

Eseményvezérelt alkalmazások fejlesztése I

1. előadás

Grafikus felület és eseményvezérlés, a Qt keretrendszer

Giachetta Roberto

<http://people.inf.elte.hu/groberto>

A Qt keretrendszer

Bemutatása

- A *Qt* egy alkalmazás-fejlesztési keretrendszer, amely számos platformot támogatja az asztali, mobil és beágyazott alkalmazások fejlesztését
- elérhető a qt.io oldalról
- támogatja a grafikus felületet, adatbázis-kezelést, multimédiát, 3D grafikát, hálózati és webes kommunikációt
- rendelkezik nyílt forráskódú (GPL) és kereskedelmi verzióval is
- fejlesztésre elsősorban a C++-t támogatja, de más nyelvekre is elérhető, valamint rendelkezik saját leíró nyelvvel (Qt Quick)



A Qt keretrendszer

A „Hello World” program

```
#include <QApplication>
#include <QLabel>
// megfelelő Qt osztályok használata

int main(int argc, char *argv[] ) {
    QApplication app(argc, argv);
    // alkalmazás osztály példányosítása
    QLabel myLabel("Hello, world!");
    // címke a felirattal
    myLabel.show(); // címke megjelenítése

    return app.exec(); // alkalmazás futtatása
}
```

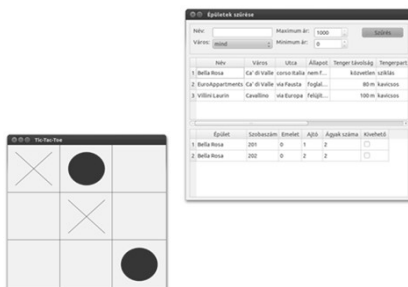
A Qt keretrendszer

A grafikus felületű alkalmazás

- Grafikus felületű alkalmazásnak* nevezzük azt a programot, amely 2D-s interaktív felhasználó felületen (*GUI*, *Graphical User Interface*) keresztül kommunikál a felhasználóval
- gazdagabb interakció a konzol felületnél, számos módon beavatkozhatunk a programfutásba
- a működés jórészt várakozás a felhasználói interakcióra
- a felület egy, vagy több ablakból (*form/window*) áll, amelyek vezérlőket (*control/widget*) tartalmaznak (pl.: nyomógombok, listák, menük, ...)
- mindig van egy aktív ablak, és egy aktív vezérlő (ezen van a *fókusz*)

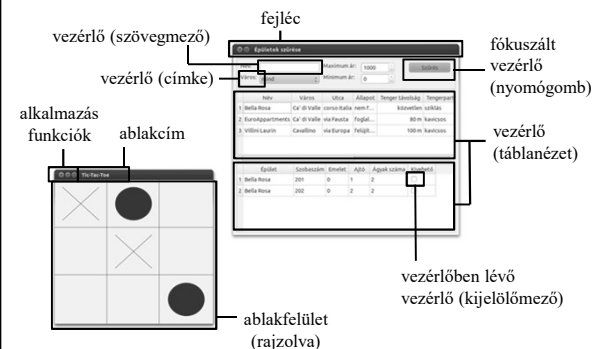
A Qt keretrendszer

A grafikus felületű alkalmazás



A Qt keretrendszer

A grafikus felületű alkalmazás



A Qt keretrendszer

A grafikus felület felépülése

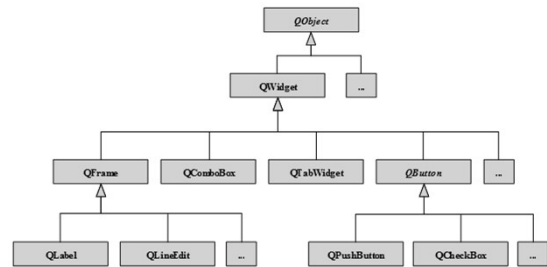
- A grafikus felület *objektumorientáltan* épül fel
 - a vezérlőket osztályokként fogalmazzuk meg, megadjuk viselkedését (*metódusokkal*, pl. kattintás, megjelenés), illetve tulajdonságait (*mezőkkel*, pl. pozíció, méret, betűtípus)
 - a vezérlők sok hasonló tulajdonsággal bírnak, így könnyen öröklődési hierarchiába szervezhetőek
 - az öröklődési lánc legelején áll az általános vezérlő, új vezérlők származtatással definiálhatóak
 - a vezérlőket felhasználhatjuk más vezérlőkben, vagy használhatjuk önállóan, azaz *ablakként*

ELTE IK, Eseményvezérelt alkalmazások fejlesztése I

1:7

A Qt keretrendszer

A grafikus felület felépülése

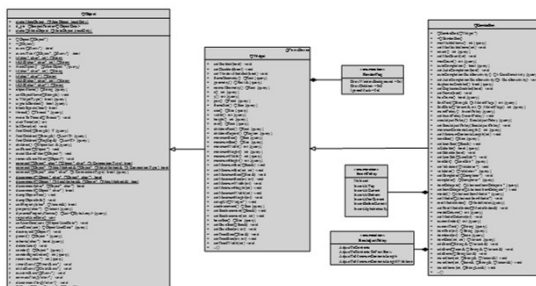


ELTE IK, Eseményvezérelt alkalmazások fejlesztése I

1:8

A Qt keretrendszer

A grafikus felület felépülése



ELTE IK, Eseményvezérelt alkalmazások fejlesztése I

1:9

A Qt keretrendszer

Fejlesztés és fordítás

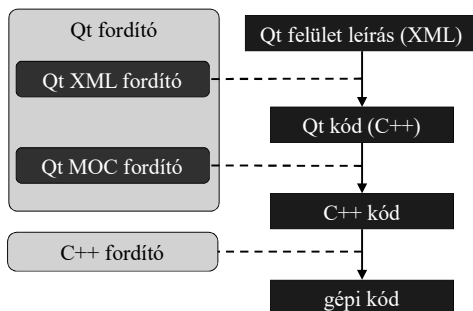
- A fejlesztés C++/Qt nyelven történik
 - elérhető a teljes C++ utasításkészlet, nyelvi könyvtár
 - a C++ nyelven felül további makrókat, kiegészítéseket tartalmaz, amelyeket a *Meta Object Compiler (MOC)* fordít le ISO C++ kódra
- Az alapértelmezett fejlesztőeszköz a Qt Creator, de más környezetekben is megjelent a Qt fejlesztés lehetősége (pl. *Code::Blocks, Visual Studio*)
- Külön tervezőprogram (*Qt Designer*) adott a grafikus felület létrehozására, amely XML nyelven írja la felület felépítését, ez automatikusan C++/Qt kódra fordul

ELTE IK, Eseményvezérelt alkalmazások fejlesztése I

1:10

A Qt keretrendszer

Fejlesztés és fordítás



ELTE IK, Eseményvezérelt alkalmazások fejlesztése I

1:11

A Qt keretrendszer

Fejlesztés és fordítás

- A fordítás projektszinten történik, a szükséges információk *projektfájlokban (.pro)* tárolódnak, amely tartalmazza
 - a felhasznált modulokat, csatlakozásokat
 - forrásfájlok, erőforrások (pl. kép, szöveg,) listáját
 - eredmény paramétereket
- A fordítás közvetlenül is elvégezhető a fordítóval:


```

qmake -project
# projektfájl automatikus létrehozása
qmake # fordítófájl (makefile) előállítás
make # projekt fájlnak megfelelő fordítás és
# szerkesztés végrehajtása
            
```

ELTE IK, Eseményvezérelt alkalmazások fejlesztése I

1:12

A Qt keretrendszer	
Modulok	
<ul style="list-style-type: none"> A keretrendszer felépítése modularizált, a legfontosabb modulok: <ul style="list-style-type: none"> központi modul (<code>QtCore</code>) grafikus felület (<code>QtGui</code>), grafikus vezérlők (<code>QtWidgets</code>) adatbázis-kezelés (<code>QtSql</code>) A projektben használandó modulokat a projektfájlban kell megadnunk, pl.: <code>QT += core gui widgets</code> Egy modul tartalmát osztályonként, illetve egyszerre is betölthetjük az aktuális fájlba (pl. <code>#include <QtGui></code>) 	
ELTE IK, Eseményvezérelt alkalmazások fejlesztése I	1:13

A Qt keretrendszer	
Osztályhierarchia	
<ul style="list-style-type: none"> A nyelvi könyvtár osztályainak jelentős része teljes származtatási hierarchiában helyezkedik el <ul style="list-style-type: none"> minden egy őszosztály (<code>QObject</code>) leszármazottja az őszosztály biztosítja az eseménykezelést (<code>connect</code>), a tulajdonságkezelést, az időzítést (<code>timer</code>), stb. Számos segédosztállyal rendelkezik, pl.: <ul style="list-style-type: none"> adatszerkezetek (<code>QVector</code>, <code>QStack</code>, <code>QLinkedList</code>, ...) fájl és fájlrendszer kezelés (<code>QFile</code>, <code>QTextStream</code>, <code>QDir</code>, ...) párhuzamosság és aszinkron végrehajtás (<code>QThread</code>, <code>QSemaphore</code>, <code>QFuture</code>, ...) 	
ELTE IK, Eseményvezérelt alkalmazások fejlesztése I	1:14

A Qt keretrendszer	
Szövegkezelés	
<ul style="list-style-type: none"> Qt-ben a karakterek 16 bites Unicode (UTF8) kódolásúak <ul style="list-style-type: none"> már a <code>QObject</code> típus biztosítja a kódolási konverziót egy osztályszintű művelettel (<code>QObject::toUtf8</code>) A karakterek kezelését a <code>QChar</code> típus biztosítja, míg szövegre a <code>QString</code> típus alkalmazható <ul style="list-style-type: none"> kompatibilis a C++ standard könyvtár <code>string</code> típusával, pl.: <code>QString::fromStdString(stdstr)</code> megkülönbözteti az üres és a nem létező szöveget (<code>isNull</code>, <code>isEmpty</code>) alkalmas típuskonverziókra, pl. <code>QString::number(4)</code>, <code>str.toInt()</code> 	
ELTE IK, Eseményvezérelt alkalmazások fejlesztése I	1:15

A Qt keretrendszer	
Grafikus felületű alkalmazások vezérlése	
<ul style="list-style-type: none"> A konzol felületű alkalmazások csak billentyűzettől fogadnak bemenetet a programfutás meghatározott pontjain, a vezérlés módját mi szabályozzuk (pl. főprogram, menü) A grafikus felületű alkalmazások billentyűzettől és egérettől (érintőképernyőtől, stb.) fogadnak bemenetet a programfutás szinte bármely pillanatában, a vezérlés módja előre definiált A grafikus felületű alkalmazás vezérlését az <i>alkalmazás osztály</i> (<i>application class</i>) látja el <ul style="list-style-type: none"> kezeli a felhasználói bevitelt, a felület elemeit, beállítja az alkalmazástulajdonságokat (megjelenés, elérési útvonat, ...) a Qt-ben az alkalmazást a <code>QApplication</code> típus biztosítja 	
ELTE IK, Eseményvezérelt alkalmazások fejlesztése I	1:16

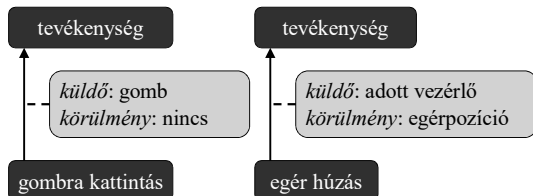
A Qt keretrendszer	
Eseménykezelés	
<ul style="list-style-type: none"> A grafikus felületen tehát számos módon és ponton kezdeményezhetünk interakciót a programmal A program által kezelhető, lereagálható interakciókat nevezzük <i>eseményeknek</i> (<i>event/signal</i>), az interakció kezdeményezését nevezzük az esemény <i>kiváltásának</i> <ul style="list-style-type: none"> pl.: gombra kattintás, egér húzás, listaelem kiválasztás 	
<div style="display: flex; justify-content: space-around; margin-top: 10px;"> <div style="border: 1px solid black; background-color: #333; color: white; padding: 5px 10px; border-radius: 5px;">gombra kattintás</div> <div style="border: 1px solid black; background-color: #333; color: white; padding: 5px 10px; border-radius: 5px;">egér húzás</div> </div>	
ELTE IK, Eseményvezérelt alkalmazások fejlesztése I	1:17

A Qt keretrendszer	
Eseménykezelés	
<ul style="list-style-type: none"> Az eseménynek van: <ul style="list-style-type: none"> <i>küldője</i> (<i>sender</i>): kiváltja az eseményt, pl. gomb, lista <i>körülményei</i> (<i>arguments</i>): meghatározza az esemény paramétereit, pl. egér pozíciója a húzáskor, kiválasztott listaelem indexe 	
ELTE IK, Eseményvezérelt alkalmazások fejlesztése I	1:18

A Qt keretrendszer

Eseménykezelés

- Az eseményekre reagálva a program futtathat egy alprogramot, ezt nevezzük *eseménykezelőnek* (*event handler/slot*)
- ha nem biztosítunk eseménykezelőt az eseményhez, akkor az *lekezeletlen* marad



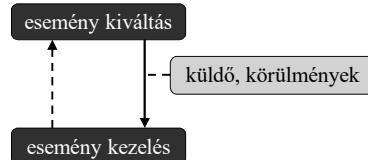
ELTE IK, Eseményvezérelt alkalmazások fejlesztése I

1:19

A Qt keretrendszer

Eseménykezelés

- Az *összetett események* úgy valósulnak meg, hogy a program egy egyszerű eseményre kivált egy másikat
- pl.: az egérrel kattintunk, és az egér a gombon van, akkor kiváltódik a gomb kattintása esemény
- tehát az eseménykezelés egy több lépcsős, *ciklikus folyamat*



ELTE IK, Eseményvezérelt alkalmazások fejlesztése I

1:20

A Qt keretrendszer

Eseménykezelő társítás

- Az eseménykezeléshez összekapcsoljuk az eseményt az eseménykezelővel, ezt *társításnak* nevezzük
- Qt-ban ehhez a `connect` metódust használjuk, pl.:
`connect(&button, SIGNAL(clicked()), &app, SLOT(quit()));`
- megadjuk, mely küldő objektum (*sender*) mely eseményére (*SIGNAL*) mely fogadó objektum (*receiver*) mely eseménykezelője (*SLOT*) kell, hogy fusson
- mindig mutatókat adunk meg, bármely két alkalmas objektum összeköthető
- a kötés elvégezhető `QObject` leszármazott típusban, illetve statikus metódus hivatkozással

ELTE IK, Eseményvezérelt alkalmazások fejlesztése I

1:21

A Qt keretrendszer

Eseménykezelő társítás

```
#include <QApplication>
#include <QPushButton>

int main(int argc, char *argv[]){
    QApplication app(argc, argv); // alkalmazás
    QPushButton quit("Quit"); // gomb
    quit.resize(75, 30); // méret
    quit.setFont(QFont("Times", 18, QFont::Bold));
    // betűtípus
    QObject::connect(&quit, SIGNAL(clicked()),
                    &app, SLOT(quit()));
    quit.show(); // gomb megjelenítése
    return app.exec();
}
```

ELTE IK, Eseményvezérelt alkalmazások fejlesztése I

1:22