

Zaawansowane programowanie w C++

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

Wielowątkowość - QThread

Szkielet klasy, której obiekty mają być uruchamiane w osobnym wątku:

```
class Worker : public QObject {  
    Q_OBJECT  
public:  
    Worker();  
    ~Worker();  
public slots:  
    void process();  
signals:  
    void finished();  
    void error(QString err);  
};
```

Wielowątkowość - QThread

Kod, który ma być wykonany w osobnym wątku powinien być umieszczony wewnątrz metody `process()`

```
class Worker : public QObject {  
    Q_OBJECT  
public:  
    Worker();  
    ~Worker();  
public slots:  
    void process();  
signals:  
    void finished();  
    void error(QString err);  
};
```

Wielowątkowość - QThread

```
QThread* thread = new QThread();
```

```
Worker* worker = new Worker();
```

```
worker->moveToThread(thread);
```

```
connect( worker, &Worker::error, this, &MyClass::errorString);
```

```
connect( thread, &QThread::started, worker, &Worker::process);
```

```
connect( worker, &Worker::finished, thread, &QThread::quit);
```

```
connect( worker, &Worker::finished, worker, &Worker::deleteLater);
```

```
connect( thread, &QThread::finished, thread, &QThread::deleteLater);
```

```
thread->start();
```

Wielowątkowość - QThread

QThread

- Klasa kontrolująca wątek (klasę Worker)
- Sygnały:
 - `started()` – emitowany w momencie uruchomienia wątku
 - `finished()` – emitowany w momencie zakończenia wątku
- Sloty:
 - `start()` – uruchomienie wątku
 - `exit(int)`, `quit()`, `terminate()` – zakończenie działania wątku

Wielowątkowość - QThread

`deleteLater()`

- Metoda klasy `QObject`
- Usuwa obiekt z pamięci w momencie gdy zakończy się przetwarzanie bieżącego zdarzenia
- Uwaga! To nie jest odpowiednik garbage collector z Java