Qt –視窗影像處理應用程式

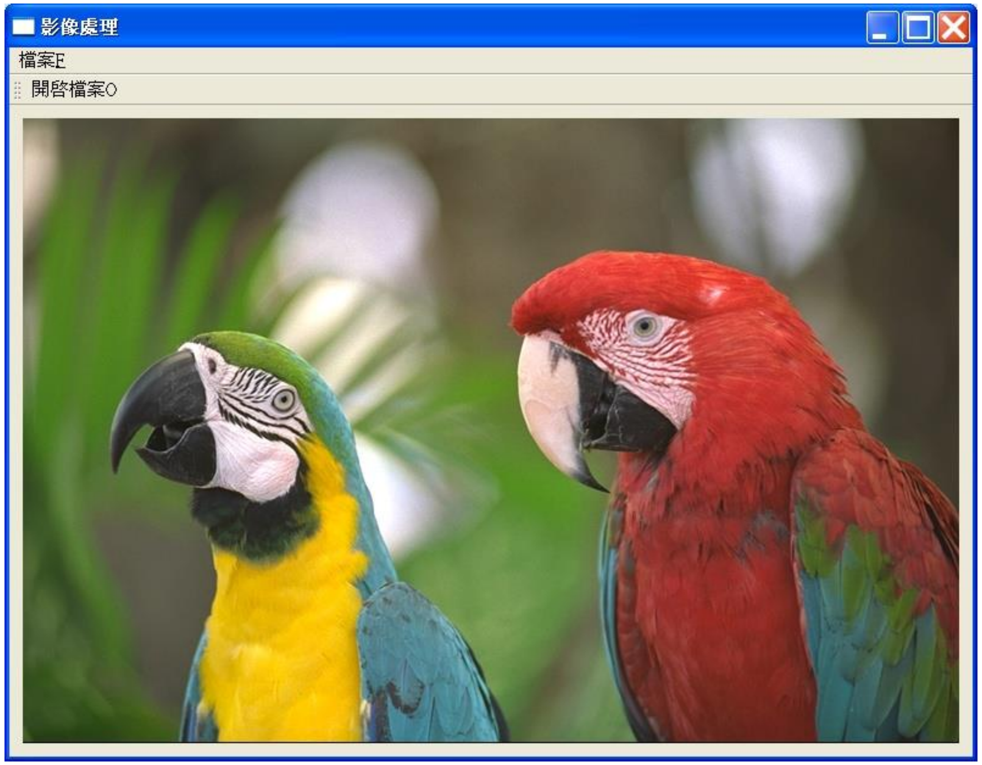
Yih-Chuan Lin
CSIE Windows Programming Class
National Formosa University

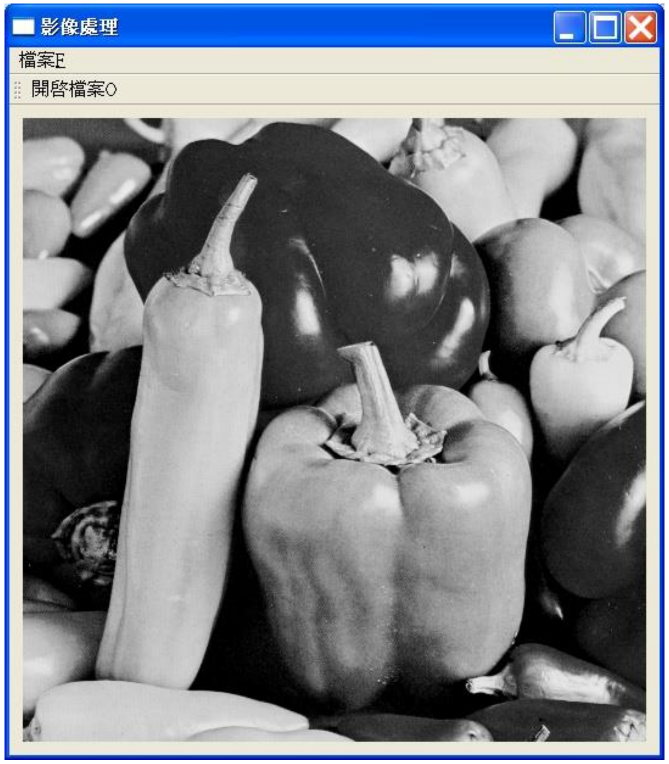
Image Processor

- 學習目的
  - 熟練Qt MainWindow之視窗程式開發
  - 練習Qt Qimage、及QPixmap類別製作影像處理視窗程式
  - 學習如何以程式碼建立視窗人機介面

Image Processor

以Qt 主視窗框架實作 視窗影像處理應用程式

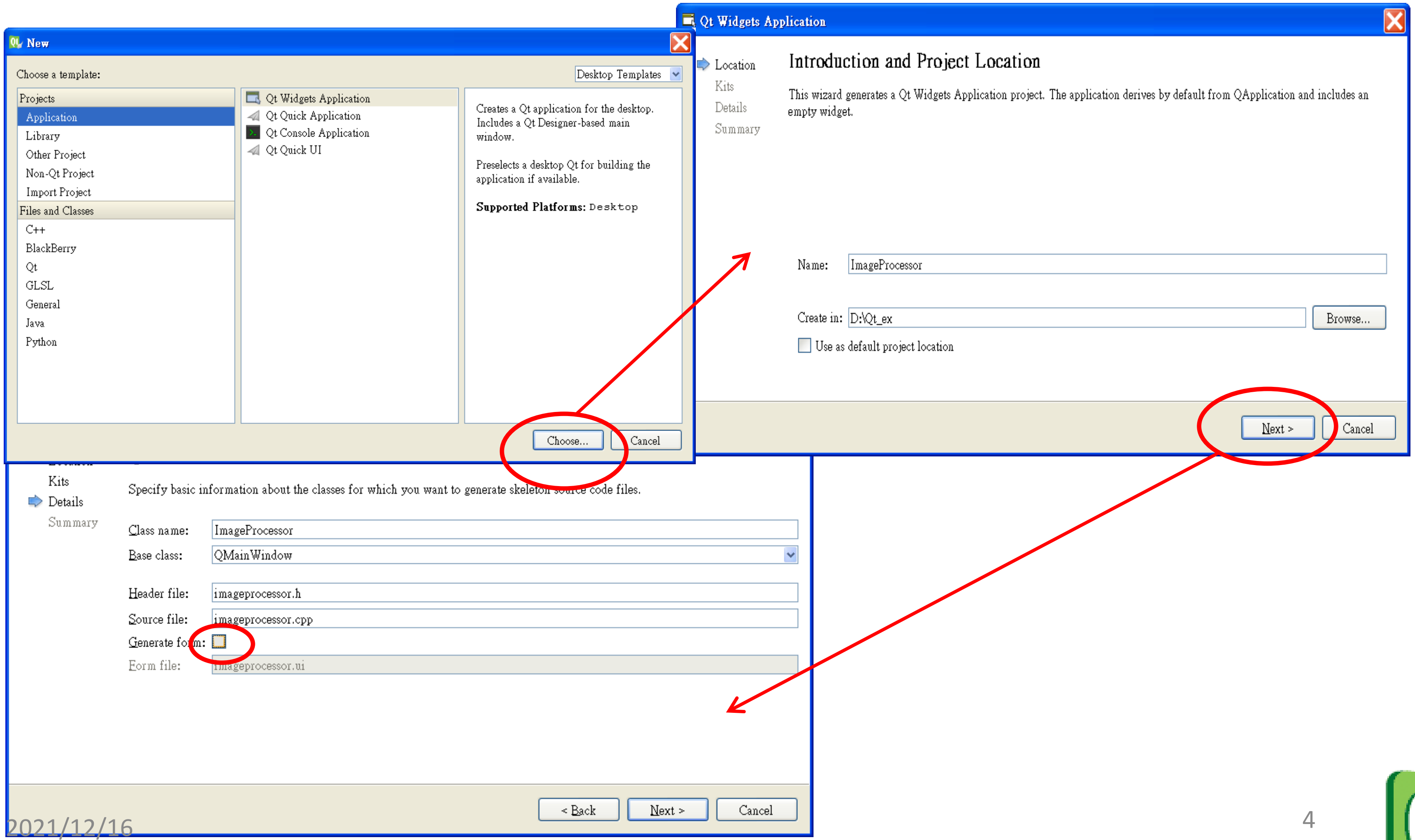




相關類別：
QMainWindow, QMenuBar, QToolBar, QImage, QPixmap.

Image Processor

- **Create a mainwindow application:**



Creates a Qt application for the desktop. Includes a Qt Designer-based main window.

Preselects a desktop Qt for building the application if available.

Supported Platforms: Desktop

Qt Widgets Application

Introduction and Project Location

This wizard generates a Qt Widgets Application project. The application derives by default from QApplication and includes an empty widget.

Name: ImageProcessor

Create in: D:\Qt_ex

Use as default project location

Specify basic information about the classes for which you want to generate skeleton source code files.

Class name: ImageProcessor

Base class: QMainWindow

Header file: imageprocessor.h

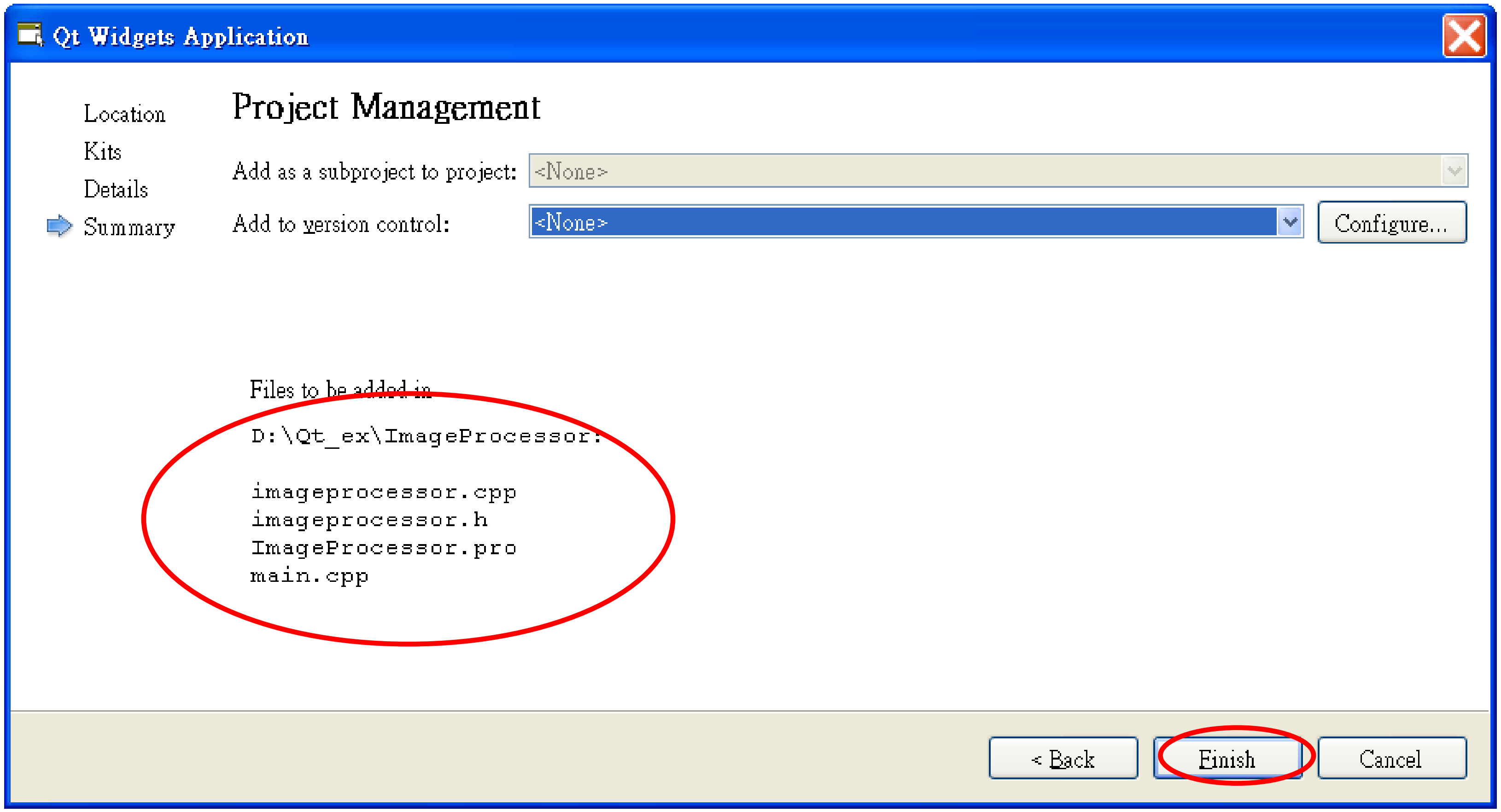
Source file: imageprocessor.cpp

Generate form: ☐

Form file: imageprocessor.ui

Image Processor

- **Create a mainwindow application:**


Qt Widgets Application

Location
Kits
Details
Summary

Project Management

Add as a subproject to project: <None>

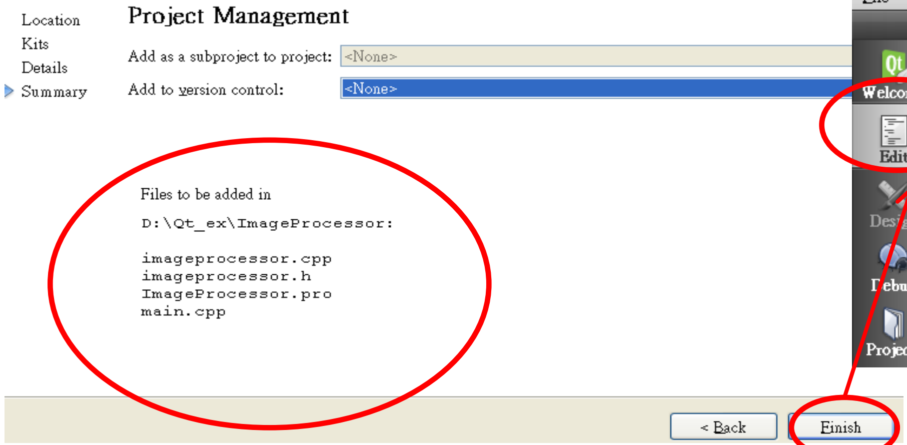
Add to version control: <None>

Files to be added in

```
D:\Qt_ex\ImageProcessor:

imageprocessor.cpp
imageprocessor.h
ImageProcessor.pro
main.cpp
```

Qt- MainWindow

- **Create a mainwindow application:**



Location
Kits
Details
Summary

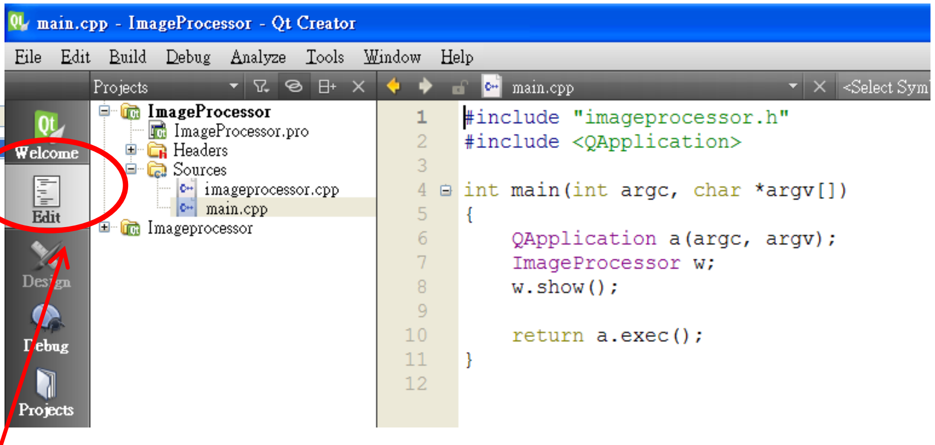
Project Management

Add as a subproject to project: <None>
Add to version control: <None>

```
Files to be added in
D:\Qt_ex\ImageProcessor:

imageprocessor.cpp
imageprocessor.h
ImageProcessor.pro
main.cpp
```

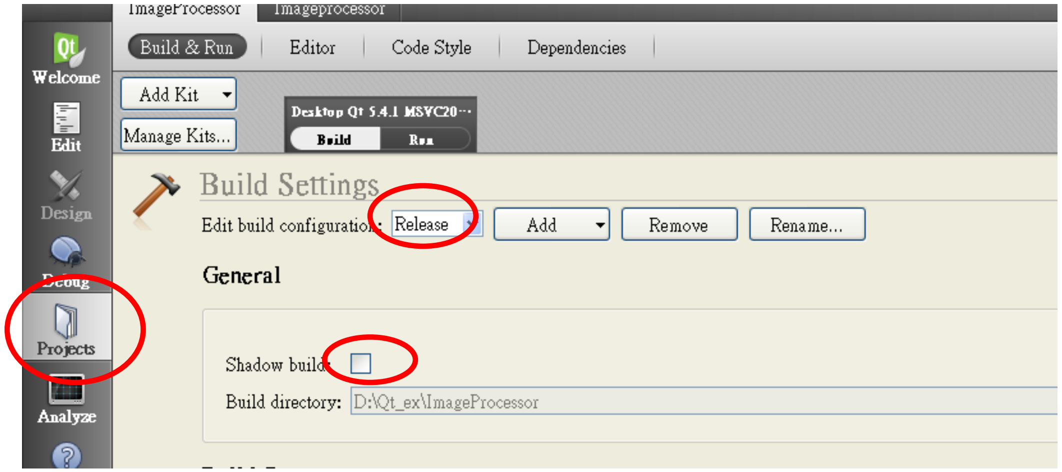
< Back   Finish

檢視程式進入點



main.cpp - ImageProcessor - Qt Creator

```
#include "imageprocessor.h"
#include <QApplication>

int main(int argc, char *argv[])
{
    QApplication a(argc, argv);
    ImageProcessor w;
    w.show();

    return a.exec();
}
```



Build Settings
Edit build configuration: Release

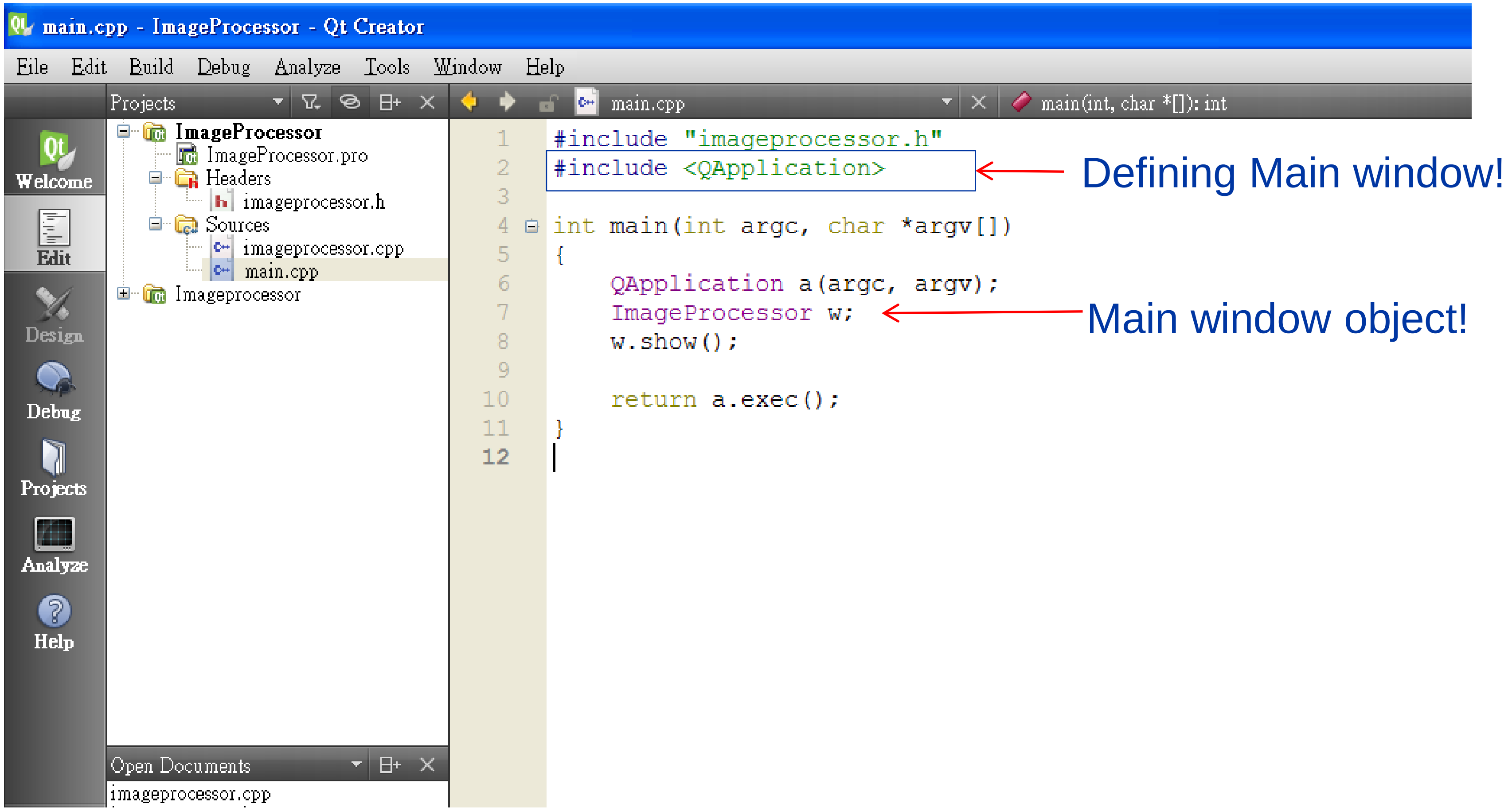
General

Shadow build: ☐
Build directory: D:\Qt_ex\ImageProcessor

取消分離建置
(Shadow build)

Image Processor

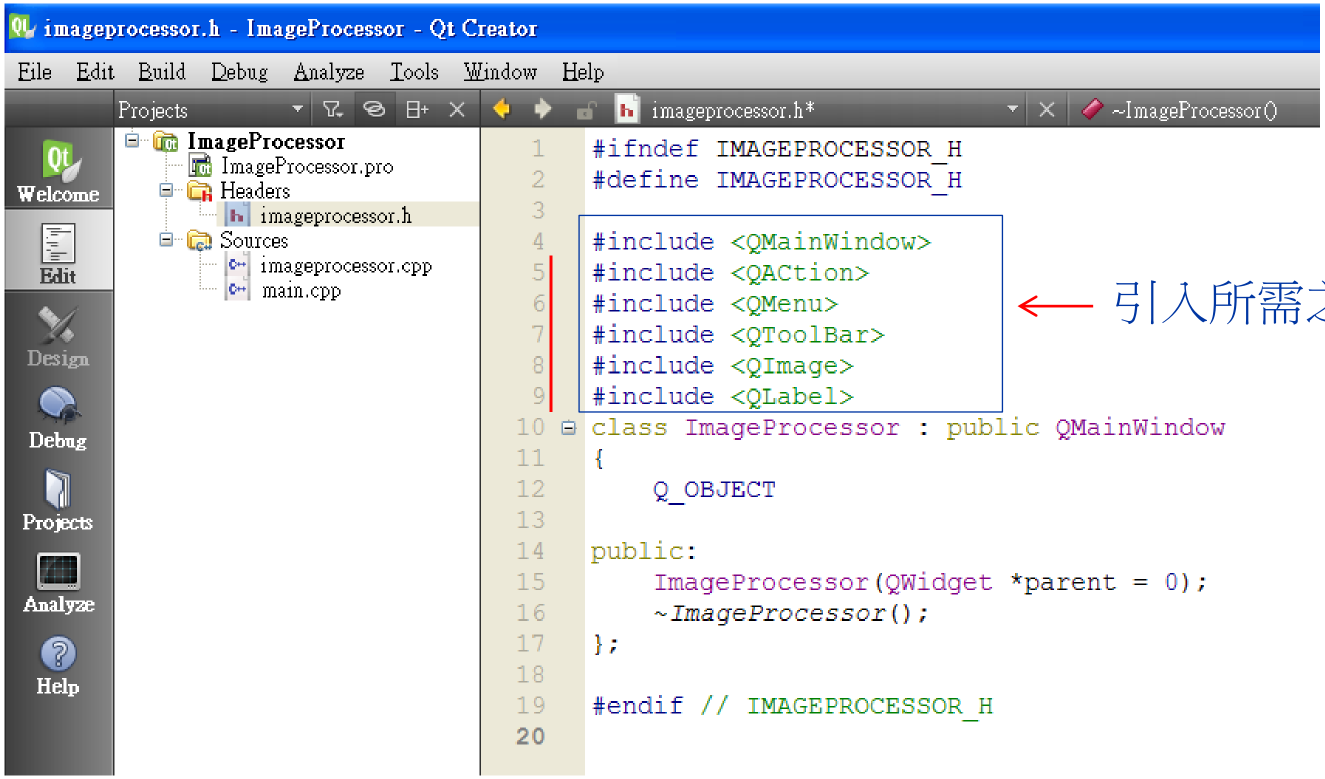
- **Check out the main function:**



```cpp
#include "imageprocessor.h"
#include <QApplication>

int main(int argc, char *argv[])
{
    QApplication a(argc, argv);
    ImageProcessor w;
    w.show();

    return a.exec();
}
```

Image Processor

- **Add ImageProcessor's members:**



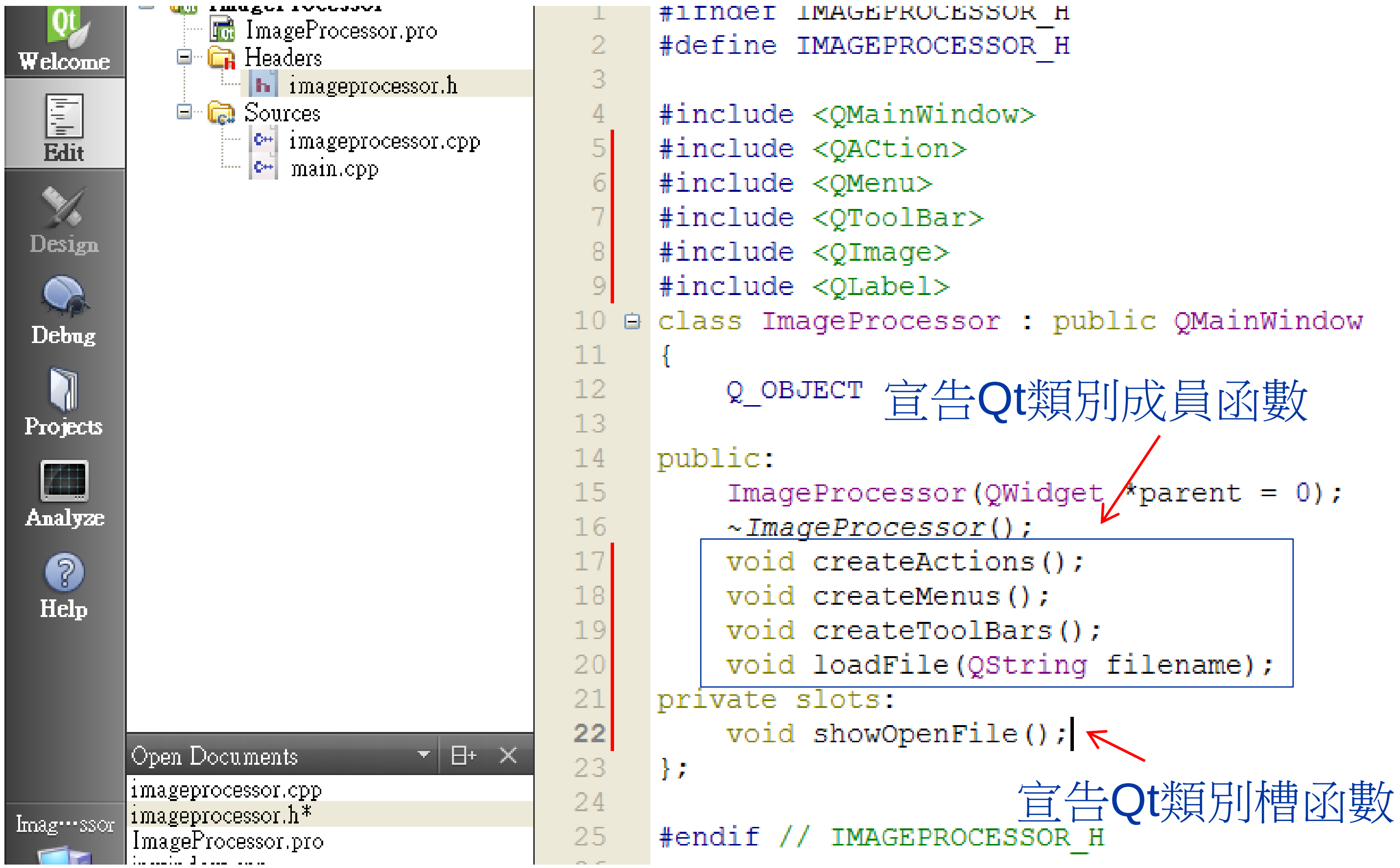
```
#ifndef IMAGEPROCESSOR_H
#define IMAGEPROCESSOR_H

#include <QMainWindow>
#include <QAction>
#include <QMenu>
#include <QToolBar>
#include <QImage>
#include <QLabel>
class ImageProcessor : public QMainWindow
{
    Q_OBJECT

public:
    ImageProcessor(QWidget *parent = 0);
    ~ImageProcessor();
};

#endif // IMAGEPROCESSOR_H
```

← 引入所需之Qt類別標頭檔

2021/12/16

Image Processor

- **Add ImageProcessor's members:**



```
#ifndef IMAGEPROCESSOR_H
#define IMAGEPROCESSOR_H

#include <QMainWindow>
#include <QAction>
#include <QMenu>
#include <QToolBar>
#include <QImage>
#include <QLabel>
class ImageProcessor : public QMainWindow
{
    Q_OBJECT

public:
    ImageProcessor(QWidget *parent = 0);
    ~ImageProcessor();
    void createActions();
    void createMenus();
    void createToolBars();
    void loadFile(QString filename);
private slots:
    void showOpenFile();
};

#endif // IMAGEPROCESSOR_H
```

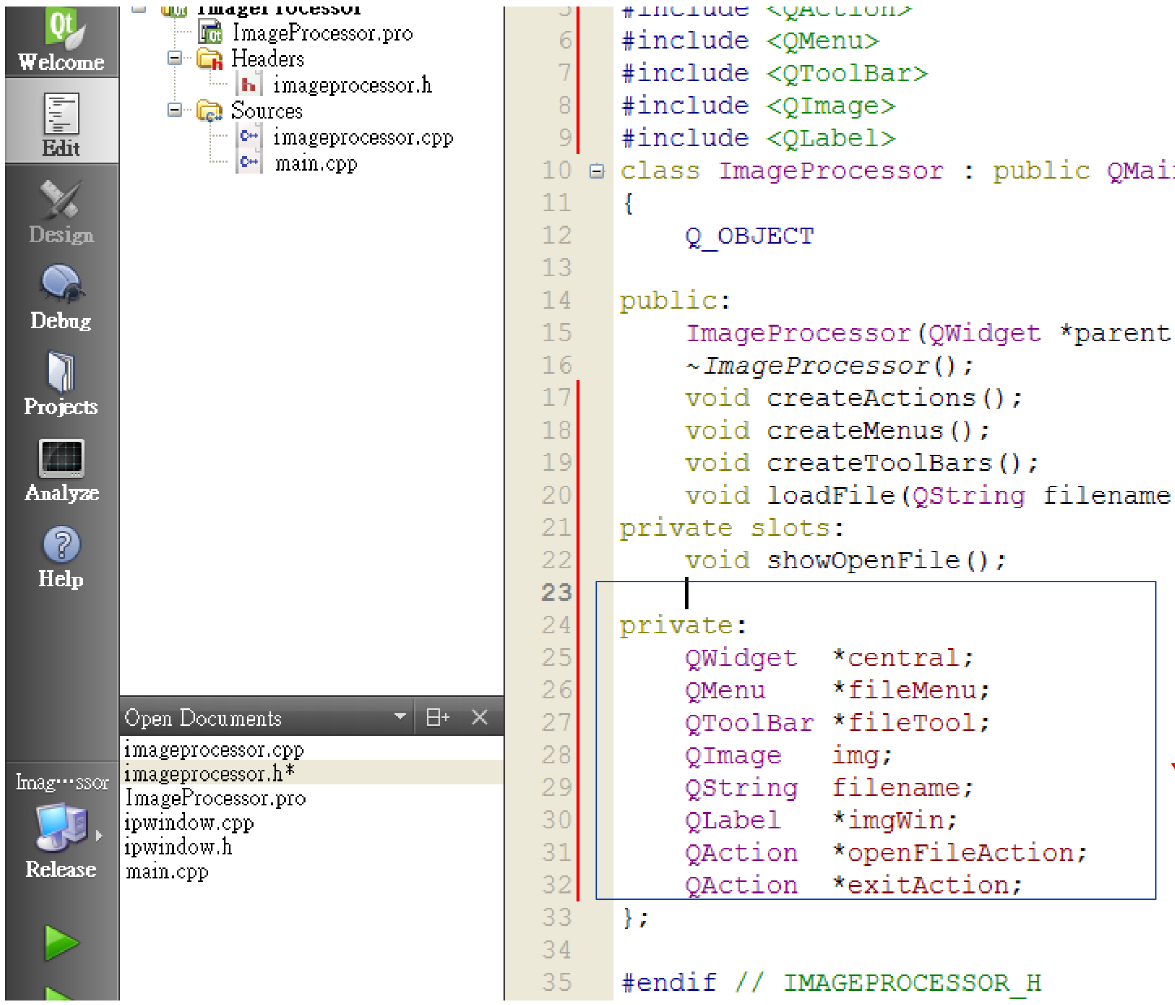
宣告Qt類別成員函數

宣告Qt類別槽函數

2021/12/16

Image Processor

- **Add ImageProcessor's members:**



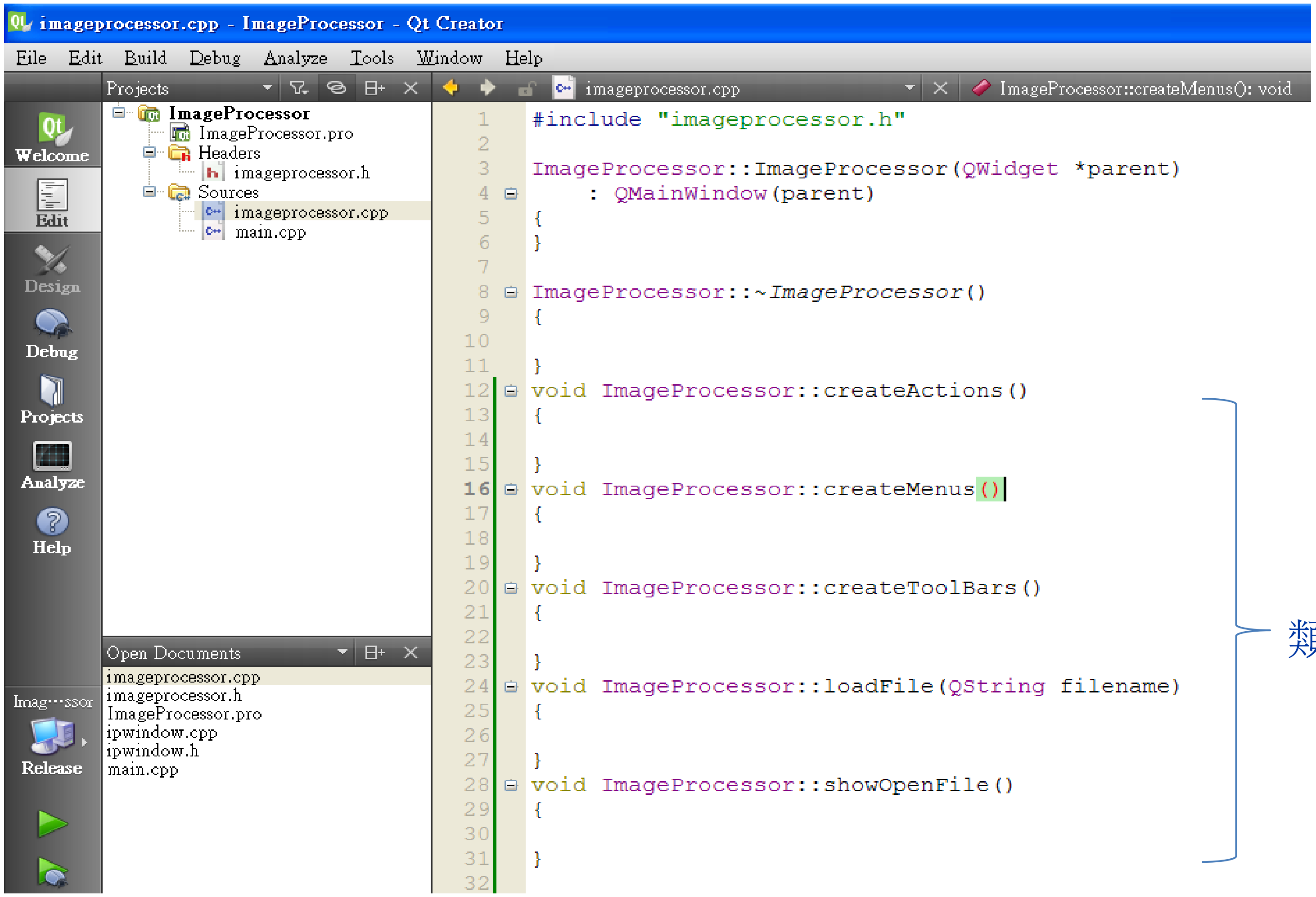
```
#include <QAction>
#include <QMenu>
#include <QToolBar>
#include <QImage>
#include <QLabel>
class ImageProcessor : public QMainWindow
{
    Q_OBJECT

public:
    ImageProcessor(QWidget *parent = 0);
    ~ImageProcessor();
    void createActions();
    void createMenus();
    void createToolBars();
    void loadFile(QString filename);
private slots:
    void showOpenFile();

private:
    QWidget  *central;
    QMenu    *fileMenu;
    QToolBar *fileTool;
    QImage   img;
    QString  filename;
    QLabel   *imgWin;
    QAction  *openFileAction;
    QAction  *exitAction;
};

#endif // IMAGEPROCESSOR_H
```

宣告Qt類別成員變數

2021/12/16

Image Processor

- **Implement ImageProcessor's member functions:**



```cpp
#include "imageprocessor.h"

ImageProcessor::ImageProcessor(QWidget *parent)
    : QMainWindow(parent)
{
}

ImageProcessor::~ImageProcessor()
{

}
void ImageProcessor::createActions()
{

}
void ImageProcessor::createMenus()
{

}
void ImageProcessor::createToolBars()
{

}
void ImageProcessor::loadFile(QString filename)
{

}
void ImageProcessor::showOpenFile()
{

}
```

類別成員函數

Image Processor

- **Build and execute the program:**

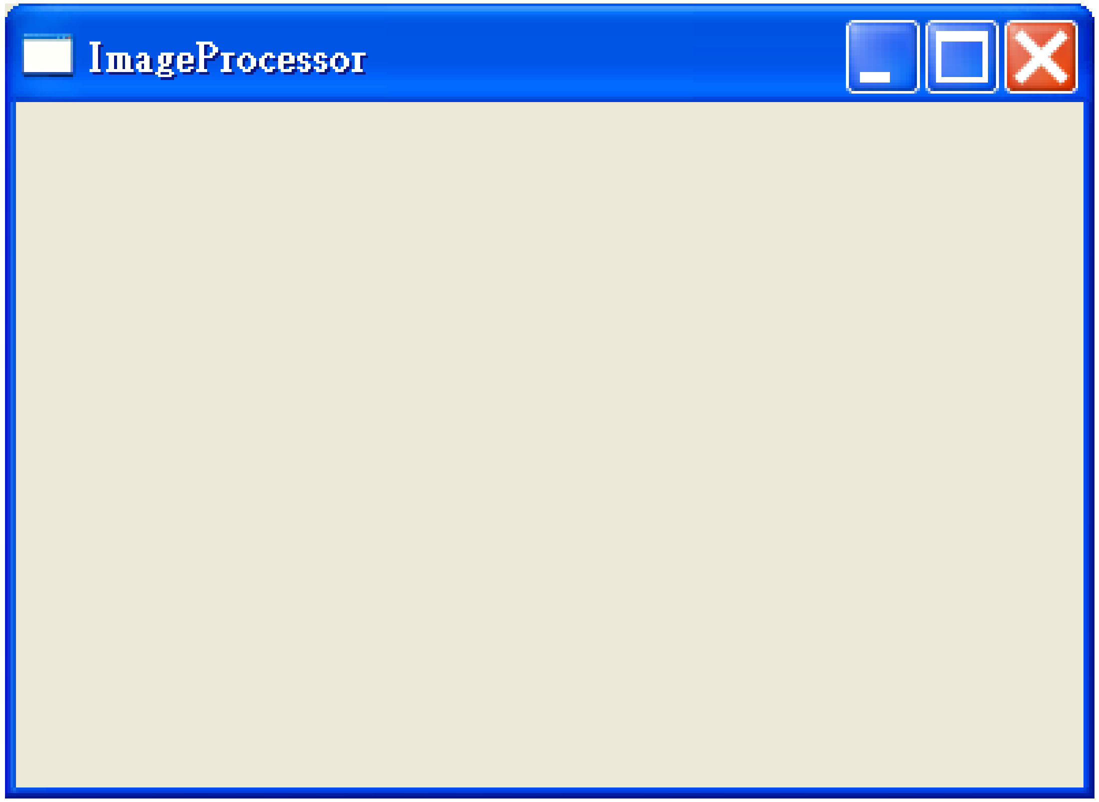
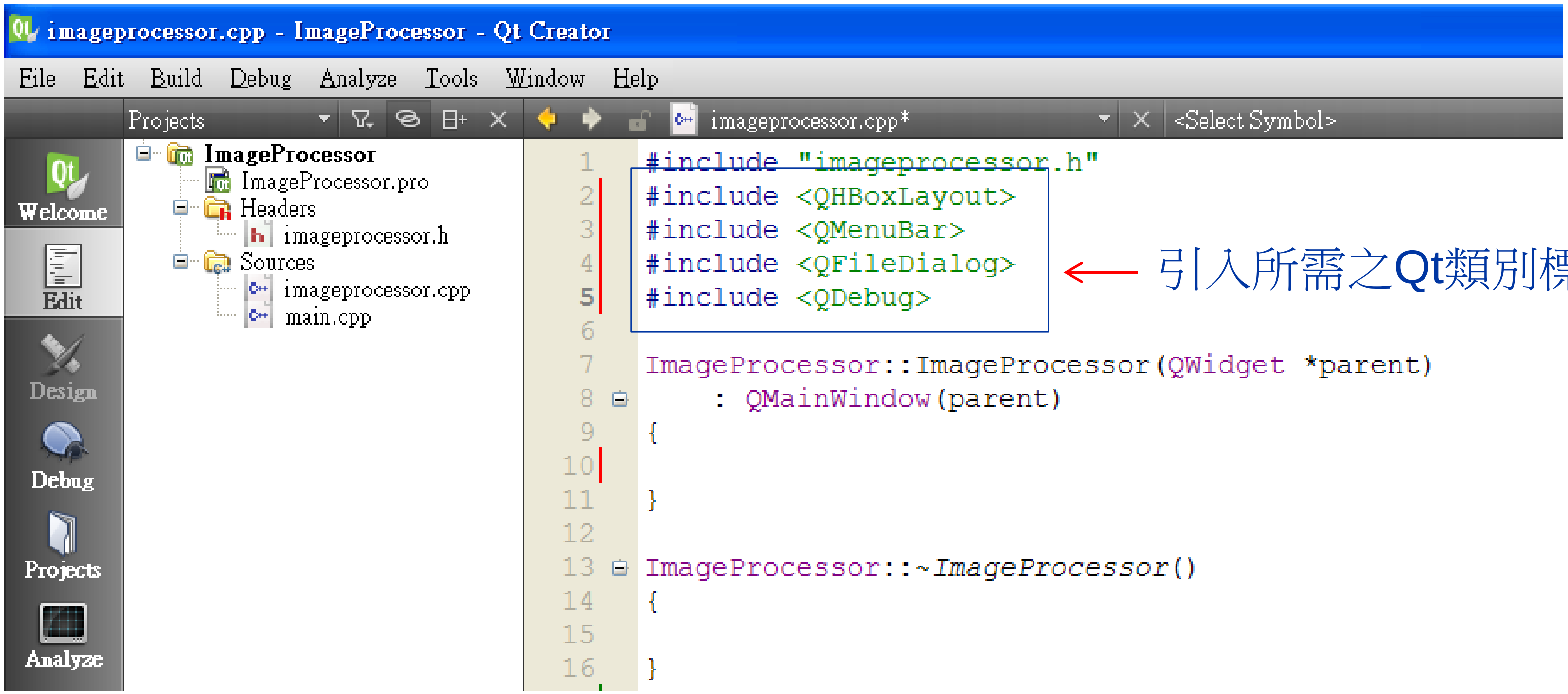


Image Processor

- **Implement ImageProcessor's constructor:**

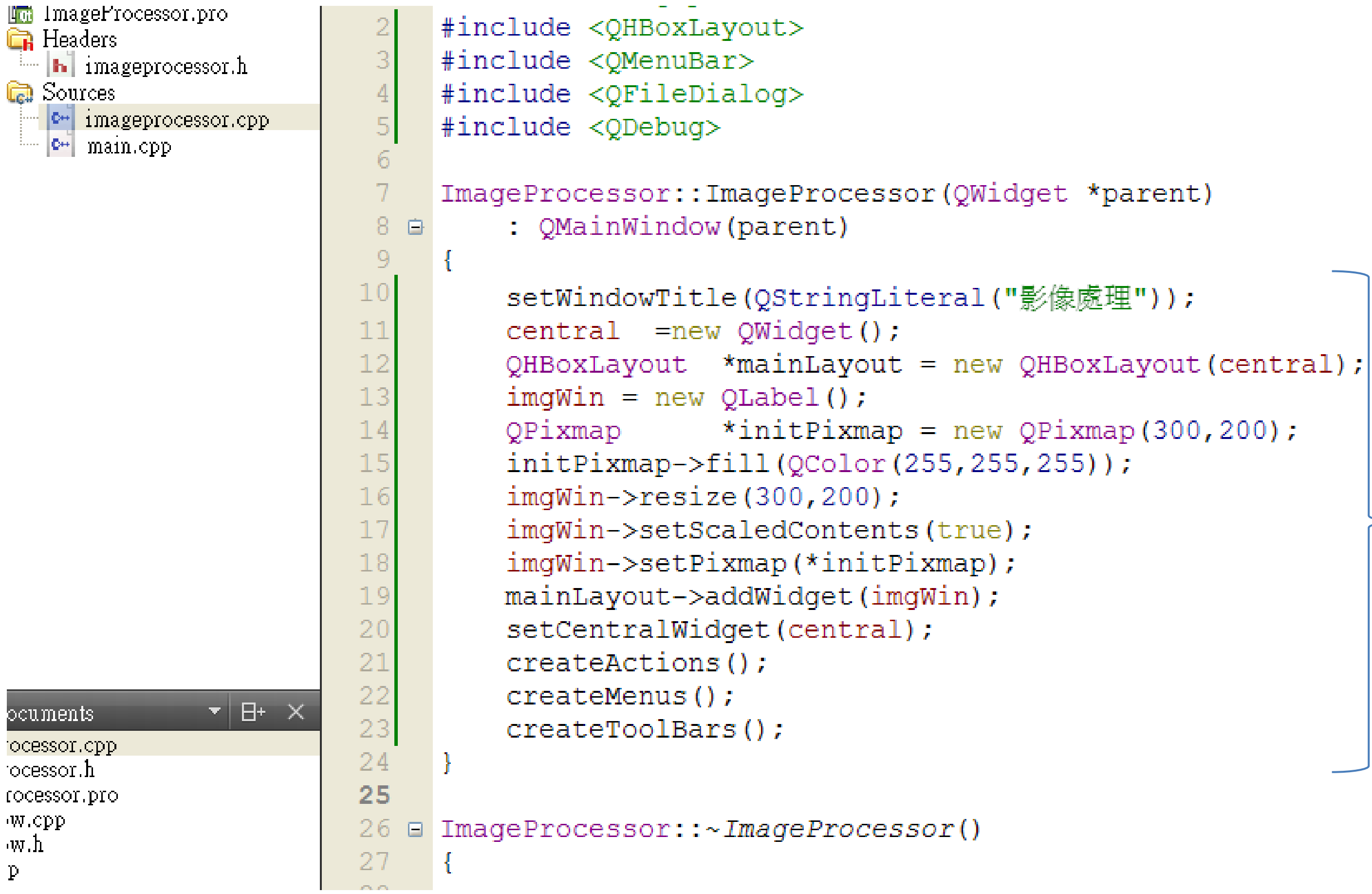


```
#include "imageprocessor.h"
#include <QHBoxLayout>
#include <QMenuBar>
#include <QFileDialog>
#include <QDebug>

ImageProcessor::ImageProcessor(QWidget *parent)
     : QMainWindow(parent)
{

}

ImageProcessor::~ImageProcessor()
{

}
```

← 引入所需之Qt類別標頭檔

Image Processor

- **Implement ImageProcessor's member functions:**



```
#include <QHBoxLayout>
#include <QMenuBar>
#include <QFileDialog>
#include <QDebug>

ImageProcessor::ImageProcessor(QWidget *parent)
    : QMainWindow(parent)
{
    setWindowTitle(QStringLiteral("影像處理"));
    central  =new QWidget();
    QHBoxLayout  *mainLayout = new QHBoxLayout(central);
    imgWin = new QLabel();
    QPixmap      *initPixmap = new QPixmap(300,200);
    initPixmap->fill(QColor(255,255,255));
    imgWin->resize(300,200);
    imgWin->setScaledContents(true);
    imgWin->setPixmap(*initPixmap);
    mainLayout->addWidget(imgWin);
    setCentralWidget(central);
    createActions();
    createMenus();
    createToolBars();
}

ImageProcessor::~ImageProcessor()
{
```

$\left.\right\}$ 建構函數程式碼

Image Processor

- **Build and execute the program:**

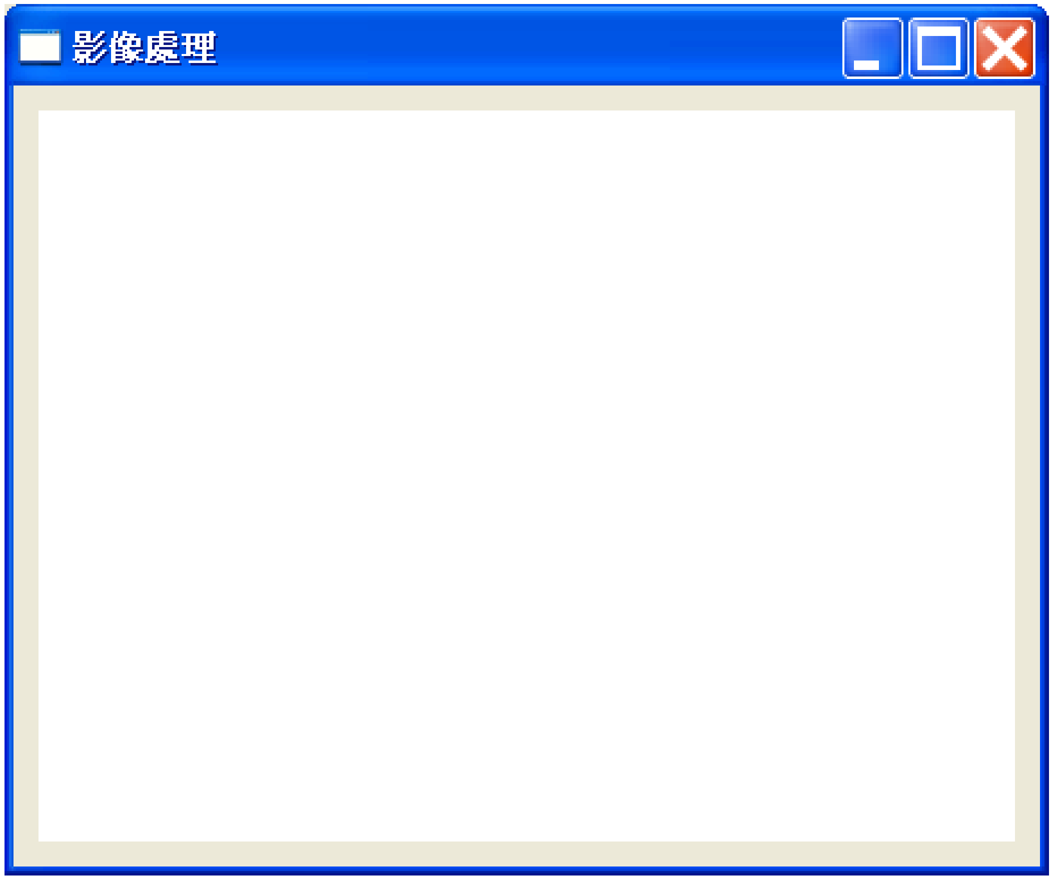
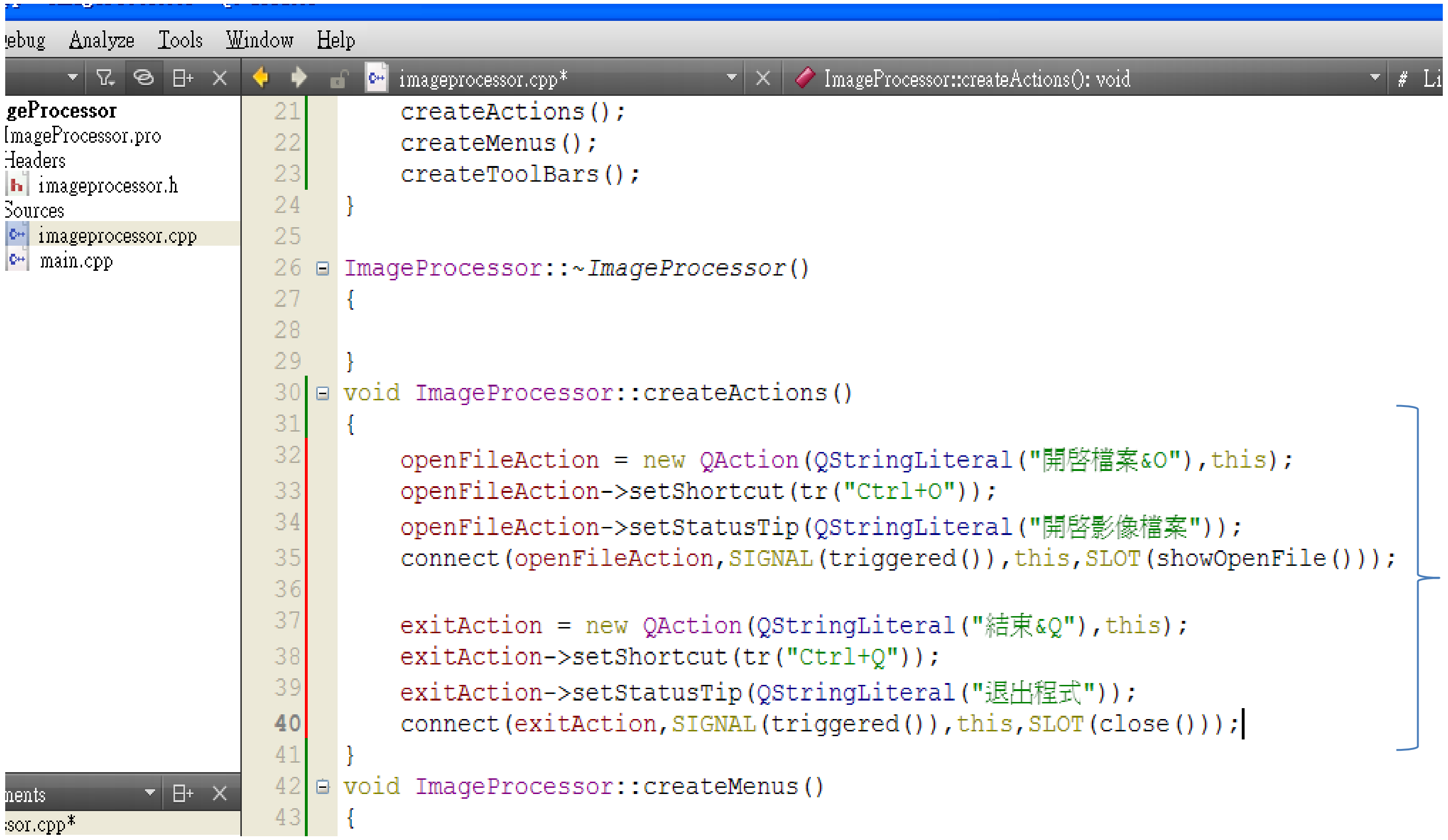


Image Processor

- **Implement ImageProcessor’s member functions:**

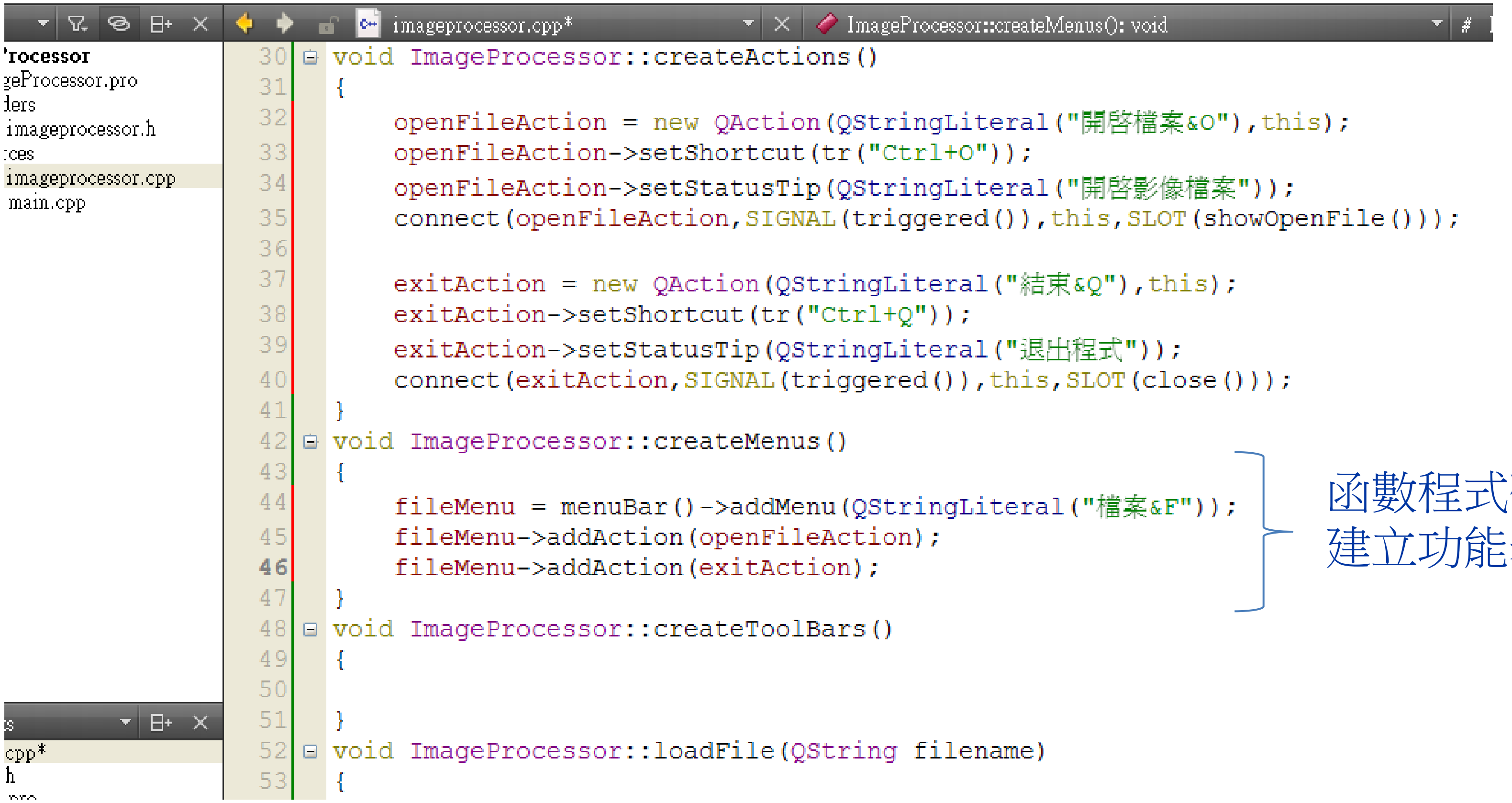


```
    createActions();
    createMenus();
    createToolBars();
}

ImageProcessor::~ImageProcessor()
{

}
void ImageProcessor::createActions()
{
    openFileAction = new QAction(QStringLiteral("開啟檔案&O"),this);
    openFileAction->setShortcut(tr("Ctrl+O"));
    openFileAction->setStatusTip(QStringLiteral("開啟影像檔案"));
    connect(openFileAction,SIGNAL(triggered()),this,SLOT(showOpenFile()));

    exitAction = new QAction(QStringLiteral("結束&Q"),this);
    exitAction->setShortcut(tr("Ctrl+Q"));
    exitAction->setStatusTip(QStringLiteral("退出程式"));
    connect(exitAction,SIGNAL(triggered()),this,SLOT(close()));
}
void ImageProcessor::createMenus()
{
```

函數程式碼建立動作物件

Image Processor

- **Implement ImageProcessor's member functions:**



imageprocessor.cpp* ImageProcessor::createMenus(): void

```
void ImageProcessor::createActions()
{
    openFileAction = new QAction(QStringLiteral("開啓檔案&O"),this);
    openFileAction->setShortcut(tr("Ctrl+O"));
    openFileAction->setStatusTip(QStringLiteral("開啓影像檔案"));
    connect(openFileAction,SIGNAL(triggered()),this,SLOT(showOpenFile()));

    exitAction = new QAction(QStringLiteral("結束&Q"),this);
    exitAction->setShortcut(tr("Ctrl+Q"));
    exitAction->setStatusTip(QStringLiteral("退出程式"));
    connect(exitAction,SIGNAL(triggered()),this,SLOT(close()));
}
void ImageProcessor::createMenus()
{
    fileMenu = menuBar()->addMenu(QStringLiteral("檔案&F"));
    fileMenu->addAction(openFileAction);
    fileMenu->addAction(exitAction);
}
void ImageProcessor::createToolBars()
{

}
void ImageProcessor::loadFile(QString filename)
{
```

函數程式碼
建立功能表

2021/12/16

Image Processor

- **Build and execute the program:**

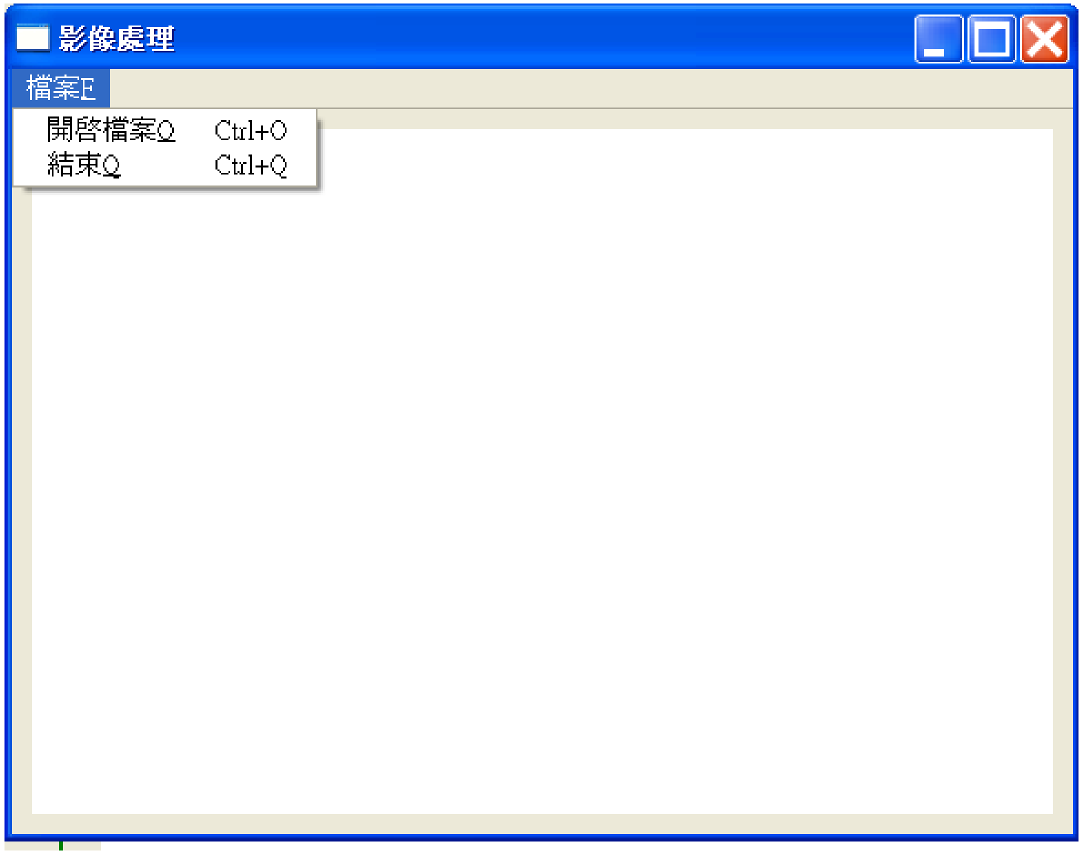
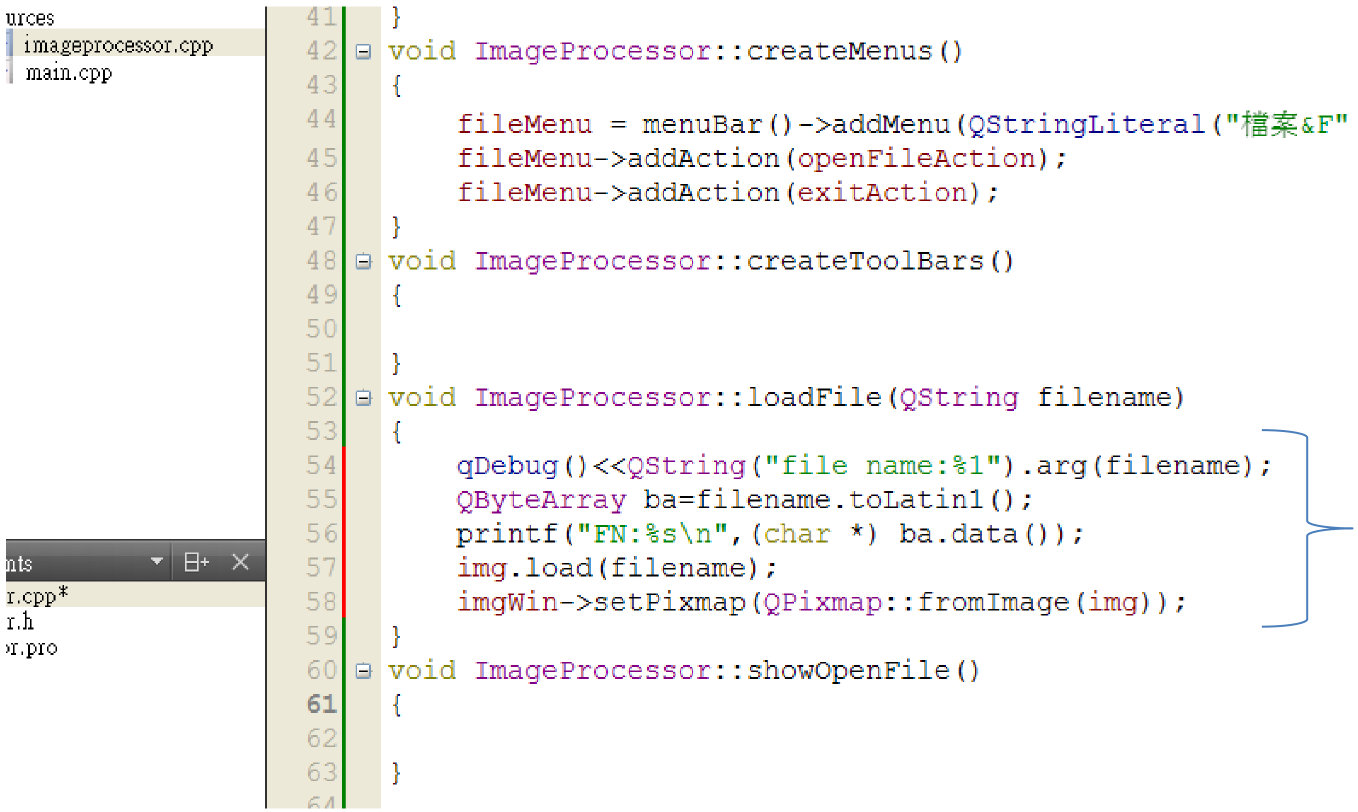


Image Processor

- **Implement ImageProcessor's member functions:**



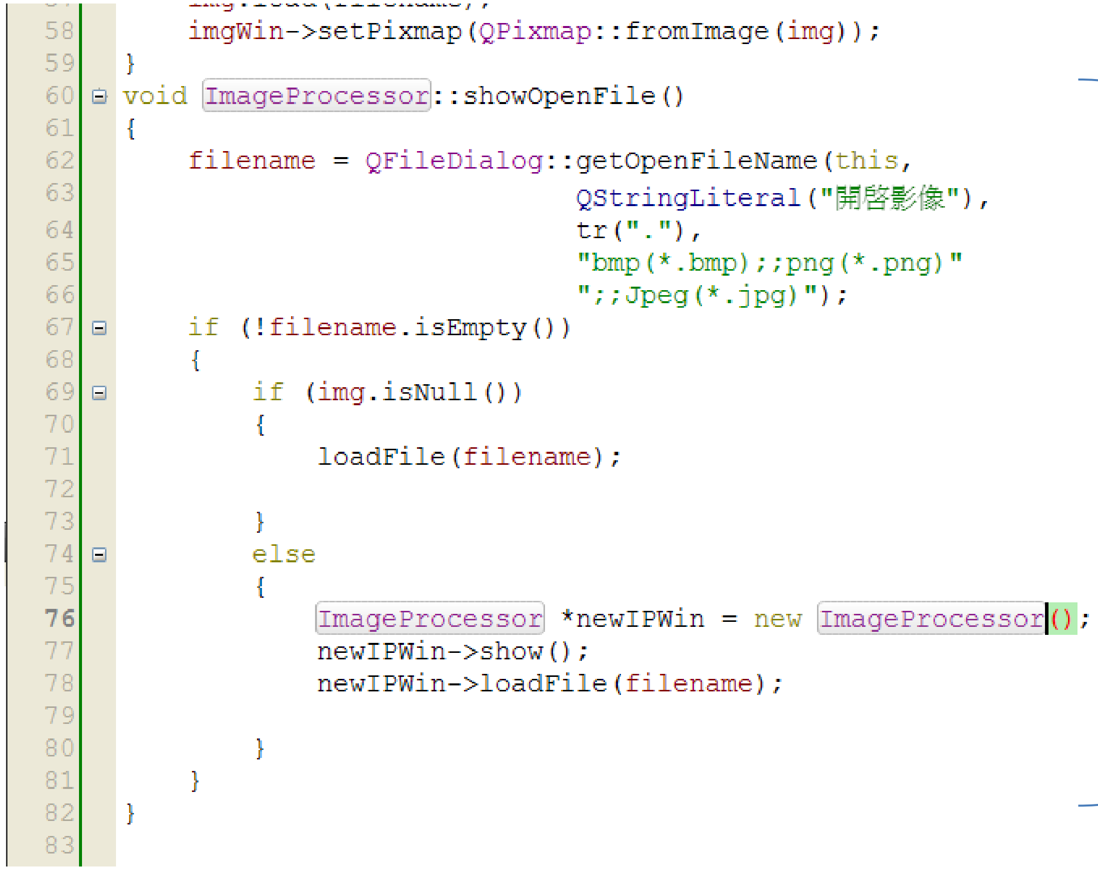
urces
imageprocessor.cpp
main.cpp

```
}
void ImageProcessor::createMenus()
{
    fileMenu = menuBar()->addMenu(QStringLiteral("檔案&F"
    fileMenu->addAction(openFileAction);
    fileMenu->addAction(exitAction);
}
void ImageProcessor::createToolBars()
{

}
void ImageProcessor::loadFile(QString filename)
{
    qDebug()<<QString("file name:%1").arg(filename);
    QByteArray ba=filename.toLatin1();
    printf("FN:%s\n",(char *) ba.data());
    img.load(filename);
    imgWin->setPixmap(QPixmap::fromImage(img));
}
void ImageProcessor::showOpenFile()
{

}
```

函數程式碼載入影像檔並顯示於視窗中心元件上。

Image Processor

- **Implement ImageProcessor's member functions:**



```
        imgWin->setPixmap(QPixmap::fromImage(img));
    }
    void ImageProcessor::showOpenFile()
    {
        filename = QFileDialog::getOpenFileName(this,
                                QStringLiteral("開啟影像"),
                                tr("."),
                                "bmp(*.bmp);;png(*.png)"
                                ";;Jpeg(*.jpg)");
        if (!filename.isEmpty())
        {
            if (img.isNull())
            {
                loadFile(filename);

            }
            else
            {
                ImageProcessor *newIPWin = new ImageProcessor();
                newIPWin->show();
                newIPWin->loadFile(filename);

            }
        }
    }
```

函數程式碼，啟動檔案對話盒，供使用者挑選檔案路徑與檔名後，呼叫載入影像檔並顯示於視窗中心元件上。

Image Processor

- **Build and execute the program:**

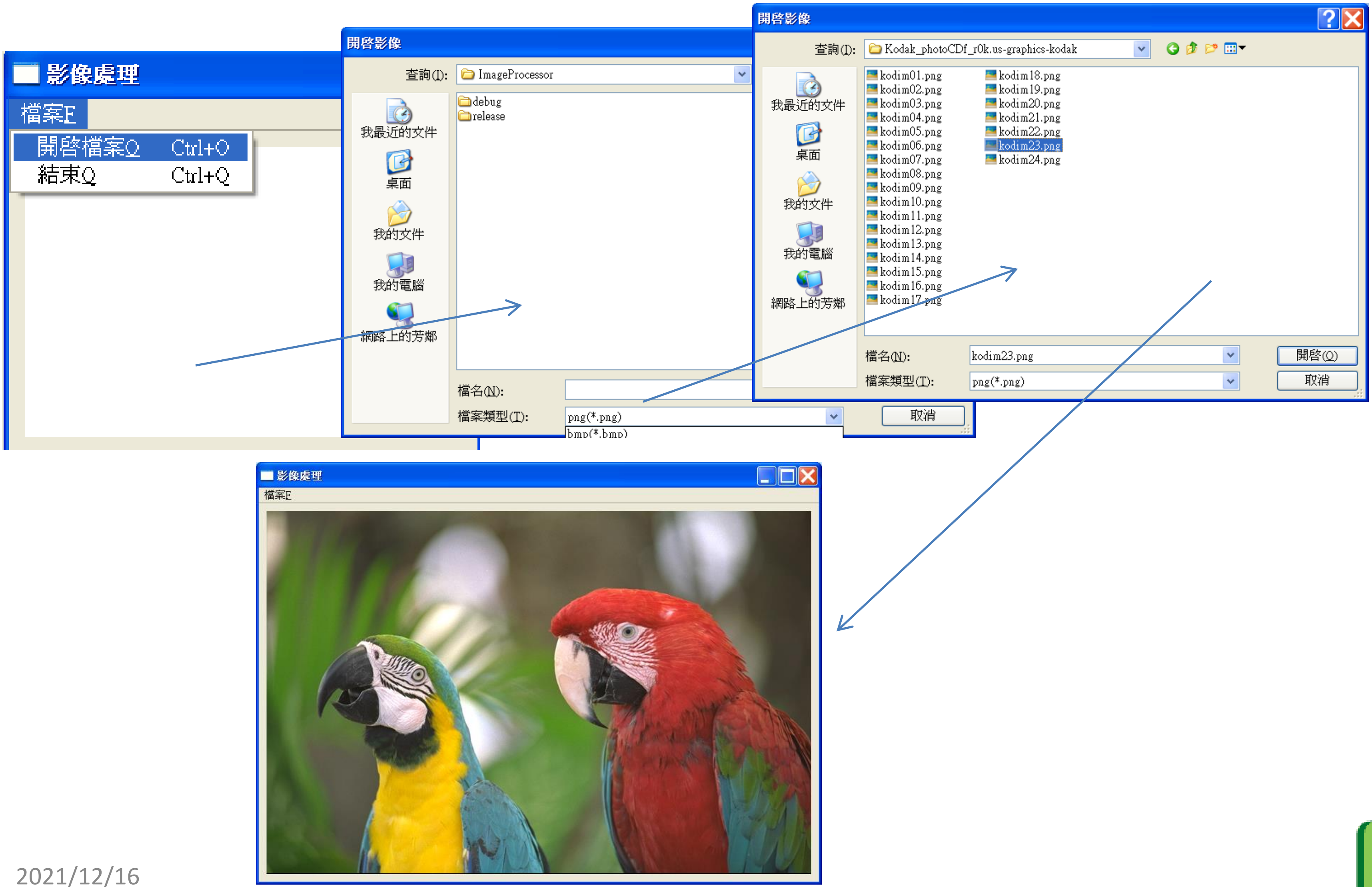
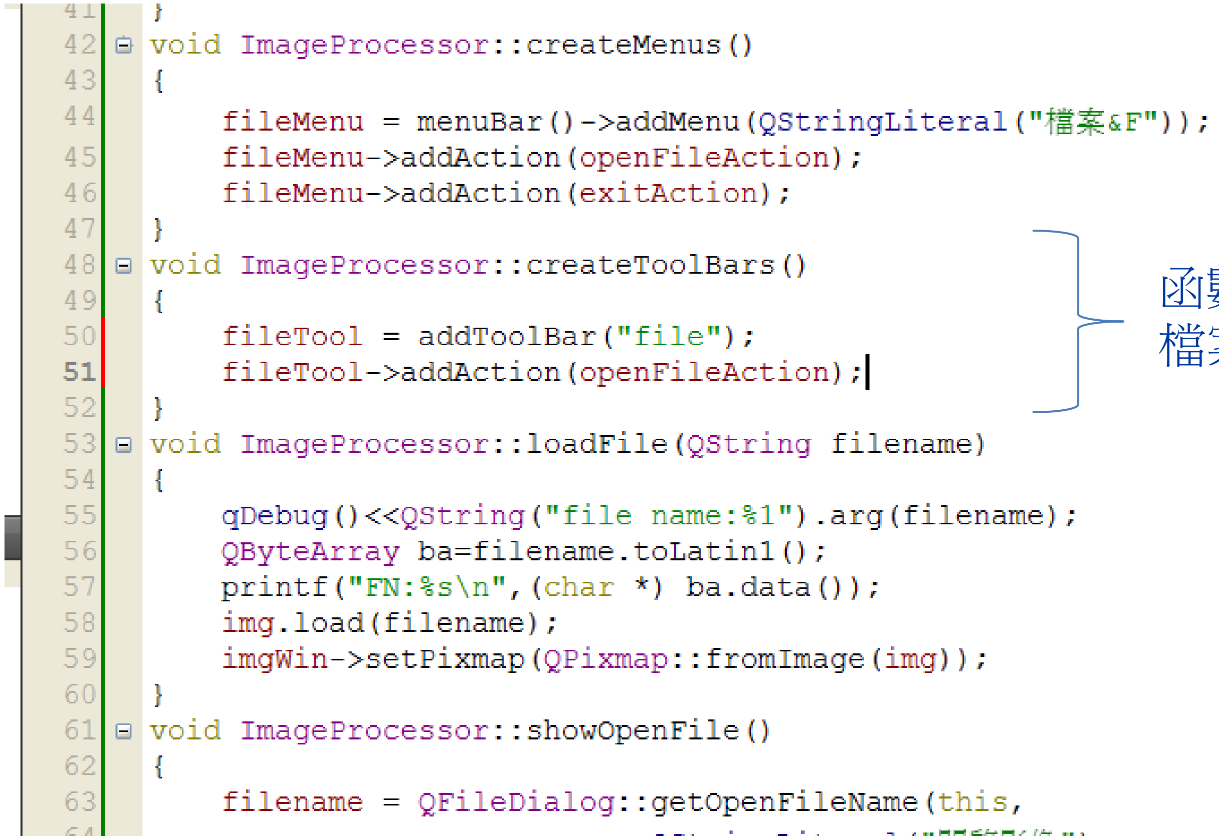


Image Processor

- **Implement ImageProcessor's member functions:**



```
}
void ImageProcessor::createMenus()
{
    fileMenu = menuBar()->addMenu(QStringLiteral("檔案&F"));
    fileMenu->addAction(openFileAction);
    fileMenu->addAction(exitAction);
}
void ImageProcessor::createToolBars()
{
    fileTool = addToolBar("file");
    fileTool->addAction(openFileAction);
}
void ImageProcessor::loadFile(QString filename)
{
    qDebug()<<QString("file name:%1").arg(filename);
    QByteArray ba=filename.toLatin1();
    printf("FN:%s\n",(char *) ba.data());
    img.load(filename);
    imgWin->setPixmap(QPixmap::fromImage(img));
}
void ImageProcessor::showOpenFile()
{
    filename = QFileDialog::getOpenFileName(this,
```

函數程式碼，建立
檔案工具列。

2021/12/16

Image Processor

- **Build and execute the program:**

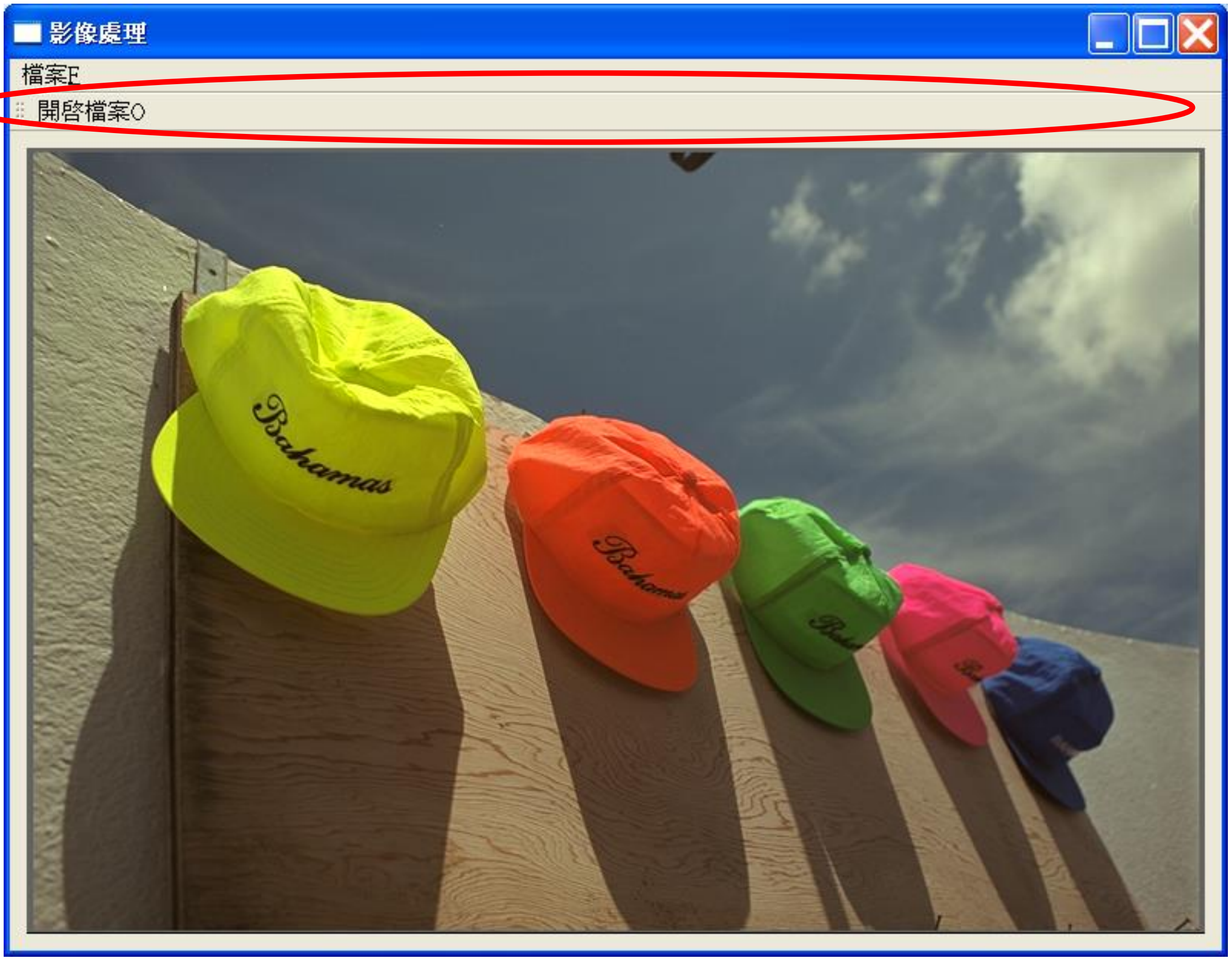


Image Processor

- **Exercise: Add more tools into the program; for example,**

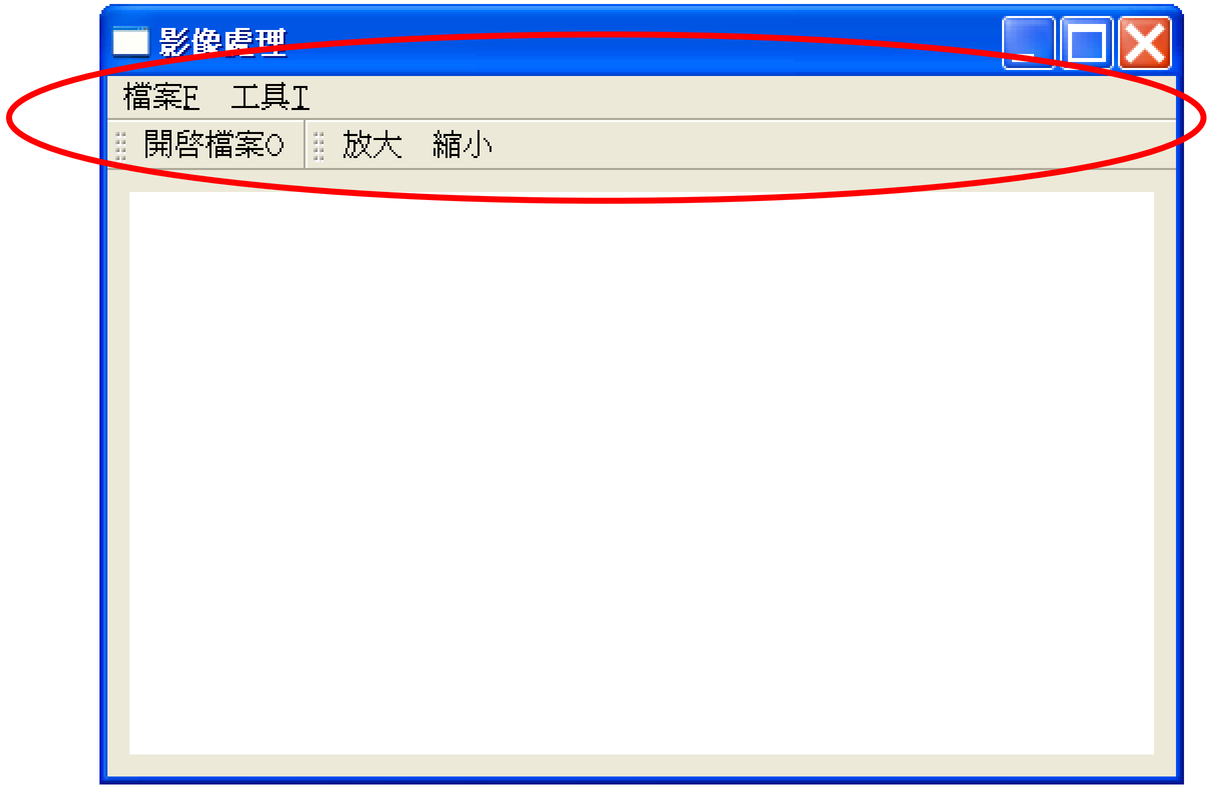
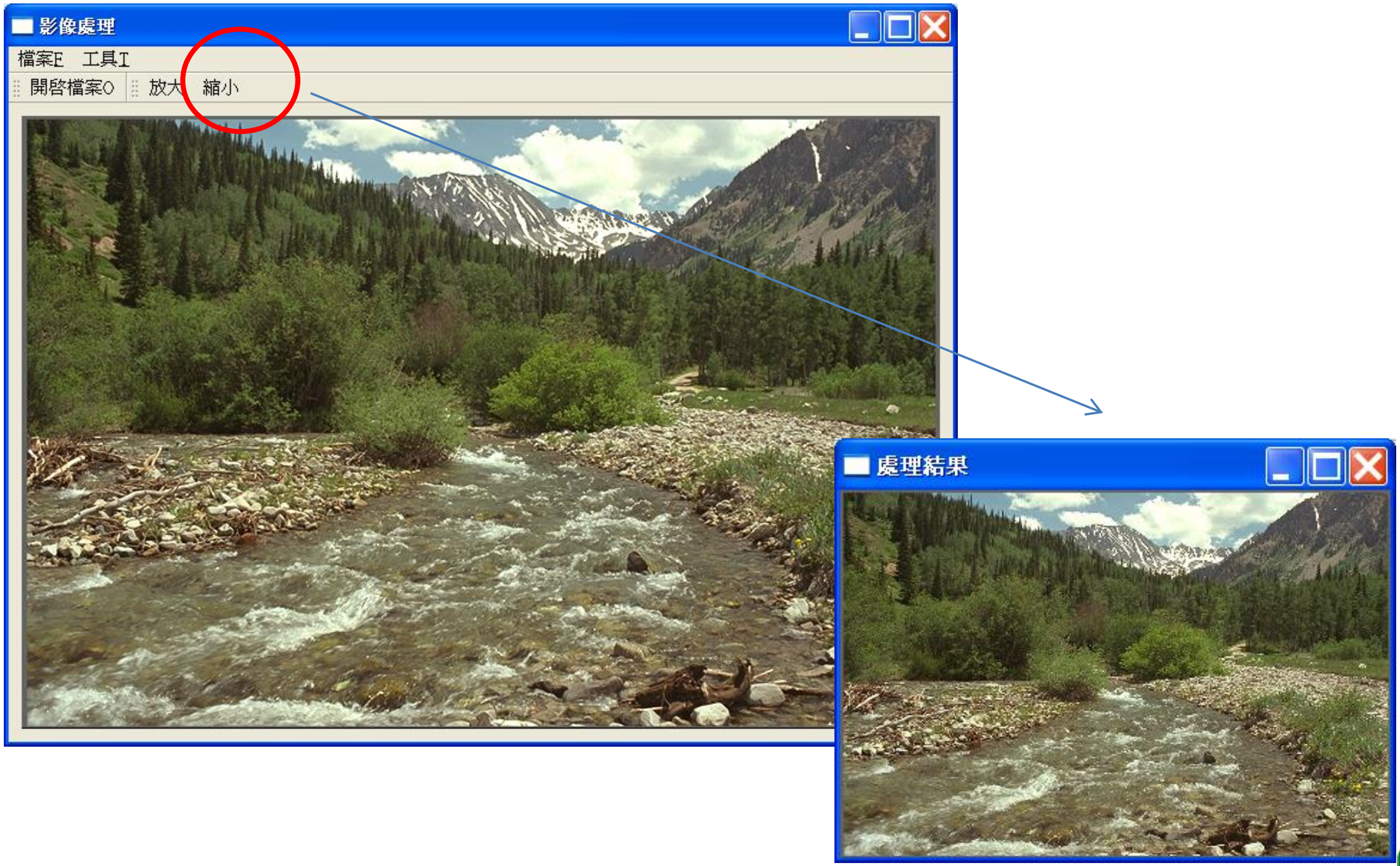


Image Processor

- **Exercise: Add more tools into the program; for example,**


影像處理
檔案F
工具T
開啓檔案O
放大
縮小
處理結果