

# Laboratorium Praktyka Programowania Python PyQt5

Prowadzący:

Jakub Szyguła: [jakub.szygula@polsl.pl](mailto:jakub.szygula@polsl.pl)

Dariusz Marek: [dariusz.marek@polsl.pl](mailto:dariusz.marek@polsl.pl)

Pokój: 906

18 października 2019

## Spis treści

<b>1</b>	<b>Wprowadzenie</b>	<b>2</b>
1.1	Wymagania . . . . .	2
1.2	Dokumentacja . . . . .	2
1.3	Instalacja modułu . . . . .	2
<b>2</b>	<b>Proste aplikacje okienkowe</b>	<b>2</b>
2.1	Przykład 1 - Aplikacja wyświetlająca tekst . . . . .	2
2.2	Przykład 2 - Aplikacja wyświetlająca przycisk i dodatkowe okno zapytania	3
2.3	Przykład 3 - Wyświetlanie grupy przycisków . . . . .	4
2.4	Przykład 4 - Aplikacja wyświetlająca obraz . . . . .	5
2.5	Przykład 5 - Aplikacja umożliwiająca wybór pliku . . . . .	5
2.6	Przykład 6 - Aplikacja wyświetlająca okno tekstowe . . . . .	6
<b>3</b>	<b>Zadania do wykonania</b>	<b>6</b>
3.1	Zadanie 1 . . . . .	6
3.2	Zadanie 2 . . . . .	6
3.3	Zadanie 3 . . . . .	7
<b>4</b>	<b>Sprawozdanie</b>	<b>7</b>

# 1 Wprowadzenie

PyQt to biblioteka (zbiór bibliotek) umożliwiających projektowanie aplikacji okienkowych wykorzystujący framework Qt. Umożliwia tworzenie aplikacji na wiele platform i systemów operacyjnych.

## 1.1 Wymagania

Instrukcja została przygotowana z wykorzystaniem Python 3.7 i wymaga posiadania:

1. Python 3.7
2. Dodanego Python'a do Path w zmiennych środowiskowych systemu Windows np.:
  - (a) C : \Python37\
  - (b) C : \Python37\Scripts\

## 1.2 Dokumentacja

<https://www.riverbankcomputing.com/static/Docs/PyQt5/>

## 1.3 Instalacja modułu

Instalacja modułu PyQt5 za pomocą modułu pip do Python'a. W konsoli (cmd) należy wpisać:

```
pip install pyqt5
```

# 2 Proste aplikacje okienkowe

Tworzenie prostych aplikacji okienkowych za pomocą PyQt polega na:

1. Inicjalizacji okna
2. Stworzeniu kontrolki np. tekstu czy przycisku wywołującego odpowiednią funkcję
3. Podłączeniu kontroli do aplikacji
4. Wyrenderowaniu okna aplikacji

## 2.1 Przykład 1 - Aplikacja wyświetlająca tekst

Dołączanie modułów QApplication, QLabel z pakietu PyQt5.QtWidgets

```
from PyQt5.QtWidgets import QApplication, QLabel
```

Inicjalizacja okna aplikacji

```
app = QApplication([])
```

Tworzenie prostego tekstu, który zostanie wyświetlony w aplikacji

```
label = QLabel("Tekst w pierwszej aplikacji okienkowej")
```

Wywołanie aplikacji z wyświetleniem tekstu

```
label.show()  
app.exec_()
```

## 2.2 Przykład 2 - Aplikacja wyświetlająca przycisk i dodatkowe okno zapytania

Dołączanie modułów QApplication, QMessageBox, QPushButton z pakietu PyQt5.QtWidgets

```
from PyQt5.QtWidgets import QApplication, QMessageBox, QPushButton
```

Inicjalizacja okna aplikacji

```
app = QApplication([])
```

Inicjalizacja przycisku

```
button = QPushButton('To jest przycisk')
```

Stworzenie funkcji wyświetlającej okno powiadomienia

```
def on_button_clicked():  
    alert = QMessageBox()  
    alert.setText('Przycisk został naciśnięty!')  
    alert.exec_()
```

Przypisanie funkcji z punktu 3 do przycisku

```
button.clicked.connect(on_button_clicked)
```

Dodanie wywołania aplikacji z wyświetleniem przycisku

```
button.show()  
app.exec_()
```

## 2.3 Przykład 3 - Wyświetlanie grupy przycisków

```
from PyQt5.QtWidgets import QApplication, QPushButton, QGroupBox,   
    ↳ QGridLayout  
  
app = QApplication([])  
  
horizontalGroupBox = QGroupBox("Nazwa grupy")  
layout = QGridLayout()  
  
layout.addWidget(QPushButton('B1'), 0, 0)  
layout.addWidget(QPushButton('B2'), 0, 1)  
layout.addWidget(QPushButton('B3'), 0, 2)  
layout.addWidget(QPushButton('B4'), 1, 0)  
layout.addWidget(QPushButton('B5'), 1, 1)  
layout.addWidget(QPushButton('B6'), 1, 2)  
  
horizontalGroupBox.setLayout(layout)  
  
horizontalGroupBox.show()  
app.exec_()
```

## 2.4 Przykład 4 - Aplikacja wyświetlająca obraz

```
from PyQt5.QtWidgets import QApplication, QWidget, QLabel
from PyQt5.QtGui import QIcon, QPixmap

app = QApplication([])
widget = QWidget()

widget.setWindowTitle("Title!!")

label = QLabel(widget)
pixmap = QPixmap("<ścieżka do pliku z obrazem>")
label.setPixmap(pixmap)
widget.resize(pixmap.width(), pixmap.height())

widget.show()
app.exec_()
```

## 2.5 Przykład 5 - Aplikacja umożliwiająca wybór pliku

```
from PyQt5.QtWidgets import QApplication, QWidget, QInputDialog, QLineEdit, QFileDialog

app = QApplication([])
widget = QWidget()

widget.setWindowTitle("Title!!")

options = QFileDialog.Options()
options |= QFileDialog.DontUseNativeDialog
fileName, _ = QFileDialog.getOpenFileName(widget, "Wybierz plik obrazu",
    "", "All Files (*);;Python Files (*.py)", options=options)
if fileName:
    print(fileName)

widget.show()
app.exec_()
```

## 2.6 Przykład 6 - Aplikacja wyświetlająca okno tekstowe

```
from PyQt5.QtWidgets import QApplication, QWidget, QDialog, \
    QLineEdit, QFileDialog, QTextEdit

app = QApplication([])
widget = QWidget()

widget.setWindowTitle("Title!!")

text = QTextEdit(widget)
text.insertPlainText("You can write text here.\n")
text.resize(400,200)

widget.show()
app.exec_()
```

## 3 Zadania do wykonania

Wykorzystując przykłady, dokumentację oraz Internet napisz 3 aplikacje.

### 3.1 Zadanie 1

Napisz aplikację, która będzie umożliwiać:

1. Wybranie pliku z obrazem (wybór tylko plików o odpowiednim rozszerzeniu)
2. Wyświetlenie wybranego obrazu z punktu 1
3. Zamknięcie aplikacji za pomocą dodatkowego przycisku

### 3.2 Zadanie 2

Napisz prosty notatnik, który będzie umożliwiać:

1. Wybranie pliku za pomocą przycisku
2. Wyświetlanie otwartego pliku
3. Zapis tekstu do pliku (wskazówka: `QFileDialog.getSaveFileName(...)`)

### 3.3 Zadanie 3

Napisz prostą aplikację skleającą tekst z dwóch pól tekstowych (A i B), wyświetlającą wynik w trzecim polu (C). Pole wynikowe powinno być automatycznie aktualizowane po zmianie tekstów zawartych w polach (A i B). Pole wynikowe (C) powinno mieć zablokowaną możliwość wpisywania.

Wskazówka:

```
pole_tekstowe.setDisabled(True/False)
```

```
pole_tekstowe.textChanged.connect(funkcja)
```

## 4 Sprawozdanie

Proszę przesłać na platformę zdalnej edukacji (<https://platforma.polsl.pl/>):

1. Kody źródłowe zadań
2. Krótkie sprawozdanie z wykonanych zadań
3. Skład sekcji