

QT Signals & Slots

Teoria

Mechanizm Qt do porozumiewania się elementów aplikacji zaimplementowany w oparciu o wzorzec Observer. Podczas wystąpienia konkretnego zdarzenia (*przyciśnięcie przycisku, najechanie myszą*) emitowany jest Sygnał. Natomiast Slot to nic innego jak funkcja przypisana do obsługi danego sygnału. Po połączeniu sygnału ze slotem za pomocą metody `connect()` taka funkcja jest wykonywana automatycznie, gdy sygnał zostanie wyemitowany.

Cechy slotów i sygnałów:

- Sygnał może być połączony z wieloma Slotami
- Sygnał może być połączony z innym Sygnałem
- Slot może być połączony z wieloma Sygnałami
- połączenia mogą być synchroniczne (*bezpośrednie*) lub asynchroniczne (*kolejkowane*)
- połączenia można tworzyć pomiędzy wątkami
- metoda `connect()` odpowiada za łączenie Sygnału ze Slotem
- emitowanie sygnału w PyQt odbywa się przy użyciu metody `emit()`
- jest wiele predefiniowanych Sygnałów i Slotów, ale można też tworzyć własne sygnały
- metoda `disconnect()` odłącza od siebie Sygnał od Slotu

Praktyka

We frameworku Qt dla Pythona istnieją dwa podejścia do obsługi Sygnałów i Slotów.

Stare podejście:

```
QtCore.SIGNAL( SYGNAŁ , funkcja )
QtCore.SLOT()
```

```
QObject.connect( Widget , SIGNAL( 'SYGNAŁ()' , funkcja ) )
```

Nowe podejście:

```
Signal(typ_danych)
Slot()
Widget.SYGNAŁ.connect( funkcja )
```

Przykłady kodu

Przykład (bez parametrów):

```
import sys
from PySide2.QtWidgets import QApplication, QPushButton
from PySide2.QtCore import SIGNAL, QObject      # stare podejście

def mojaFunkcja():
    print("Kliknięto mnie!")

app = QApplication(sys.argv)
btn = QPushButton("Kliknij mnie")

QObject.connect(btn, SIGNAL('clicked()'), mojaFunkcja)    # stare podejście
btn.clicked.connect(mojaFunkcja)    # nowe podejście

btn.show()
sys.exit(app.exec_())
```

Przykład (z parametrami):

```
import sys
from PySide2.QtWidgets import QApplication, QPushButton
from PySide2.QtCore import QObject, Signal, Slot

def mojaFunkcja(mamArgument):
    print("Kliknięto mnie!")
    print(mamArgument)

class Sygnalista(QObject):
    sygnalek = Signal(str)

app = QApplication(sys.argv)
btn = QPushButton("Kliknij mnie")

syg = Sygnalista()
syg.sygnalek.connect(mojaFunkcja)
syg.sygnalek.emit("Bo TAK!")

btn.show()
sys.exit(app.exec_())
```