

Weryfikatory, designer – zasoby, rcc

Bogdan Kreczmer

bogdan.kreczmer@pwr.wroc.pl

Zakład Podstaw Cybernetyki i Robotyki
Instytut Informatyki, Automatyki i Robotyki
Politechnika Wrocławska

Kurs: Wizualizacja danych sensorycznych

Copyright©2017 Bogdan Kreczmer

Niniejszy dokument zawiera materiały do wykładu dotyczącego programowania obiektowego. Jest on udostępniony pod warunkiem wykorzystania wyłącznie do własnych prywatnych potrzeb i może on być kopiowany wyłącznie w całości, razem z niniejszą stroną tytułową.

Niniejsza prezentacja została wykonana przy użyciu systemu składu \LaTeX oraz stylu beamer, którego autorem jest Till Tantau.

Strona domowa projektu Beamer:

<http://latex-beamer.sourceforge.net>

1 Weryfikatory

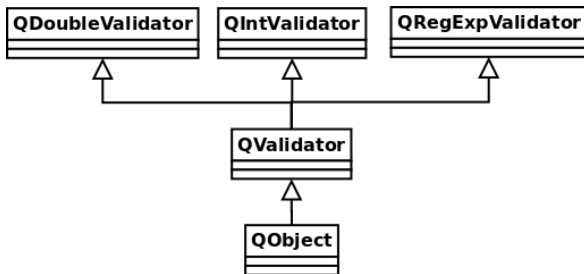
2 Łączenie sygnałów i slotów

Zestawienie weryfikatorów

QIntValidator – weryfikator dla liczb całkowitych.

QDoubleValidator – weryfikator dla liczb zmiennoprzecinkowych.

QRegExpValidator – weryfikator wykorzystujący wyrażenia regularne.



Przykład skojarzenia

```
class Kanwa: public QWidget {
    Q_OBJECT
public:
    Kanwa(QWidget *wRodzic = nullptr);
};

Kanwa::Kanwa(QWidget *wRodzic): QWidget(wRodzic)
{
    QLineEdit *wEdycja = new WierszEdycji(this);
    wEdycja->setValidator( new QIntValidator(this) );
}
```

Przykład połączenia

```
class Kanwa: public QWidget {
    Q_OBJECT
public:
    Kanwa(QWidget *wRodzic = nullptr);
};

Kanwa::Kanwa(QWidget *wRodzic):    QWidget(wRodzic)
{
    QLineEdit *wEdycja = new WierszEdycji(this);
    wEdycja->setValidator( new QIntValidator(this) );
}
```

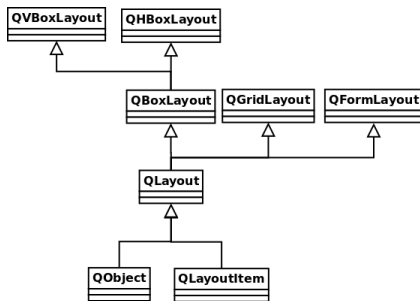
Zestawienie weryfikatorów

QHBoxLayout – zarządca geometrii horyzontalnej

QVBoxLayout – zarządca geometrii wertykalnej

QGridLayout – zarządca geometrii kraty

QFormLayout – grupuje etykiety i widżety

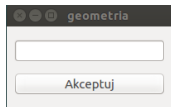


Przykład skojarzenia zarządcy geometrii

```
Kanwa::Kanwa(QWidget *wRodzic): QWidget(wRodzic)
{
    QLineEdit *wEdycja = new QLineEdit(this);
    wEdycja->setValidator( new QIntValidator(this) );

    QPushButton *wPrzycisk = new QPushButton(" Akceptuj" );
    QVBoxLayout *wWZarzdca = new QVBoxLayout(this);

    wWZarzdca->addWidget(wEdycja);
    wWZarzdca->addWidget(wPrzycisk);
}
```

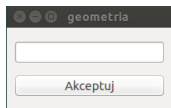


Przykład skojarzenia zarządcy geometrii

```
Kanwa::Kanwa(QWidget *wRodzic):  QWidget(wRodzic)
{
    QLineEdit *wEdycja = new QLineEdit(this);
    wEdycja->setValidator( new QIntValidator(this) );

    QPushButton *wPrzycisk = new QPushButton(" Akceptuj" );
    QVBoxLayout *wWZarzdca = new QVBoxLayout();

    wWZarzdca->addWidget( wEdycja );
    wWZarzdca->addWidget( wPrzycisk );
    setLayout( wWZarzdca );
}
```



Zagnieżdżanie zarządzców geometrii

```

Kanwa::Kanwa(QWidget *wRodzic): QWidget(wRodzic)
{
    QLineEdit *wEdycja = new QLineEdit(this);

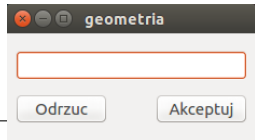
    QPushButton *wPrzycisk_Akce = new QPushButton("Akceptuj");
    QPushButton *wPrzycisk_Odrz = new QPushButton("Odrzuc");
    QVBoxLayout *wWZarządca = new QVBoxLayout();

    wWZarządca->addWidget(wEdycja);

    QHBoxLayout *wHZarządca = new QHBoxLayout();
    wHZarządca->addWidget(wPrzycisk_Odrz);
    wHZarządca->addItem(
        new QSpacerItem(10,10, QSizePolicy::Expanding,
                        QSizePolicy::Minimum));
    wHZarządca->addWidget(wPrzycisk_Akce);

    wWZarządca->addLayout(wHZarządca);
    setLayout(wWZarządca);
}

```



Zagnieżdżanie zarządzców geometrii

```

Kanwa :: Kanwa(QWidget *wRodzic):  QWidget(wRodzic)
{
    QLineEdit *wEdycja = new QLineEdit( this );

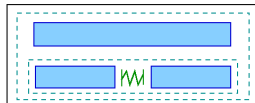
    QPushButton *wPrzycisk_Akce = new QPushButton(" Akceptuj" );
    QPushButton *wPrzycisk_Odrz = new QPushButton(" Odrzuc" );
    QVBoxLayout *wWZarządca = new QVBoxLayout ();

    wWZarządca->addWidget( wEdycja );

    QHBoxLayout *wHZarządca = new QHBoxLayout ();
    wHZarządca->addWidget( wPrzycisk_Odrz );
    wHZarządca->addItem (
        new QSpacerItem(10,10, QSizePolicy :: Expanding ,
                        QSizePolicy :: Minimum));
    wHZarządca->addWidget( wPrzycisk_Akce );

    wWZarządca->addLayout( wHZarządca );
    setLayout( wWZarządca );
}

```



Tradycyjne łączenie sygnałów i slotów

```
class Kanwa: public QWidget {
    Q_OBJECT
public:
    Kanwa(QWidget *wRodzic = nullptr);

public slots:
    void GdyNacisniecie();
};

Kanwa::Kanwa(QWidget *wRodzic): QWidget(wRodzic)
{
    ...
    QPushButton *wPrzycisk_Akce = new QPushButton(" Akceptuj" );
    ...
    connect( wPrzycisk_Akce , SIGNAL( clicked() ) ,
            this , SLOT( GdyNacisniecie() ) );
}
```

Automatyczne łączenie sygnałów i slotów

```
class Kanwa: public QWidget {
    Q_OBJECT
public:
    Kanwa(QWidget *wRodzic = nullptr);

public slots:
    void on_Przycisk_Akceptuj_clicked ();
};

Kanwa::Kanwa(QWidget *wRodzic): QWidget(wRodzic)
{
    ...
    QPushButton *wPrzycisk_Akce = new QPushButton(" Akceptuj" );
    wPrzycisk_Akce->setObjectName(" Przycisk_Akceptuj" );
    ...
    QMetaObject::connectSlotsByName( this );
}
```

Schemat tworzenia nazwy slotu

```

class Kanwa: public QWidget {
    ...
    public slots:
        void on_Przycisk_Akceptuj_clicked ();
};

Kanwa::Kanwa(QWidget *wRodzic):   QWidget(wRodzic)
{
    ...
    QPushButton *wPrzycisk_Akce = new QPushButton(" Akceptuj" );
    wPrzycisk_Akce->setObjectName(" Przycisk_Akceptuj" );
    ...
}

```

Schemat nazwy slotu

on_<nazwa_obiektu>_<nazwa_sygnału>

Koniec prezentacji
Dziękuję za uwagę