

Zaawansowane programowanie w C++

Dr Michał Tanaś (<http://mtanas.web.amu.edu.pl/>)

QT - Meta Objects System

```
class MojaKlasa : public QObject {
```

```
    Q_OBJECT
```

```
signals:
```

```
    void MojSygnal1();
```

```
    void MojSygnal2(int);
```

```
public slots:
```

```
    void MojSlot1();
```

```
    void MojSlot2(int);
```

```
    void MojSlot3(int,int,int);
```

```
};
```

QT - Meta Objects System

Sygnały:

- Funkcje oznaczone **signals**: są wywoływane w nieprzewidywalnych momentach czasu (użytkownik kliknął na przycisk, upłynął limit czasu zegara, itp.).
- Dla funkcji oznaczonych jako **signals**: pisze się WYŁĄCZNIE DEKLARACJĘ! Definicje tych funkcji tworzone są automatycznie przez moc (Meta Object Compiler).
- Nie jest istotne czy funkcje **signals**: są publiczne czy prywatne.
- Pełnią rolę przerwań programowych albo zdarzeń QT

QT - Meta Objects System

Sloty:

- Są (prawie) normalnymi metodami klasy.
- Posiadają normalną deklarację i definicję.
- Mogą być wywoływane w normalny sposób.
- Mogą być public, protected lub private.
- Mogą być wywoływane w reakcji na wywołanie funkcji typu sygnał
- Pełnią rolę procedur obsługi przerwań programowych w QT.

QT - Meta Objects System

```
bool QObject::connect(  
    QObject *sender, char *signal,  
    QObject *receiver, char *slot)
```

- Powoduje że funkcja **slot** stanie się procedurą obsługi zdarzenia **signal**.
- **sender** jest wskaźnikiem do obiektu zawierającego sygnał
- **receiver** jest wskaźnikiem do obiektu zawierającego slot

QT - Meta Objects System

```
bool QObject::connect(  
    QObject *sender, char *signal,  
    QObject *receiver, char *slot)
```

- Sygnał i slot podaje się przy pomocy makrodefinicji **SIGNAL()** i **SLOT()**.
- Sygnał i slot muszą mieć takie same argumenty
- **W makrach SIGNAL i SLOT nie podaje się nazw argumentów (tylko typy).**

QT - Meta Objects System

Możliwości `connect`

Podłączenie slotu do sygnału:

```
connect(obiekt1,SIGNAL(signal1()),obiekt2,SLOT(slot1()));
```

Podłączenie wielu slotów do sygnału:

```
connect(obiekt1,SIGNAL(signal1()),obiekt2,SLOT(slot1()));
```

```
connect(obiekt1,SIGNAL(signal1()),obiekt2,SLOT(slot2()));
```

```
connect(obiekt1,SIGNAL(signal1()),obiekt3,SLOT(slot1()));
```

QT - Meta Objects System

Możliwości `connect`

Podłączenie sygnału do sygnału:

```
connect(obiekt1,SIGNAL(signal1()),obiekt2,SIGNAL(signal1()));
```

Podłączenie wielu sygnałów do sygnału:

```
connect(obiekt1,SIGNAL(signal1()),obiekt2,SIGNAL(signal1()));
```

```
connect(obiekt1,SIGNAL(signal1()),obiekt2,SIGNAL(signal1()));
```

```
connect(obiekt1,SIGNAL(signal1()),obiekt2,SIGNAL(signal1()));
```

QT - Meta Objects System

Sygnały z parametrami:

```
connect(obiekt1,SIGNAL(signal1(int,double)),  
obiekt2,SLOT(slot1(int,double)));
```

O czym trzeba pamiętać:

- Argumenty sygnału i slotu muszą być zgodne (inaczej `connect` nie będzie działać).
- **W makrach `SIGNAL()` i `SLOT()` nie wolno podawać nazw zmiennych, podaje się tylko i wyłącznie typy!**

QT - Meta Objects System

Rozłączenie konkretnego sygnału i slotu:

```
disconnect(obiekt1,SIGNAL(signal1()), obiekt2,SLOT(slot2()));
```

Rozłączenie konkretnego sygnału i wszystkich slotów/sygnałów z danego obiektu:

```
disconnect(obiekt1,SIGNAL(signal1()),obiekt2);
```

QT - Meta Objects System

Rozłączenie konkretnego sygnału i wszystkich slotów/sygnałów z wszystkich obiektów:

```
disconnect(obiekt1,SIGNAL(signal1()));
```

Rozłączenie wszystkich sygnałów danego obiektu i wszystkich slotów/sygnałów z wszystkich obiektów:

```
disconnect(obiekt1);
```

QT - Meta Objects System

Wysyłanie sygnałów – przy pomocy `emit`

```
emit MojSygna1();  
emit MojSygna1(20,x);
```

QT - Meta Objects System

Kompilowanie klas z `QObject`:

1. Na każdym pliku zawierającym definicję klasy z `QObject` musi być użyte narzędzie `moc` (meta object compiler) wchodzące w skład pakietu QT.
2. `moc` dla każdego takiego pliku utworzy DODATKOWY plik źródłowy.
3. Pliki źródłowe tworzone przez `moc` muszą być normalnie skompilowane i zlinkowane tak jak gdyby były częściami kodu źródłowego.

QT - Meta Objects System

Ograniczenia stosowania sygnałów i slotów:

- Nie można stosować `Q_OBJECT` w szablonach (`template`)
- Nie można stosować `Q_OBJECT` w klasach zagnieżdżonych.
- W przypadku gdy klasa ma więcej niż jedną klasę-rodzica:
 - **Klasa zawierająca `Q_OBJECT` musi być wymieniona jako pierwsza na liście klas-rodziców.**
 - Żadna z pozostałych klas-rodziców nie może zawierać `Q_OBJECT`

QT - Meta Objects System

Ograniczenia stosowania sygnałów i slotów:

- Wskaźniki na funkcje można używać jako argumentów sygnałów slotów wyłącznie za pośrednictwem `typedef`.
- W sekcjach `signals:` i `slots:` nie można deklarować funkcji i klas zaprzyjaźnionych.
- W sekcji `signals:` i `slots:` nie można deklarować konstruktorów.
- **Odziedziczone sloty i sygnały nie mogą być zmieniane w klasie potomnej.**
- W wywołaniach sygnałów i slotów w argumentach nie mogą być stosowane makrodefinicje.