

QT Eğitimi - 1 : Merhaba Dünya

Önder Arslan

<onder@xcoders.net>

Sürüm 1.0

Özet

Bu belge qt öğrenimi için bir dizi şeklinde hazırlanmıştır. C++ bilmek ön koşuldur.

Giriş

Böyle bir yazı yazma fikri qt asistanı (qtassistant) 'nda bulunan eğitimler (tutorial) 'e çalışırken ortaya çıktı. Oldukça iyi hazırlanmış bu eğitimlere benzer Türkçe belgeler dizisi hazırlamanın yararlı olacağını düşündüm. Öncelikle bu eğitimlerdeki örnekleri kullanacaktım ama Kaya Oğuz 'un trolltech ile yazışmasından sonra bu belgeleri kullanmaya izin verilmediğini öğrenmiş olduk. Bu sebeple belgeler kendine özgü olacak. Bakarsınız benim örneklerim ve anlatımım daha güzel olur :)

QT her ne kadar python ve ruby ile kullanılabilse de ben örneklerimde C++ ı kullanacağım. Bunun çok önemli bir nedeni var. Python ve ruby bilmiyorum :) . Sanırım bu yeterli bir sebep , bu nedenle ileride de benden başka bir dilde örnek göremeyeceksiniz.

Ayrıca şunu belirtmekte yarar var bu yazı dizisi ben qt 'de ilerledikçe ilerleyecek bir yazı dizisi. Bu nedenle kodlardaki açıklamalarım sıcağı sıcağına ve bir aceminin bakış açısından olacak. Kodları yazarken hata yaptığım yerleri özellikle belirteceğim. Böylece sizde aynı hataları yapmak durumunda kalmayacaksınız.

Merhaba Dünya

Qt Eğitimi -1 'in ilk programı doğal olarak "Merhaba Dünya" olacak. Klasik olarak her yeni dil öğrenildiğinde yazılan bu programın [Brian W. Kernighan](#) and [Dennis M. Ritchie](#) (K&R) 'nin yazdığı The C Programming Language 'kitabından itibaren kullanıldığını biliyorum. Bizde adeti bozmayacağız ama biraz farklı yazacağız :

```
/*--- Pencere.h ---*/

#ifdef PENCERE_H
#define PENCERE_H

#include <QPushButton>
#include <QWidget>
#include <QMessageBox>

class Pencere : public QWidget
{
    Q_OBJECT
public:
    QPushButton *btn;

    Pencere();
```

```

~Pencere();

public slots:
    void fncTiklama()
    {
        QMessageBox::information(this, "Mesaj", "Merhaba
Dunya");
    }
};
#endif

```

Pencere.h:

İlk olarak başlık (header) dosyamıza include ön işlemci komutuyla kullanacağımız Q class 'ları dahil ediyoruz. Yazımı genel olarak bilinen kontrollerin başına Q harfi getirilmiş hali diyebiliriz. Burada QPushButton, QWidget, QMessageBox kullanacağımız için yalnızca üçünü dahil ettik. Pencere isimli sınıf (class) 'ımızı **QWidget**'ten public olarak türetiyoruz (inheritance). QWidget bütün arayüz nesnelere türetilmiş temel sınıftır ve sistemden gelen fare, klavye ve diğer olayları taşır. Sınıf bloğunda ilk olarak yazdığımız **Q_OBJECT** sınıf tanımımız içerisinde kendi **SIGNAL** ve **SLOT** 'larımızı tanımlarsanız kullanmanız gereken bir macrodur. Public erişim bölgesinde QPushButton için bir gösterici, sınıfımız için bir yapıcı(constructor) ve bir yıkıcı (destructor) tanımlıyoruz. public slots bölümünde ise arayüzümüzde bir olay(event) olması durumunda çalışacak fonksiyonu yazıyoruz (Q_OBJECT i bu nedenle eklemiştik.). Burada biz mouse pushbutton 'a tıkladığı zaman çalışacak fonksiyonumuz fncTiklama fonksiyonudur. Fonksiyon "Merhaba Dünya" yazan bir QMessageBox oluşturuyor.

```

/*--- Pencere.cpp ---*/

#include "Pencere.h"

Pencere::Pencere()
{
    setFixedSize(150,100);
    btn = new QPushButton("tikla", this);
    btn->setGeometry(50,30,50,30);

    connect(btn,SIGNAL(clicked()),this,SLOT(fncTiklama()));
}

Pencere::~Pencere()
{
    delete btn;
}

```

Pencere.cpp

Bu dosyada yapıcı ve yıkıcı metodların tanımlamalarını yapıyoruz. İlk olarak Pencere'nin **setFixedSize** 'a sırasıyla genişlik ve yükseklik değerlerini geçerek penceremizin boyutlarını ayarlıyoruz. Daha sonra üzerinde tikla yazan bir button oluşturup **setGeometry** fonksiyonu ile sırasıyla konum(x, y) ve boyut(w, h) bilgilerini giriyoruz. **connect** ile button 'un clicked() olayı için sınıfımızın fncTiklama()

fonksiyonunun çalışmasını sağlıyoruz.

```
/*--- main.cpp ---*/  
  
#include "Pencere.h"  
  
#include <QApplication>  
  
int main(int argc, char *argv[])  
{  
    QApplication app  
(argc,argv);  
    Pencere pencere;  
    pencere.show();  
    return app.exec();  
}
```

main.cpp

İlk olarak **QApplication** sınıfını dahil ediyoruz. Bu sınıfın bir çok görevi var ve şimdilik programımızı başlangıç durumuna getirme (initialize) ve sonlandırma (finalization) işlemlerini gerçekleştirdiğini bilmemiz yeterli ;) Hemen bir QApplication nesnesi tanımlıyoruz ve birde tanımladığımız Pencere nesnesinden ;) sonra **show()** fonksiyonu ile penceremizi görünür yapıyoruz. ve return ile **exec()** fonksiyonun geri dönüş değerini döndürüyoruz. Hepsi bu ;)

Derleme

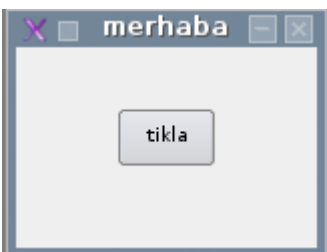
ve

Çalıştırma

Yazdığımız .h ve .cpp uzantılı dosyaları merhaba adında bir klasör içine kaydedin ve konsoldan sırasıyla aşağıdaki komutları verin:

```
qmake-qt4 -project  
qmake-qt4  
make  
  
./merhaba
```

Bu adımlardan sonra aşağıdaki görüntü ile karşılaşmış olmanız lazım ;)



Son Söz

Burada birkaç ayrıntıyı belirtmem lazım çünkü beni oldukça uğraştırdı. Birincisi eğer sınıf bildirimlerini (declaration) yukarıdaki gibi başlık dosyasında yapmayacaksanız uygun moc dosyasını .cpp dosyasında bir yere yazmanız lazım. Örneğin yukarıdaki kodların hepsini main.cpp de çalıştırmak istediğinizi varsayarsak sınıf bildiriminin altında **#include "main.moc"** yazmassanız make komutunda hata alabilirsiniz ;) Ben aldım ordan biliyorum ;) İkinci bir nokta ise qt cross platform ben bu kodları windows

'ta çalıştıracağım gibi bir düşünce de iseniz -ki böyle düşünmemekte fayda var- komut satırından qmake -project, qmake, make üçlüsünü çalıştırın(-qt4 eklemeyin). Birde normal dos ekranını açmak yerine "Başlat -> Programlar ->Qt by TrollTech v.4.1.4 (opensource) -> Qt 4.1.4 Command Prompt" ile açılan dos ekranında derlemenizi yapın, aksi halde make komutunu bulamıyor (En azından ben denedim bende bulamadı path ayarlarını yapmama rağmen).

Telif Hakkı ve Lisans

Bu belgenin, *QT Eğitimi - 1 : Merhaba Dünya* , 1.0 sürümünün **telif hakkı © 2006 Önder ARSLAN**'a aittir. Bu belgeyi, Free Software Foundation tarafından yayınlanmış bulunan GNU Özgür Belgeleme Lisansının 1.1 ya da daha sonraki sürümünün koşullarına bağlı kalarak kopyalayabilir, dağıtabilir ve/veya değiştirebilirsiniz. Bu Lisansın bir kopyasını <http://www.gnu.org/copyleft/fdl.html> adresinde bulabilirsiniz.

Linux, Linus Torvalds adına kayıtlı bir ticarî isimdir.

Qt, TrollTech adına kayıtlı bir ticari isimdir.

Feragatname

Bu belgedeki bilgilerin kullanımından doğacak sorumluluklar, ve olası zararlardan belge yazarı sorumlu tutulamaz. Bu belgedeki bilgileri uygulama sorumluluğu uygulayana aittir.

Tüm telif hakları aksi özellikle belirtilmediği sürece sahibine aittir. Belge içinde geçen herhangi bir terim bir ticarî isim ya da kuruma itibar kazandırma olarak algılanmamalıdır. Bir ürün ya da markanın kullanılmış olması ona onay verildiği anlamında görülmemelidir.