
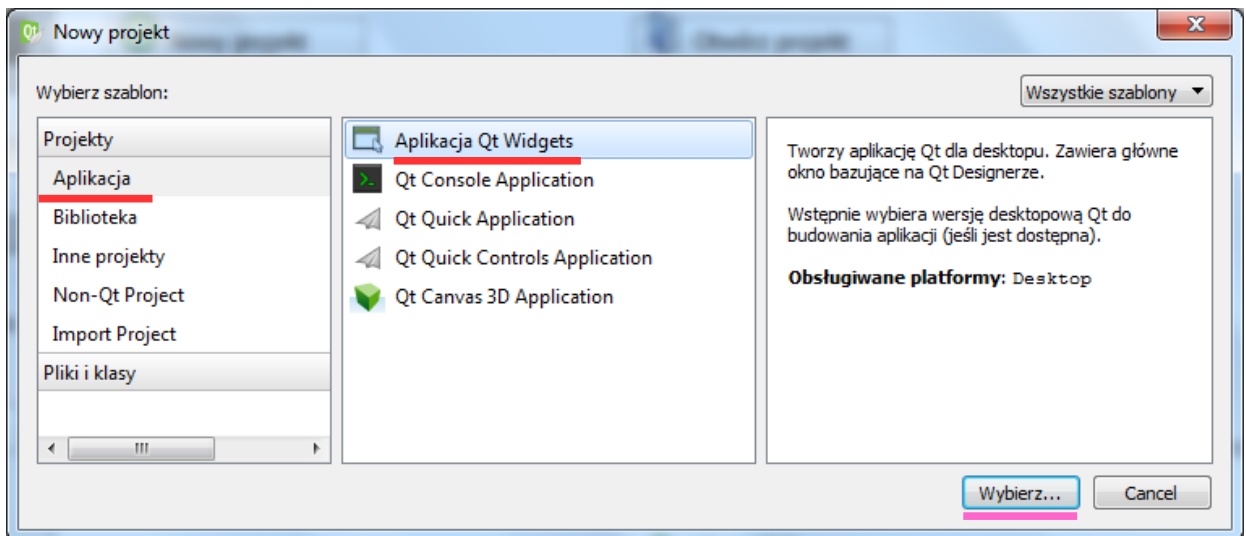


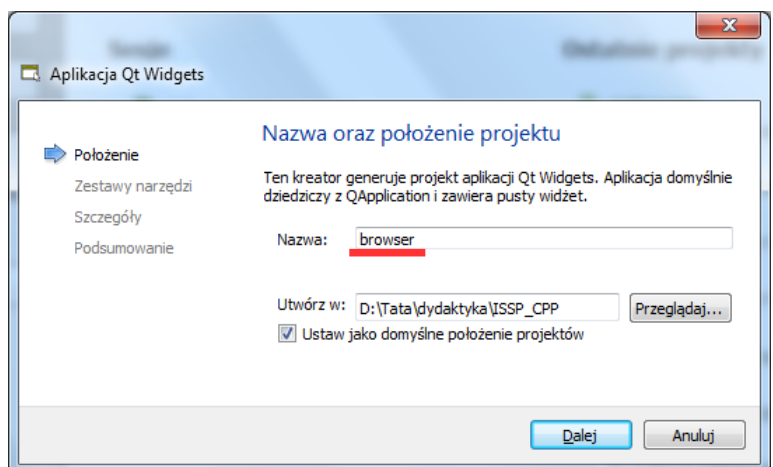
# ISSP/C++: tydzień 1

## Zadanie: napisać prostą, jednak możliwie jak najbardziej funkcjonalną przeglądarkę WWW

1. Sprawdź, czy masz dostęp do w miarę nowego wydania biblioteki Qt, w tym do programu QtCreator. Jeśli nie, zainstaluj go ze strony <http://www.qt.io/>. Dalej posługiwać się będę QtCreatorem w wersji 3.6.0 i Qt 5.5.1 (Built on Dec 15 2015 01:01:38 ) działającym w systemie Windows. Prawdopodobnie każde Qt w wersji 5.x oraz dowolny system operacyjny będą równie dobre.
2. Uruchom QtCreatora.
3. Utwórz Nowy projekt (przycisk:  Nowy projekt lub ctrl-N lub z menu Plik).
4. Wybierz szablon: 1: Aplikacja; 2: Aplikacja Qt Widgets:

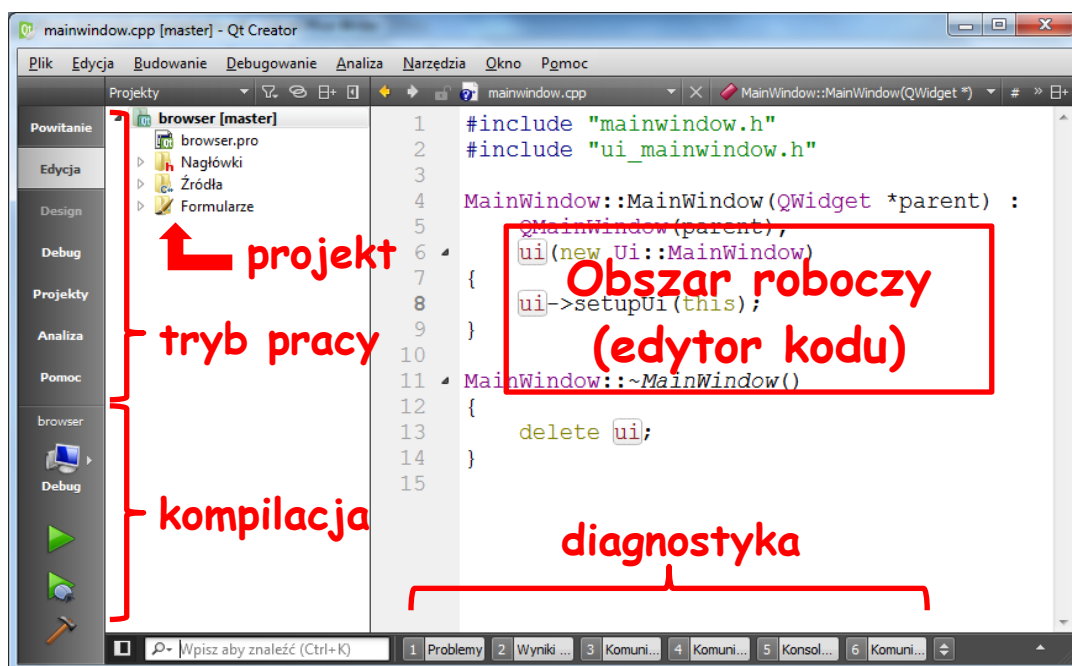


5. W kolejnym okienku ("Położenie") wybierz katalog, w którym będziesz umieszczać swoje programy pisane w QtCreatorze oraz podaj nazwę "Nazwę". Nazwa to nazwa katalogu, w którym umieścisz ten konkretny program. Ja jako nazwę podaję "browser":
6. W kolejnym okienku, "Wybór zestawu narzędzi", prawdopodobnie wystarczy kliknąć "dalej", akceptując wartości domyślne.

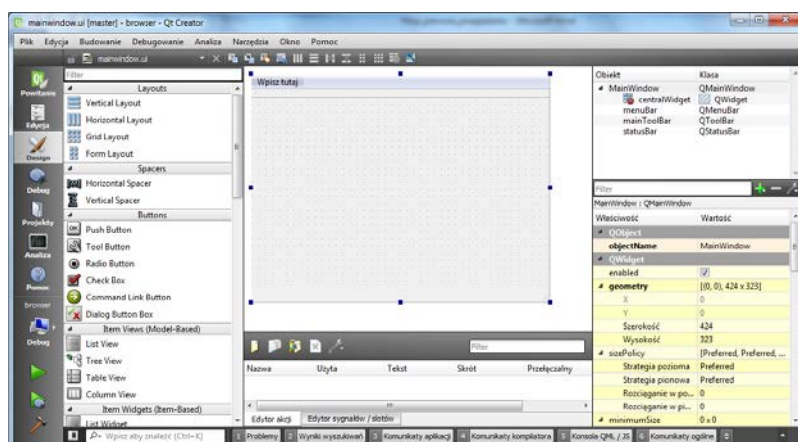
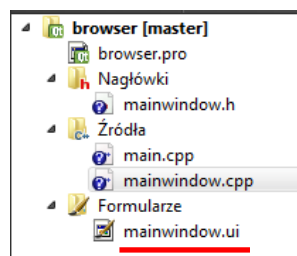


7. W kolejnym okienku też wystarczy zaakceptować wartości domyślne. Upewnij się jednak, że zaznaczono opcję "Wygeneruj formularz".
8. W kolejnym okienku nie dodawaj projektu do żadnego systemu kontroli wersji (bo stracisz mnóstwo czasu na jego konfigurację)
9. Zakończ konfigurację projektu.
10. Teraz widzisz okienko podobne do tego:

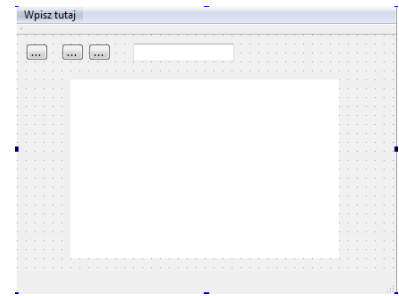
Nazwa klasy:	MainWindow
Klasa bazowa:	QMainWindow
Plik nagłówkowy:	mainwindow.h
Plik źródłowy:	mainwindow.cpp
Wygeneruj formularz:	<input checked="" type="checkbox"/>
Plik z formularzem:	mainwindow.ui



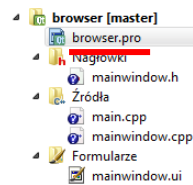
11. Skompiluj i uruchom program aby upewnić się, że QtCreator jest poprawnie skonfigurowany (np. zieloną strzałką z lewej strony na dole)
12. W panelu projektu rozwiń drzewo „Formularze” i wybierz plik `mainwindow.ui`. W ten sposób automatycznie przejdziesz w tryb projektowania graficznego interfejsu użytkownika, jak na rysunku poniżej



13. Z panelu widżetów (po prawej stronie) przeciągnij i upuść na deskę projektową trzy widżety typu ToolButton, jeden LineEdit i jeden WebView, jak na rysunku obok.



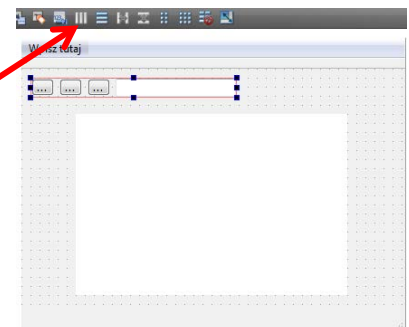
14. Program niestety się nie kompiluje. Wracamy do panelu projektu i otwieramy plik \*.pro (tu: browser.pro) i do zmiennej QT dopisujemy webkitwidgets.



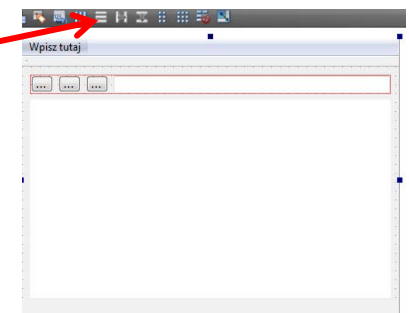
```
1 #-----
2 #
3 # Project created by QtCreator 2016-02-21T14:36:51
4 #
5 #-----
6
7 QT      += core gui webkitwidgets
8
9 greaterThan(QT_MAJOR_VERSION, 4): QT += widgets
10
11 TARGET = browser
12 TEMPLATE = app
13
14
15 SOURCES += main.cpp\
16          mainwindow.cpp
17
18 HEADERS += mainwindow.h
19
20 FORMS   += mainwindow.ui
```

15. Skompiluj i uruchom program.

16. Powrót do trybu projektowania graficznego interfejsu użytkownika. Zaznacz myszką trzy przyciski i pole edycji (np. przytrzymując klawisz ctrl lub zaznaczając myszką prostokąt przechodzący przez te widżety) i wybierz z paska narzędziowego przycisk rozmieść w poziomie (ctrl-H)

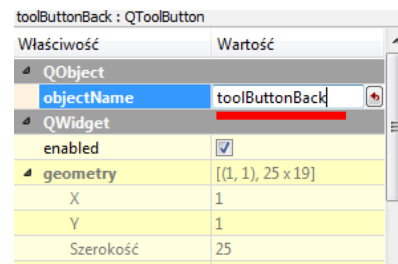


17. Kliknij w szare tło deski projektowej i z paska narzędzi wybierz rozmieść w pionie (ctrl-L) Sprawdź, że program się kompiluje, a okienko programu zachowuje się rozsądnie przy zmianie jego wielkości.



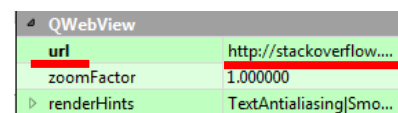
18. Usuń pasek narzędzi (tuż pod miejscem na menu). Łatwo domyślić się, jak to zrobić.

19. Zaznacz myszką pierwszy przycisk. W okienku „Edytor właściwości” (zwykle w prawym dolnym narożniku okna) nadaj mu bardziej znaczącą nazwę, np. `toolButtonBack`. Tej nazwy będziesz później używać w programie jako identyfikatora tego przycisku, dlatego nazwa powinna łatwo identyfikować dany element interfejsu użytkownika.



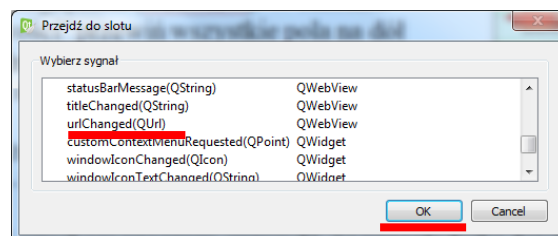
20. W podobny sposób nadaj rozsądne identyfikatory pozostałym przyciskom (np. `toolButtonForward`, `toolButtonReload`).
21. Kolejno kliknij podwójnie w każdy z przycisków i wpisz odpowiedni tekst: `&Back`, `&Forward`, `&Rewnid`. Nie zapomnij o ampersandzie (&).
22. Skompiluj i uruchom program. Sprawdź, że działają już kombinacje klawiszy `Alt-B`, `Alt-F` oraz `Alt-R`.

23. Kliknij obiekt przeglądarki (`webView`). W okienku „Edytor właściwości” przewiń wszystkie pola na dół i zainicjalizuj adres `url` przeglądarki, np. jako `http://stackoverflow.com/questions/tagged/c%2B%2B`.



24. Sprawdź, że możesz dokonać podglądu wyglądu okna swojej aplikacji kombinacją klawiszy `Ctrl-Alt-R` – bez uprzedniej kompilacji programu!
25. OK. Skompiluj i uruchom program, sprawdź, że przeglądarka działa.

26. Przechodzimy do działu o nazwie „magia”. Zaznacz obiekt przeglądarki (`webView`). Prawym klawiszem wyświetl menu kontekstowe tego widżetu. Wybierz opcję „przejdź do slotu”. Wybierz sygnał `urlChanged(QUrl)`. Wpisz kod:



```
void MainWindow::on_webView_urlChanged(const QUrl &url)
{
    ui->lineEdit->setText(url.toString());
}
```

Uwaga: zmieniłem standardową nazwę argumentu funkcji z `Arg1` na `url`.

27. Skompiluj i uruchom program. Kliknij w kilka linków i sprawdź, czy w polu edycji Twojej przeglądarki zawsze pojawia się prawidłowy url.
28. Wróć do widoku projektowania interfejsu użytkownika. Kliknij prawym klawiszem w pole edycji adresu WWW (`lineEdit`) i z menu kontekstowego wybierz „przejdź do slotu”, a następnie wybierz sygnał `editingFinished()`. Wpisz kod:

```
void MainWindow::on_lineEdit_editingFinished()
{
    QString adres = ui->lineEdit->text();
    ui->webView->setUrl(QUrl(adres));
}
```

29. Skompiluj i uruchom program. Sprawdź, czy po wpisaniu w polu adresu nowego url widżet przeglądarki wyświetla jego obraz.

30. Sprawdź, że Twoja przeglądarka obsługuje już protokół ftp (np. <ftp://ftp.funet.fi/pub/standards/RFC/rfc959.txt>) oraz https (np. <https://en.wikipedia.org/wiki/HTTPS>)

31. W analogiczny sposób wygeneruj szkielet obsługi sygnału clicked przycisku toolButtonBack. Oto kod:

```
void MainWindow::on_toolButtonBack_clicked()  
{  
    ui->webView->back();  
}
```

32. To samo dla przycisku toolButtonForward:

```
void MainWindow::on_toolButtonForward_clicked()  
{  
    ui->webView->forward();  
}
```

33. I dla przycisku toolButtonReload:

```
void MainWindow::on_toolButtonReload_clicked()  
{  
    ui->webView->reload();  
}
```

34. Skompiluj i uruchom program. Przetestuj działanie przycisków.


35. Wróć do slotu urlChanged(QUrl) obiektu webView. Dodaj kod wg poniższego schematu:

```
void MainWindow::on_webView_urlChanged(const QUrl &url)  
{  
    ui->lineEdit->setText(url.toString());  
    QWebHistory * phistory = ui->webView->history();  
    ui->toolButtonBack->setEnabled(phistory->canGoBack());  
    ui->toolButtonForward->setEnabled(phistory->canGoForward());  
}
```

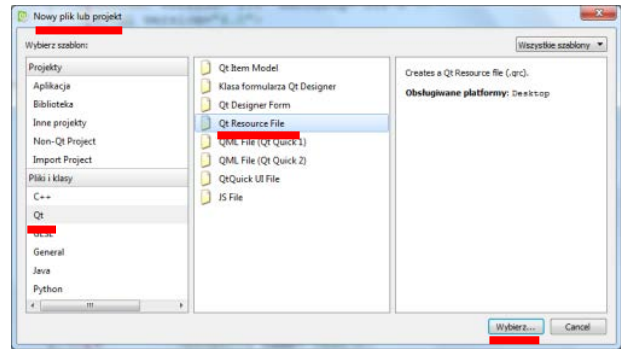
36. Program zapewne przestał się kompilować (sprawdź to!). Żeby pomóc kompilatorowi, dodaj następujący wiersz na samym początku pliku mainwindow.cpp:

```
#include <QWebHistory>
```

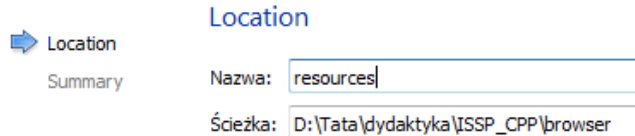
37. Skompiluj i uruchom program. Sprawdź, że przyciski są aktywne tylko wtedy, gdy ich uruchomienie ma sens.

38. Znajdź w Internecie lub swoim komputerze ikony, które odpowiadają operacjom „forward”, „back” i „reload”. Rozmiar: 16x16 pikseli. Format: np. png. Umieść je w podkatalogu ./img swojego projektu. 

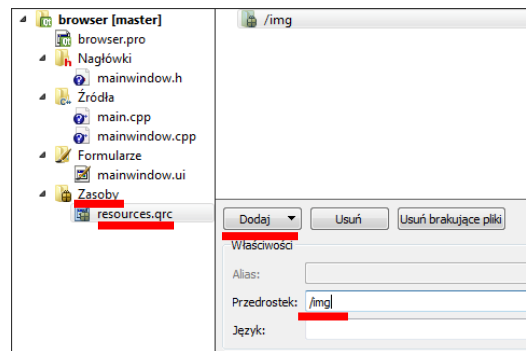
39. Wybierz z menu QtCreatora opcję Nowy plik lub projekt... (lub użyj kombinacji `ctrl-N`).  
Wybierz Qt/"Qt Resource File".



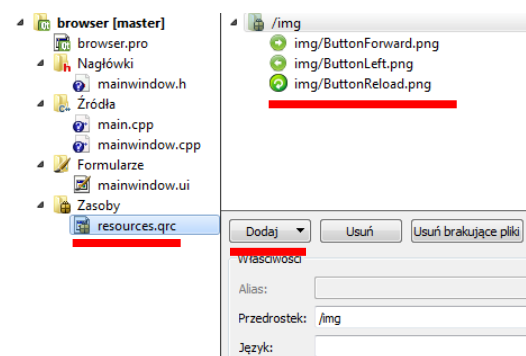
40. W kolejnym okienku wpisz jakąś rozsądną nazwę nowego pliku, np. `resources`.



41. Pomiń krok „Project management” (nie dodawaj nowego pliku do systemu kontroli wersji).
42. W panelu „projekt” pojawi się nowy element: „Zasoby”, a w nim plik `resources.qrc`. Otwórz go (z menu kontekstowego wybierz „Otwórz w edytorze”). W okienku, które się otworzy, kliknij przycisk `Dodaj` i wybierz `Dodaj` przedrostek. Ja wybrałem przedrostek `/img`, ale najlepiej wybrać po prostu `/`.

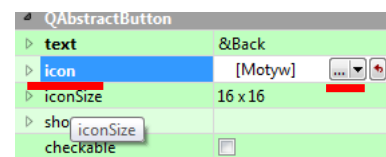


43. Ponownie kliknij przycisk `Dodaj` i dodaj do katalogu zasobów trzy ikonki, które masz przygotowane w katalogu `img`.



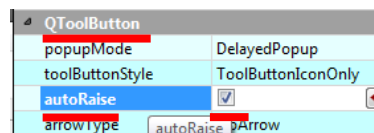
44. Koniecznie skompiluj program (inaczej inne moduły QtCreatora nie zobaczą nowych zasobów programu).

45. W edytorze interfejsu użytkownika kliknij (zaznacz) przycisk `toolButtonBack`. W panelu Edytor właściwości znajdź właściwość `icon` i wybierz jej wartość z pliku zasobów (kliknij w mały trójkącik). Zwróć uwagę na to, że z przycisku zniknął napis `Back`, a zastąpiła go ikona. Tym niemniej, skrót klawiaturowy `Alt-B` wciąż działa.



46. W analogiczny sposób podmień ikony dwóch pozostałych przycisków.

47. W panelu „Edycja właściwości” włącz właściwość `autoRaise` każdego z trzech przycisków (możesz je zaznaczyć trzy naraz i włączyć im tę właściwość jednocześnie).



48. Skompiluj i przetestuj program.

49. W pliku `mainwindow.cpp` wpisz kod, który zmieni nazwę programu wyświetlaną na jego belce tytułowej:

```
MainWindow::MainWindow(QWidget *parent) :
    QMainWindow(parent),
    ui(new Ui::MainWindow)
{
    ui->setupUi(this);
    this->setWindowTitle("Moja własna przeglądarka");
}
```

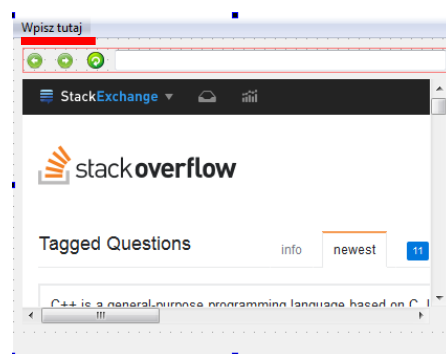
50. Skompiluj i przetestuj program.

51. Możesz jeszcze łatwo upiększyć swoją przeglądarkę. Przejdź do panelu „Edycja właściwości” (prawy dolny róg ekranu) i różnym widżetom ustaw następujące właściwości:

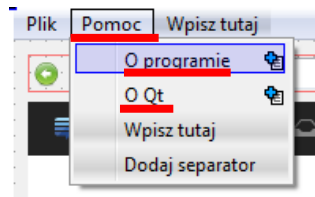
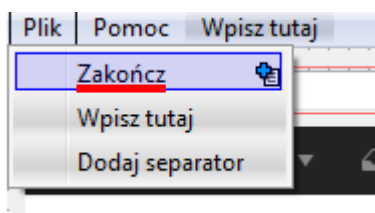
- `toolTip`
- `statusTip`
- `shortcut` (tylko przyciski)

52. Przetestuj program.

53. Jeśli masz trochę czasu, dodaj menu. W tym celu kliknij dwa razy w pole „Wpisz tutaj” w lewym górnym rogu projektu interfejsu użytkownika Twojego programu. Wpisz tam `&Plik`.

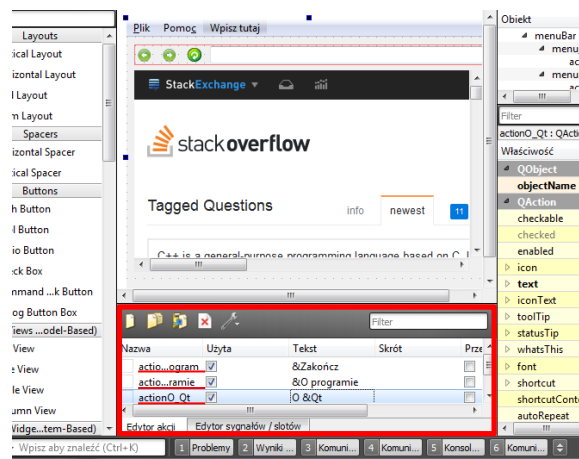


54. W analogiczny sposób dodaj do menu pozycję `&Zakończ` (pod pozycją `Plik`) oraz `Pomo&c` (jako nadrzędną) oraz podrzędne względem niej „`O programie`” i „`O Qt`”





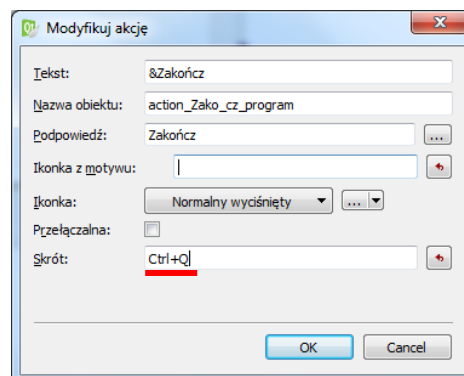
55. Zwróć uwagę na to, że w panelu „Edytor akcji” (na dole ekranu, w jego centralnej części) pojawiły się trzy nowe wpisy odpowiadające trzem nowym pozycjom w menu programu.



56. Z menu kontekstowego akcji Zakończ wybierz „Przejdź do slotu”, wybierz sygnał `triggered()` i w odpowiednim miejscu wpisz kod:

```
void MainWindow::on_action_Zako_cz_program_triggered()
{
    QApplication->quit();
}
```

57. Wróć do panelu edycji akcji, kliknij dwukrotnie we wpis odpowiadający akcji Zakończ i dodaj do niej skrót klawiaturowy, np. `ctrl-Q`



58. W podobny sposób obsłuż akcje „O programie” i „O Qt” (dodając klawisze skrótu `F1` i `ctrl-F1`):

```
#include <QMessageBox>
...
void MainWindow::on_actionOQt_triggered()
{
    QMessageBox::aboutQt(this);
}
...
void MainWindow::on_action_O_programie_triggered()
{
    QMessageBox::information(this, "Moja przeglądarka",
        "<html><span style='color:red;'>Piękny</span> program</html>");
}
```

59. Przetestuj program.



60. W wolnym czasie możesz załadować, skompilować i przetestować program przykładowy „Tab browser”, kompletną przeglądarkę WWW, która jest dystrybuowana wraz z QtCreatorem. Niestety, jest to zbyt skomplikowany program, by miało sens go teraz omawiać.

