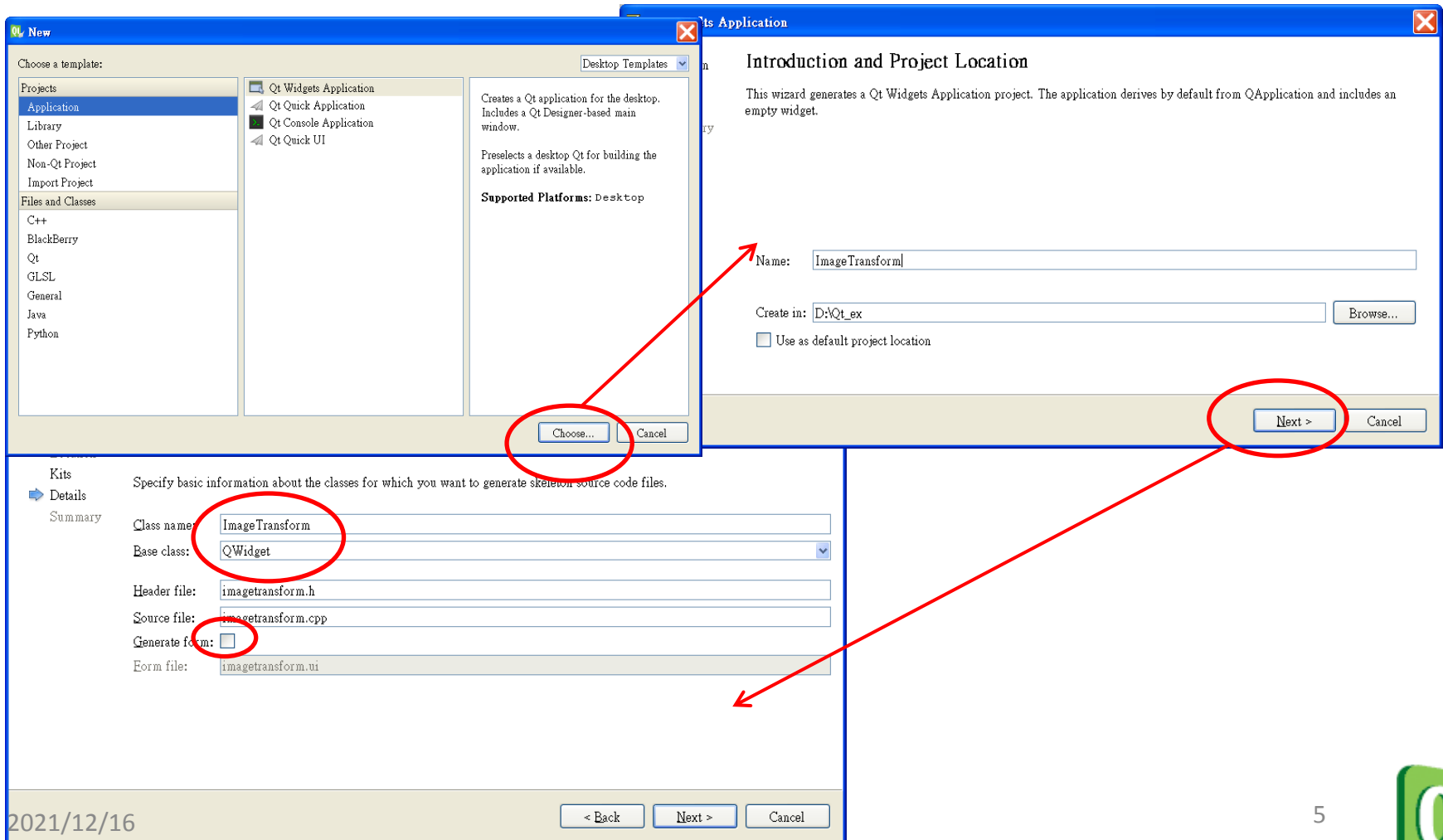
Enhanced Image Processor

- 開發步驟

- 1) 建立 ImageTransform 類別，實作幾何轉換之功能與人機介面。
- 2) 整合 imagetransform.h、及 imagetransform.cpp 檔至 ImageProcessor 專案中。
- 3) 修改 ImageTransform 專案之程式碼，整合新增之幾何轉換功能。



● 建立ImageTransform專案:

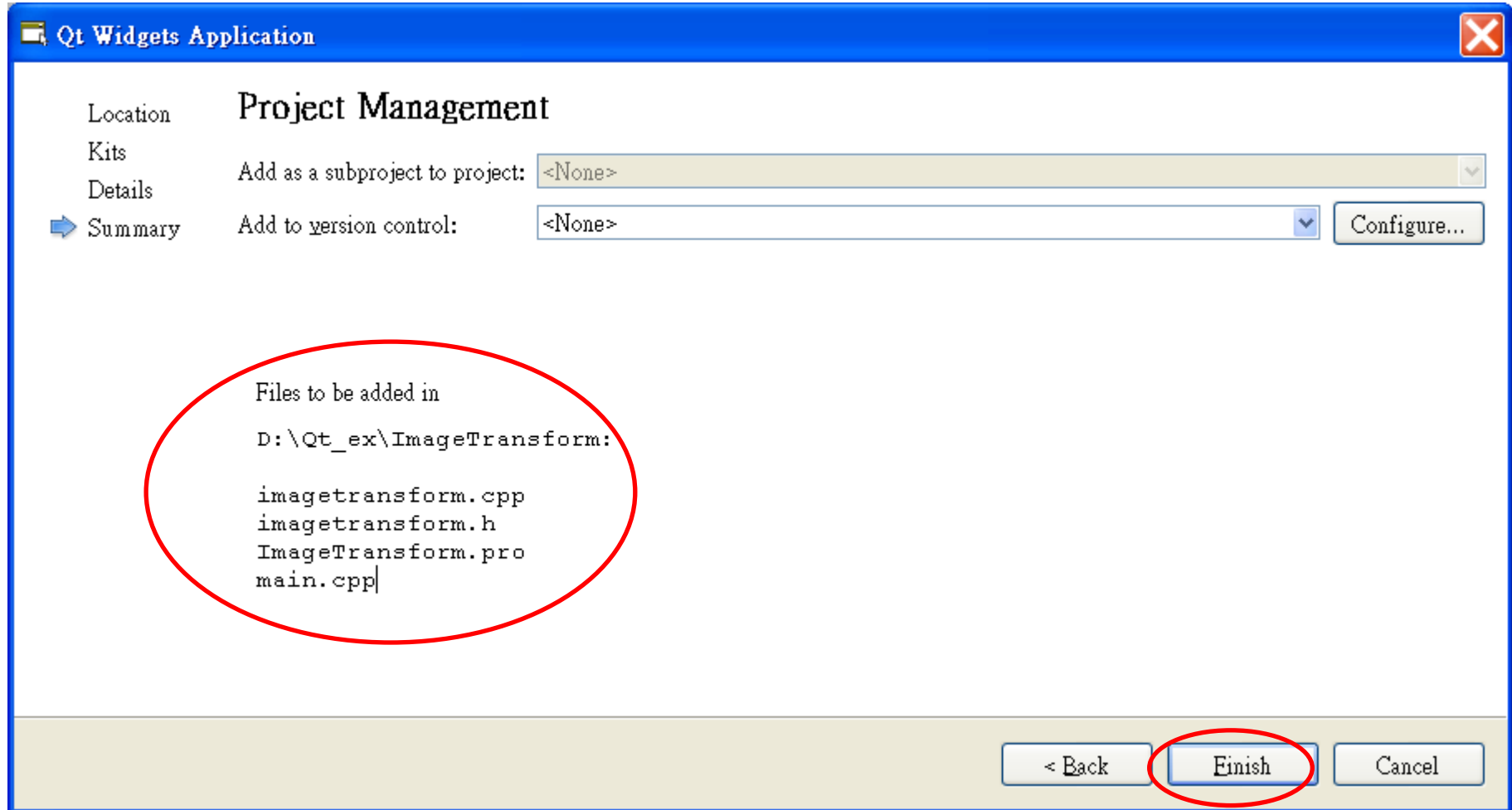




Code less.
Create more.
Deploy everywhere.

Enhanced Image Processor

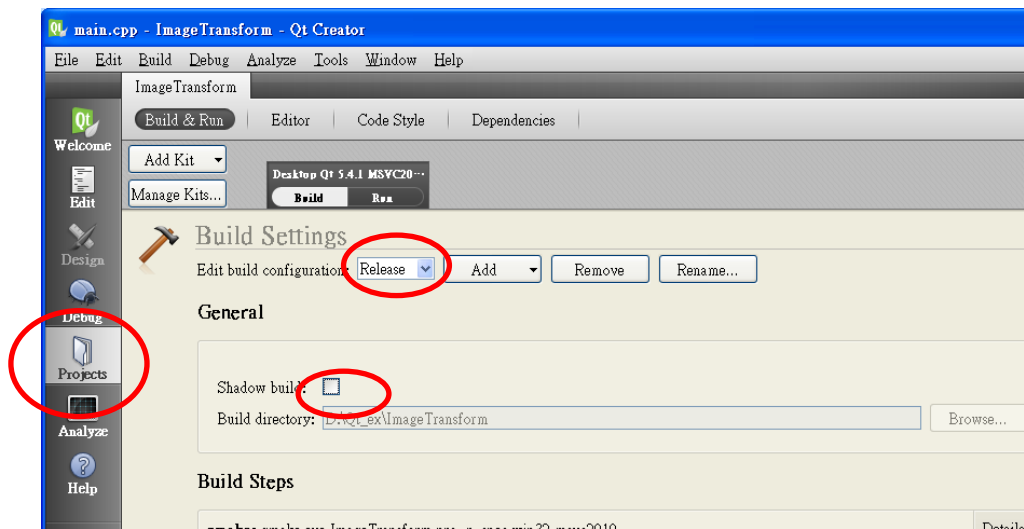
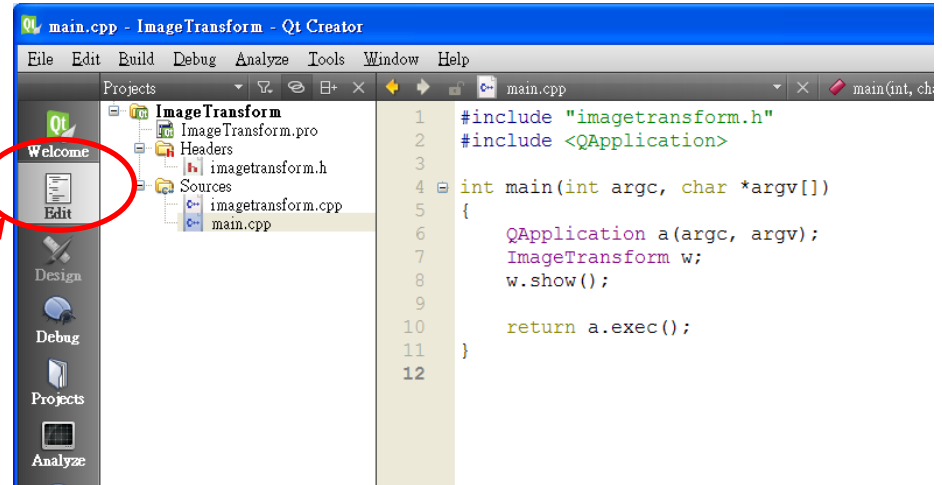
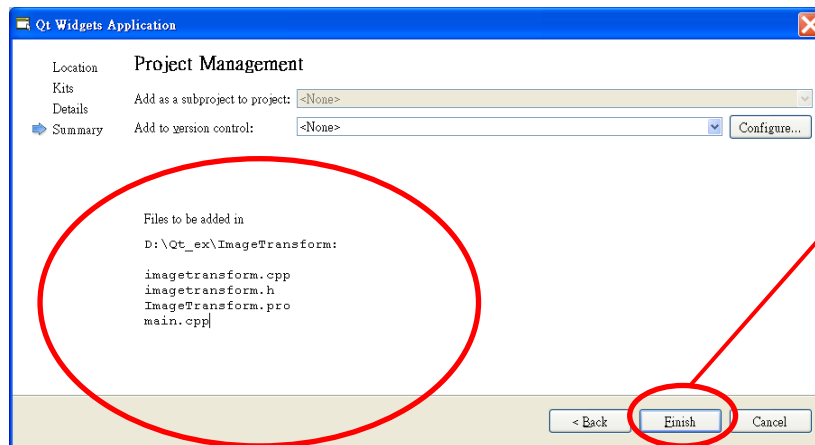
- 建立ImageTransform專案:



Enhanced Image Processor

● 建立ImageTransform專案:

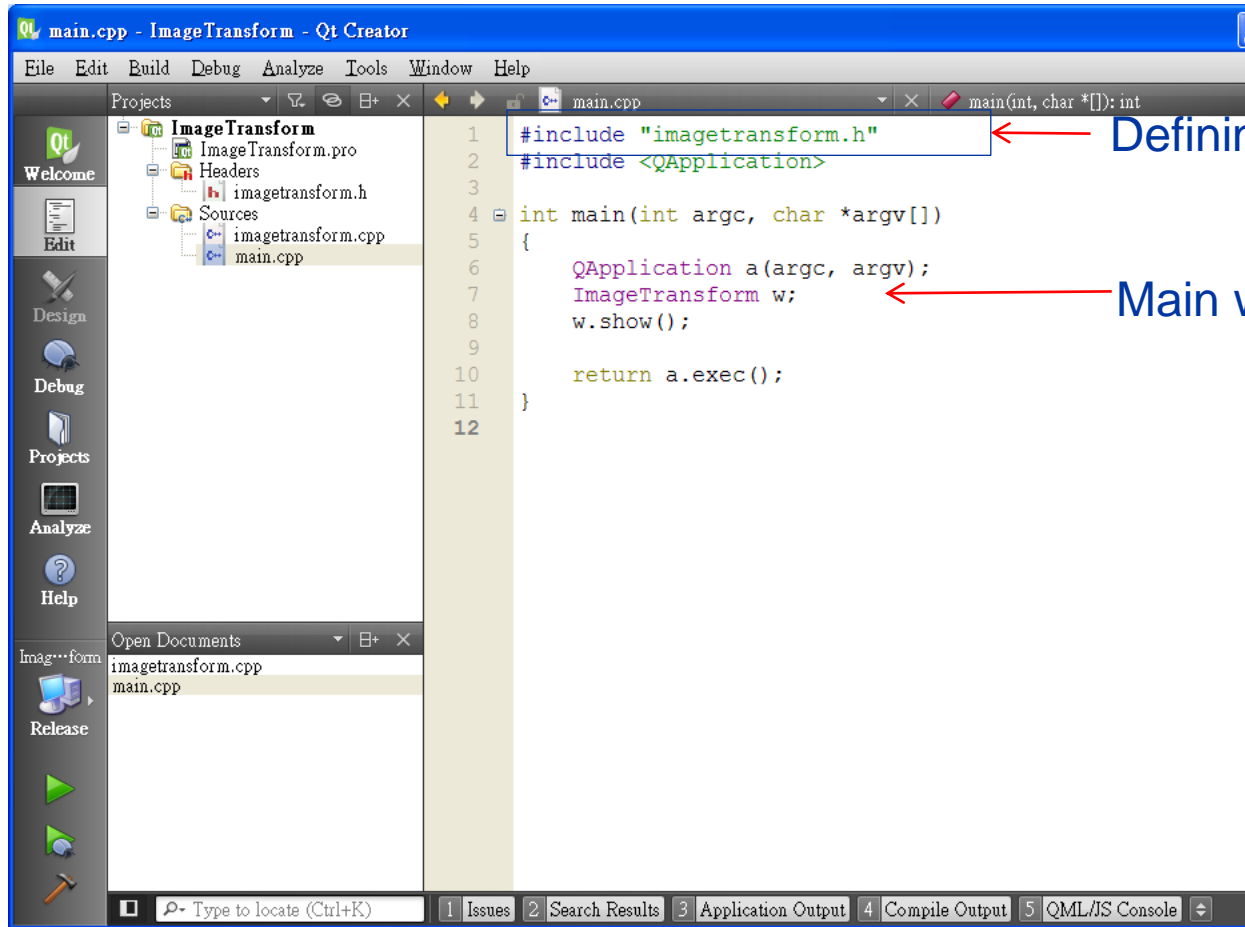
檢視程式進入點



取消分離建置
(Shadow build)

Enhanced Image Processor

- Check out the main function:



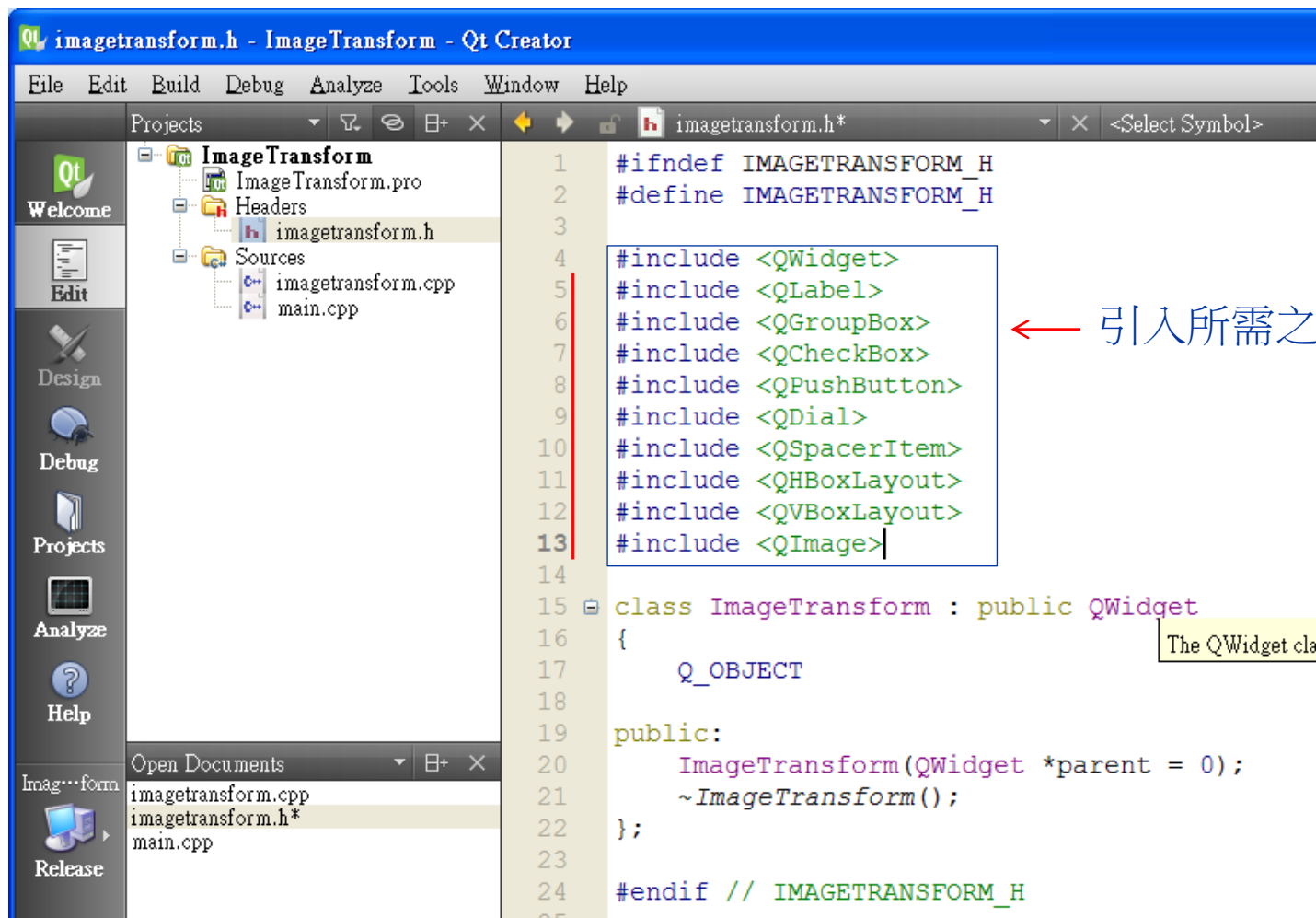
```
1 #include "imagemtransform.h"
2 #include <QApplication>
3
4 int main(int argc, char *argv[])
5 {
6     QApplication a(argc, argv);
7     ImageTransform w;
8     w.show();
9
10    return a.exec();
11 }
12
```

Defining Main window!

Main window object!

Enhanced Image Processor

● Add ImageTransform's members:



The screenshot shows the Qt Creator IDE with the file `imagetransform.h` open. The code is as follows:

```

1  #ifndef IMAGETRANSFORM_H
2  #define IMAGETRANSFORM_H
3
4  #include <QWidget>
5  #include <QLabel>
6  #include <QGroupBox>
7  #include <QCheckBox>
8  #include <QPushButton>
9  #include <QDial>
10 #include <QSpacerItem>
11 #include <QHBoxLayout>
12 #include <QVBoxLayout>
13 #include <QImage>
14
15 class ImageTransform : public QWidget
16 {
17     Q_OBJECT
18
19 public:
20     ImageTransform(QWidget *parent = 0);
21     ~ImageTransform();
22 };
23
24 #endif // IMAGETRANSFORM_H

```

A red arrow points to the include statements (lines 4-13) with the text "引入所需之Qt類別標頭檔". A tooltip for `QWidget` is visible next to line 15.

Enhanced Image Processor

● Add ImageTransform's members:

```

10 #include <QSpacerItem>
11 #include <QHBoxLayout>
12 #include <QVBoxLayout>
13 #include <QImage>
14
15 class ImageTransform : public QWidget
16 {
17     Q_OBJECT
18
19 public:
20     ImageTransform(QWidget *parent = 0);
21     ~ImageTransform();
22     QLabel        *inWin;
23     QGroupBox      *mirrorGroup;
24     QCheckBox      *hCheckBox;
25     QCheckBox      *vCheckBox;
26     QPushButton   *mirrorButton;
27     QDial          *rotateDial;
28     QSpacerItem    *vSpacer;
29     QHBoxLayout    *mainLayout;
30     QVBoxLayout    *groupLayout;
31     QVBoxLayout    *leftLayout;
32     QImage         srcImg;
33     QImage         dstImg;
34 };
35
36 #endif // IMAGETRANSFORM_H
  
```

宣告Qt類別成員變數

Enhanced Image Processor

● Add ImageTransform's members:

Qt Creator interface showing the `imagemagick.h` file. The code defines the `ImageTransform` class, which inherits from `QWidget`. The class includes member variables for UI elements and declares private slots for image processing.

```

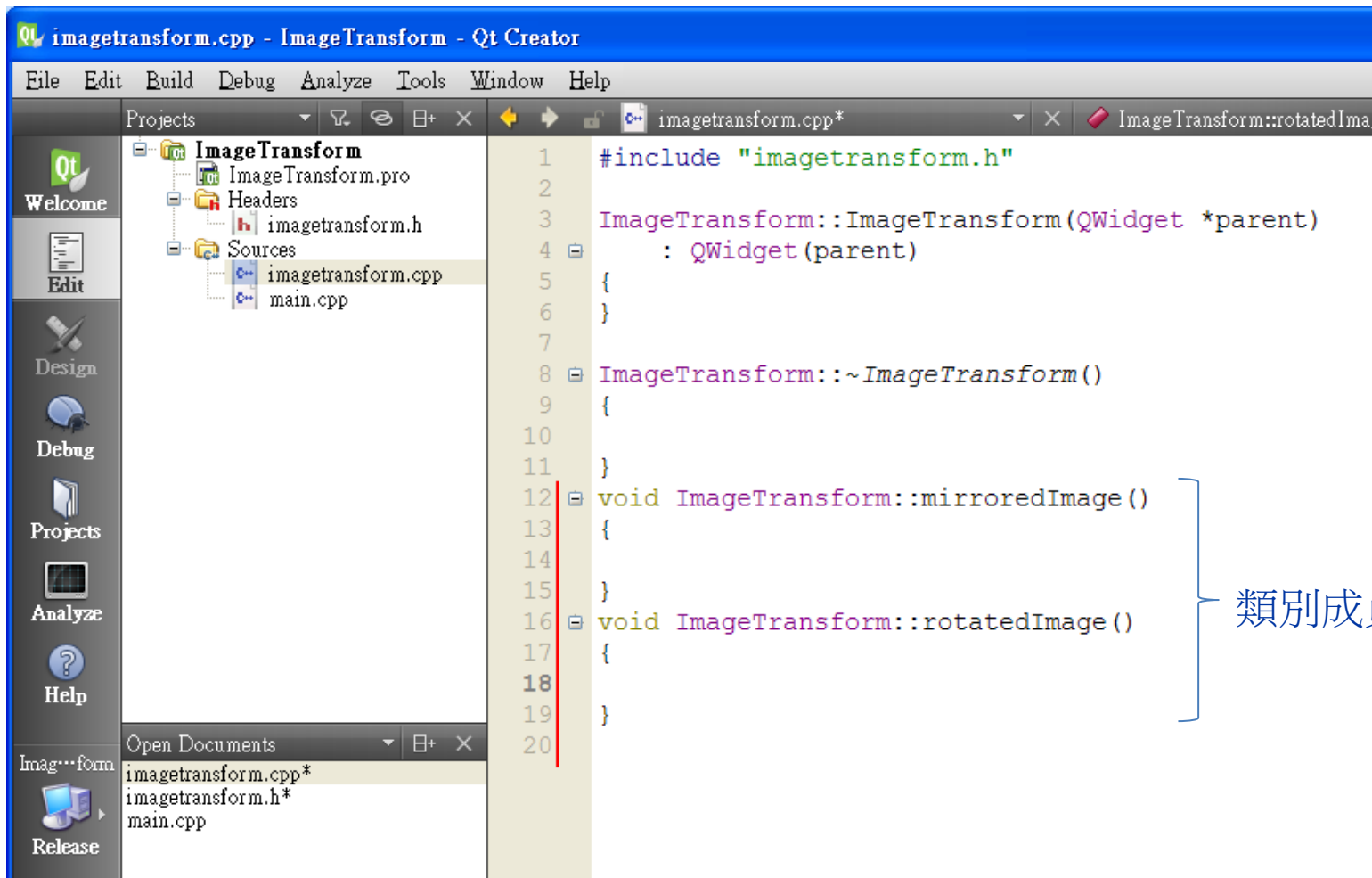
15 class ImageTransform : public QWidget
16 {
17     Q_OBJECT
18
19 public:
20     ImageTransform(QWidget *parent = 0);
21     ~ImageTransform();
22     QLabel *inWin;
23     QGroupBox *mirrorGroup;
24     QCheckBox *hCheckBox;
25     QCheckBox *vCheckBox;
26     QPushButton *mirrorButton;
27     QDial *rotateDial;
28     QSpacerItem *vSpacer;
29     QHBoxLayout *mainLayout;
30     QVBoxLayout *groupLayout;
31     QVBoxLayout *leftLayout;
32     QImage srcImg;
33     QImage dstImg;
34
35 private slots:
36     void mirroredImage();
37     void rotatedImage();
38 };
39
40 #endif // IMAGETRANSFORM_H

```

宣告Qt類別槽函數

Enhanced Image Processor

- Add ImageTransform's members:



```
1 #include "imagemtransform.h"
2
3 ImageTransform::ImageTransform(QWidget *parent)
4     : QWidget(parent)
5 {
6 }
7
8 ImageTransform::~ImageTransform()
9 {
10 }
11
12 void ImageTransform::mirroredImage ()
13 {
14 }
15
16 void ImageTransform::rotatedImage ()
17 {
18 }
19
20
```

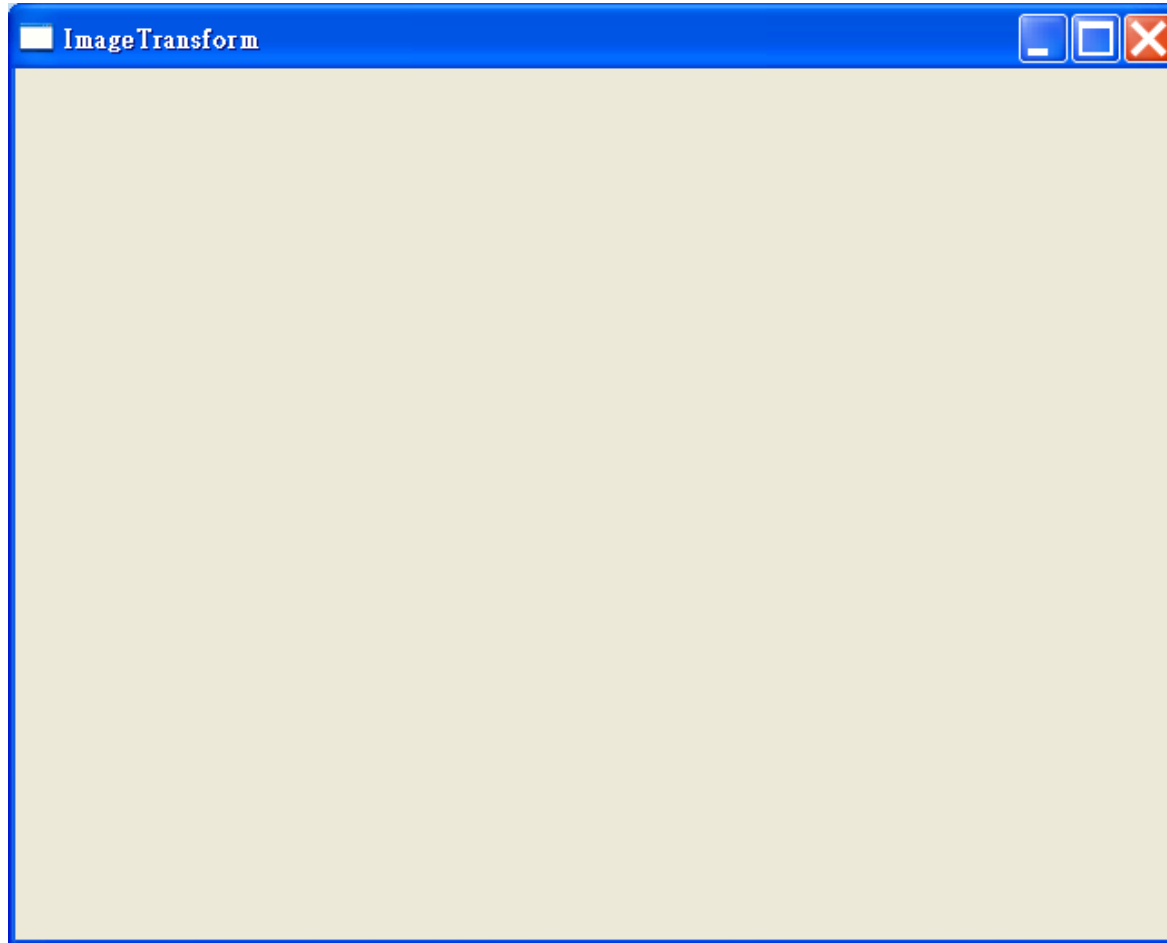
類別成員函數



Code less.
Create more.
Deploy everywhere.

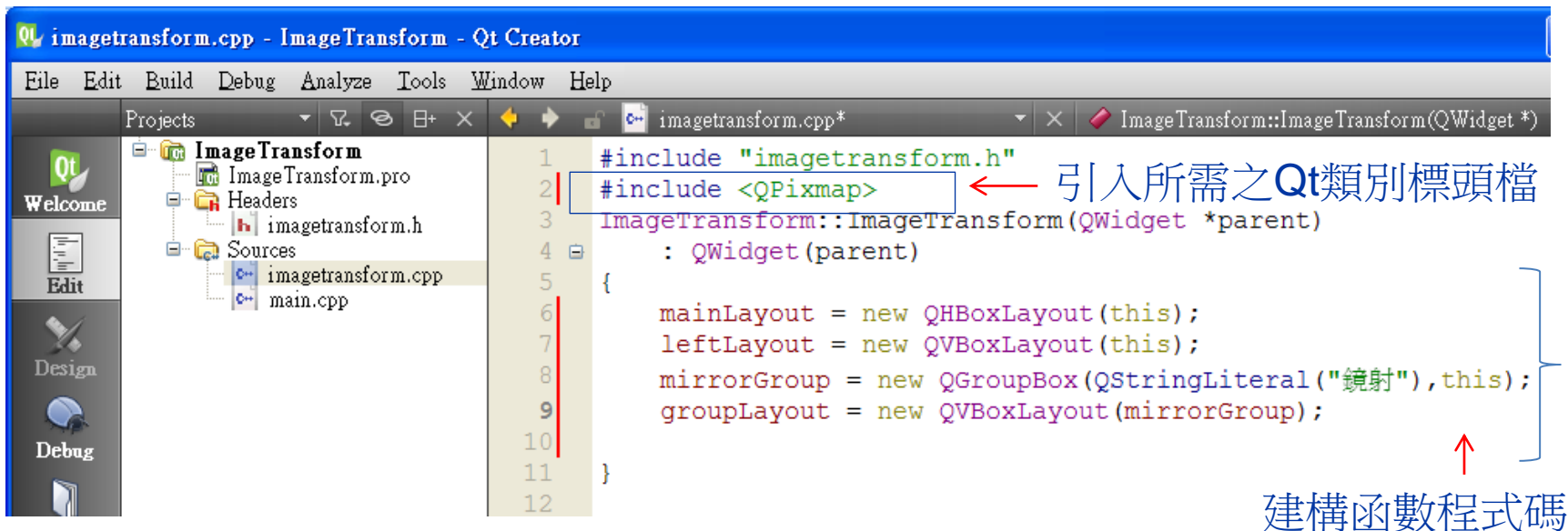
Enhanced Image Processor

- **Build and execute the program:**



Enhanced Image Processor

- Implement ImageTransform's constructor:



imagertransform.cpp - ImageTransform - Qt Creator

File Edit Build Debug Analyze Tools Window Help

Projects

- ImageTransform
 - ImageTransform.pro
 - Headers
 - imagertransform.h
 - Sources
 - imagertransform.cpp
 - main.cpp

```
1 #include "imagertransform.h"
2 #include <QPixmap>
3 ImageTransform::ImageTransform(QWidget *parent)
4     : QWidget(parent)
5 {
6     mainLayout = new QHBoxLayout(this);
7     leftLayout = new QVBoxLayout(this);
8     mirrorGroup = new QGroupBox(QStringLiteral("鏡射"), this);
9     groupLayout = new QVBoxLayout(mirrorGroup);
10
11 }
12
```

引入所需之Qt類別標頭檔

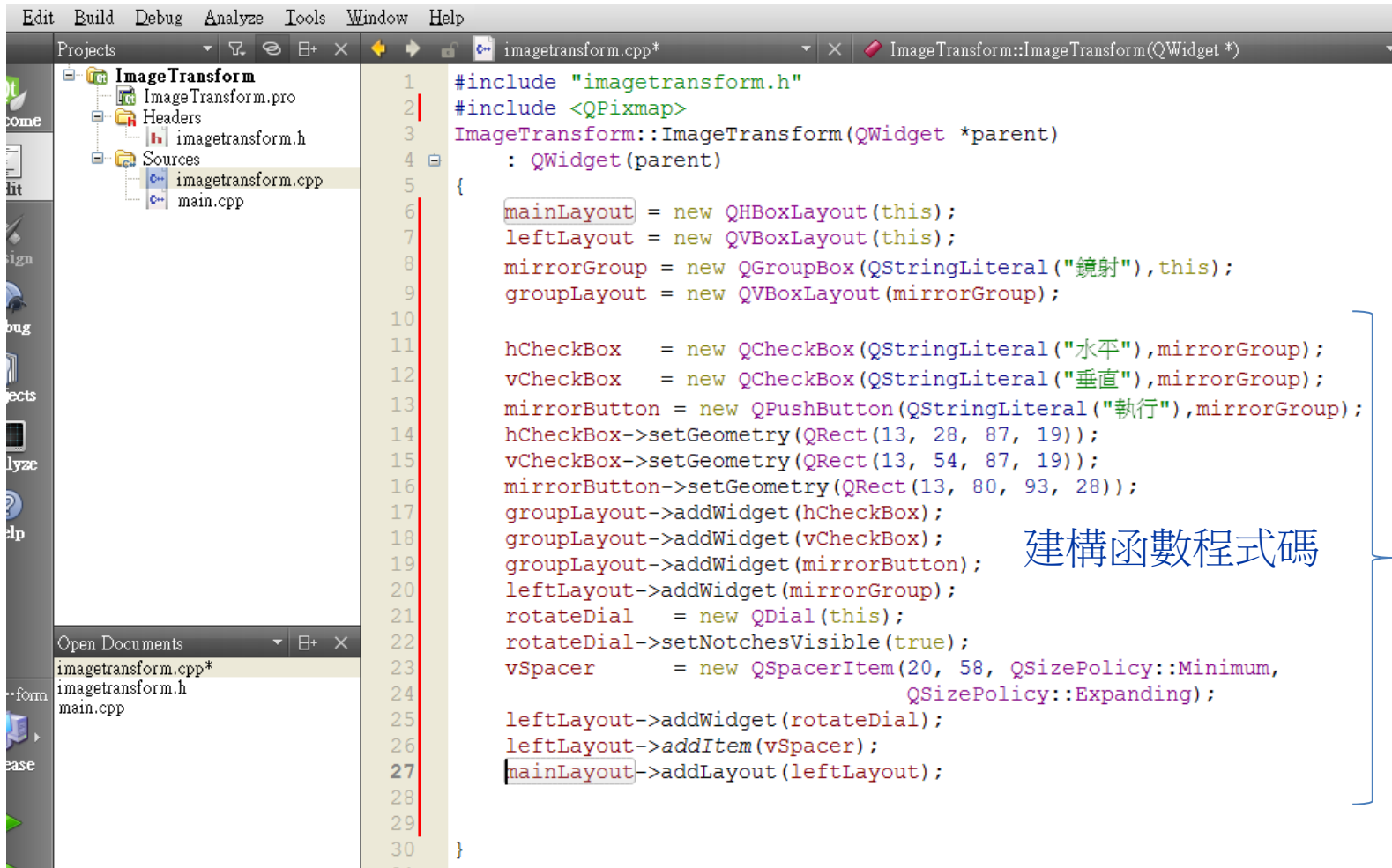
建構函數程式碼



Code less.
Create more.
Deploy everywhere.

Enhanced Image Processor

● Implement ImageTransform's member functions:



```
1 #include "imagertransform.h"
2 #include <QPixmap>
3 ImageTransform::ImageTransform(QWidget *parent)
4     : QWidget(parent)
5 {
6     mainLayout = new QHBoxLayout(this);
7     leftLayout = new QVBoxLayout(this);
8     mirrorGroup = new QGroupBox(QStringLiteral("鏡射"), this);
9     groupLayout = new QVBoxLayout(mirrorGroup);
10
11     hCheckBox = new QCheckBox(QStringLiteral("水平"), mirrorGroup);
12     vCheckBox = new QCheckBox(QStringLiteral("垂直"), mirrorGroup);
13     mirrorButton = new QPushButton(QStringLiteral("執行"), mirrorGroup);
14     hCheckBox->setGeometry(QRect(13, 28, 87, 19));
15     vCheckBox->setGeometry(QRect(13, 54, 87, 19));
16     mirrorButton->setGeometry(QRect(13, 80, 93, 28));
17     groupLayout->addWidget(hCheckBox);
18     groupLayout->addWidget(vCheckBox);
19     groupLayout->addWidget(mirrorButton);
20     leftLayout->addWidget(mirrorGroup);
21     rotateDial = new QDial(this);
22     rotateDial->setNotchesVisible(true);
23     vSpacer = new QSpacerItem(20, 58, QSizePolicy::Minimum,
24                             QSizePolicy::Expanding);
25     leftLayout->addWidget(rotateDial);
26     leftLayout->addItem(vSpacer);
27     mainLayout->addLayout(leftLayout);
28
29 }
30
```

建構函數程式碼



Enhanced Image Processor

● Implement ImageTransform's member functions:



```

27     mainLayout->addLayout(leftLayout);
28
29     inWin = new QLabel(this);
30     inWin->setScaledContents(true);
31     QPixmap *initPixmap= new QPixmap(300,200);
32     initPixmap->fill(QColor(255,255,255));
33     inWin->setPixmap(*initPixmap);
34     inWin->setSizePolicy(QSizePolicy::Expanding, QSizePolicy::Expanding);
35     if (srcImg.isNull())
36     {
37         QPixmap *initPixmap= new QPixmap(300,200);
38         initPixmap->fill(QColor(255,255,255));
39         inWin->setPixmap(*initPixmap);
40     }
41     mainLayout->addWidget(inWin);
42     connect(mirrorButton,SIGNAL(clicked()),this,SLOT(mirroredImage()));
43     connect(rotateDial,SIGNAL(valueChanged(int)),this,SLOT(rotatedImage()));
44 }
45
46 ImageTransform::~ImageTransform()
47 {
48 }
49
50 void ImageTransform::mirroredImage()
51 {
52 }
53
54 void ImageTransform::rotatedImage()
55 {
56 }
57

```

建構函數程式碼



Code less.
Create more.
Deploy everywhere.

Enhanced Image Processor

- Build and execute the program:





Code less.
Create more.
Deploy everywhere.

Enhanced Image Processor

- Implement ImageTransform's member functions:

```
35 if (srcImg.isNull())
36 {
37     QPixmap *initPixmap= new QPixmap(300,200);
38     initPixmap->fill(QColor(255,255,255));
39     inWin->setPixmap(*initPixmap);
40 }
41 mainLayout->addWidget(inWin);
42 connect (mirrorButton,SIGNAL(clicked()),this,SLOT(mirroredImage()));
43 connect (rotateDial,SIGNAL(valueChanged(int)),this,SLOT(rotatedImage()));
44 }
45
46 ImageTransform::~ImageTransform()
47 {
48 }
49
50 void ImageTransform::mirroredImage()
51 {
52     bool H,V;
53     if (srcImg.isNull())
54         return;
55     H=hCheckBox->isChecked();|
56     V=vCheckBox->isChecked();
57     dstImg = srcImg.mirrored(H,V);
58     inWin->setPixmap(QPixmap::fromImage(dstImg));
59     srcImg = dstImg;
60 }
61
62 void ImageTransform::rotatedImage()
63 {
```

鏡射槽函數程式碼





Code less.
Create more.
Deploy everywhere.

Enhanced Image Processor

- Implement ImageTransform's member functions:

```
48 }
49 }
50 void ImageTransform::mirroredImage()
51 {
52     bool H,V;
53     if (srcImg.isNull())
54         return;
55     H=hCheckBox->isChecked();
56     V=vCheckBox->isChecked();
57     dstImg = srcImg.mirrored(H,V);
58     inWin->setPixmap(QPixmap::fromImage(dstImg));
59     srcImg = dstImg;
60 }
61
62 void ImageTransform::rotatedImage()
63 {
64     QTransform tran;
65     int angle;
66     if (srcImg.isNull())
67         return;
68     angle = rotateDial->value();
69     tran.rotate(angle);
70     dstImg = srcImg.transformed(tran);
71     inWin->setPixmap(QPixmap::fromImage(dstImg));
72 }
73 }
74
```

旋轉槽函數程式碼

