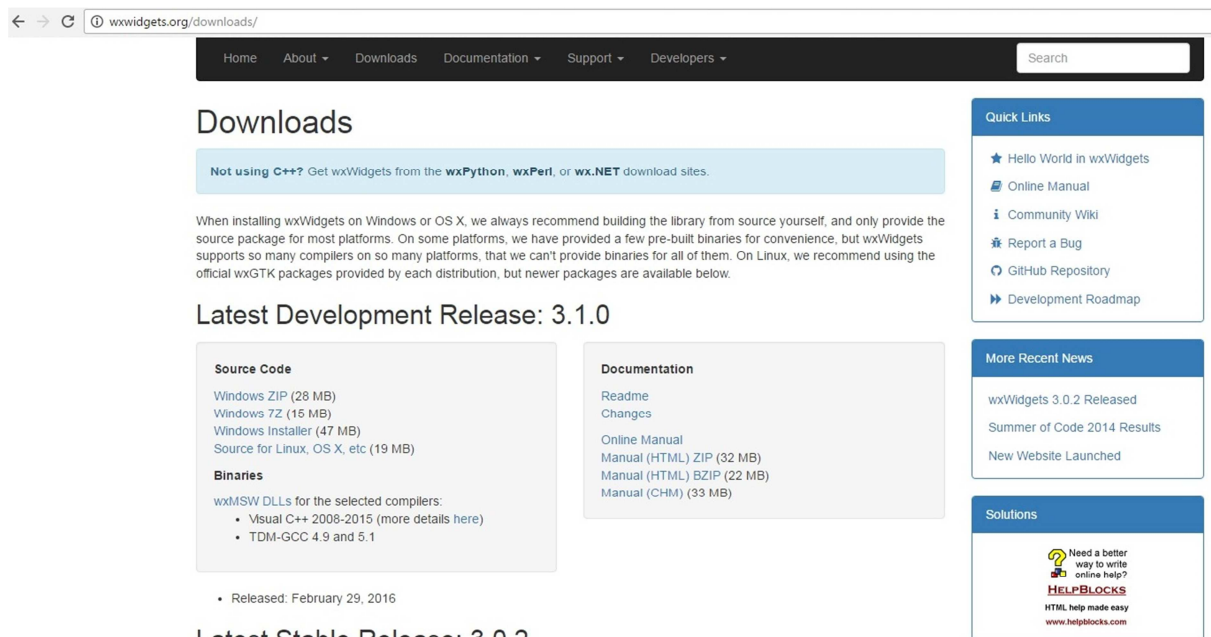


## **1. Wprowadzenie do aplikacji graficznych z wykorzystaniem WxWidgets**

*WxWidgets* stanowi bibliotekę klas *C++*, która umożliwia tworzenie aplikacji okienkowych zarówno w systemie *Windows*, *Linux*, jak i *MacOS*. Ponieważ biblioteka *wxWidgets* wykorzystuje elementy graficzne środowiska, w którym tworzony jest program, użytkownik może obserwować postępy działania aplikacji. Poza tym, niewątpliwą zaletą jest możliwość przeniesienia programu na inną platformę bez konieczności nadmiernych modyfikacji kodu. Ponieważ biblioteka na chwilę obecną jest rozpowszechniana na licencji zbliżonej do *LGPL*, można z niej bezpłatnie korzystać również do celów komercyjnych.

Z biblioteki *wxWidgets* można z powodzeniem korzystać w zintegrowanym środowisku programistycznym *Code Blocks*. Po pobraniu *IDE* ze strony: <http://www.codeblocks.org/downloads> (dla najnowszej wersji: <https://sourceforge.net/projects/codeblocks/files/Binaries/16.01/Windows/codeblocks-16.01mingw-setup.exe/download>) oraz rozpoczęciu procesu instalacji, program zostanie automatycznie zainstalowany w katalogu domyślnym: *C:\Program Files\CodeBlocks* (dla systemów 32-bitowych) lub *C:\Program Files (x86)\CodeBlocks* (dla systemu *Windows* 64-bitowego). Do zmiennej środowiskowej *Path* należy dołączyć ścieżkę dostępu do katalogu kompilatora *MingW*, która umożliwi korzystanie z narzędzi kompilatora *MingW* z poziomu wiersza poleceń. W tym celu należy kliknąć na: *Panel sterowania\System i zabezpieczenia\System* lub alternatywnie wybierając ikonę *Komputer*, kliknąć na prawy przycisk myszki, wybrać *Właściwości*, następnie *Zaawansowane Ustawienia Systemu*, kliknąć na przycisk *Zmienne Środowiskowe*, po czym wyszukać i wybrać do edycji zmienną o nazwie *Path*. W przypadku standardowej instalacji w systemie *Windows* 64-bitowym podana wyżej ścieżka dana jest następująco: *C:\Program Files (x86)\CodeBlocks\MinGW\bin*. Po jej dopisaniu oraz zatwierdzeniu operacji przyciskiem *OK*, należy zamknąć applet *System*.

W przypadku poprawnego zainstalowania środowiska *Code Blocks*, w następnej kolejności należy zainstalować bibliotekę *wxWidgets*. W tym celu należy pobrać pliki źródłowe biblioteki ze strony: <http://www.wxwidgets.org/downloads/> (Rys. 1)



Rys. 1. Różne wersje bibliotek *WxWidgets*

Po ukończeniu procesu pobierania wybranej biblioteki np. wersji *3.1.0.1* oraz jej instalacji, katalog *C:\wxWidgets-3.1.0* będzie zawierał niezbędne pliki źródłowe, które muszą zostać skompilowane. Poprzez uprzednie dodanie do zmiennej środowiskowej *Path* ścieżki dostępu do kompilatora *MingW*, czynność ta jest możliwa do realizacji z poziomu wiersza poleceń.

W tym celu, dla danej instalacji, należy w wierszu poleceń wprowadzić następujące dane:

1. `cd C:\wxWidgets-3.1.0\build\msw`
2. `mingw32-make -f makefile.gcc clean`
3. `mingw32-make -f makefile.gcc BUILD=debug SHARED=0 MONOLITHIC=1 UNICODE=1`
4. `mingw32-make -f makefile.gcc BUILD=release SHARED=0 MONOLITHIC=1 UNICODE=1`

Wybrano następujące, przykładowe opcje kompilacji:

*BUILD=release* – określono wersję biblioteki *release*, która nie posiada danych użytecznych podczas *debugowania* (dzięki temu czas kompilacji będzie krótszy);

*SHARED=0* – wybrano opcję dołączania plików bibliotecznych do programu wynikowego (tworzony będzie pojedynczy plik wykonawczy). Alternatywny wybór

wersji *SHARED=1* umożliwia tworzenie biblioteki dynamicznej *dll*, wykorzystywanej podczas działania programu. Wówczas program będzie krótszy lecz odbędzie się to kosztem konieczności dołączania do programu biblioteki;

*MONOLITHIC=1* – elementy będą dołączane tylko do jednego pliku bibliotecznego;

*UNICODE=1* – wybrano kodowanie znaków *Unicode*, stanowiące standard dla systemu operacyjnego *Windows*.

Należy zauważyć, że pełny proces kompilacji może trwać nawet kilkadziesiąt minut. Po jego zakończeniu, wszystkie pliki potrzebne do tworzenia aplikacji z wykorzystaniem biblioteki *wxWidgets* będą dostępne dla użytkownika środowiska *IDE*.

## 2. Struktura aplikacji

Każda aplikacja korzystająca z biblioteki *wxWidgets* składa się z kilku podstawowych elementów. Oprócz klasy aplikacji, są to: klasy okien (w tym okien dialogowych), klasy kontrolek i innych elementów aplikacji oraz tabele zdarzeń, których celem jest przyporządkowanie obsługi zdarzeniom funkcji. Pisząc program w pierwszej kolejności należy zadeklarować klasę reprezentującą aplikację, która dziedziczy po klasie *wxApp*, później zaś klasy okien oraz kontrolek. Program nie będzie zawierał funkcji *main()*, charakterystycznej dla programów napisanych w języku C++. Funkcja ta zawarta jest w makrze *IMPLEMENT\_APP*, w którym nazwa klasy aplikacji stanowi parametr makra. Podczas uruchamiania programu w pierwszej kolejności tworzony jest obiekt klasy aplikacji, potem wykonywana jest metoda *OnInit()* klasy aplikacji, którą przeciąża się poprzez wprowadzenie kodu utworzenia okna głównego oraz niezbędnych kontrolek.

Zgodnie z powyższym każda aplikacja wykorzystująca bibliotekę *wxWidgets* zawiera:

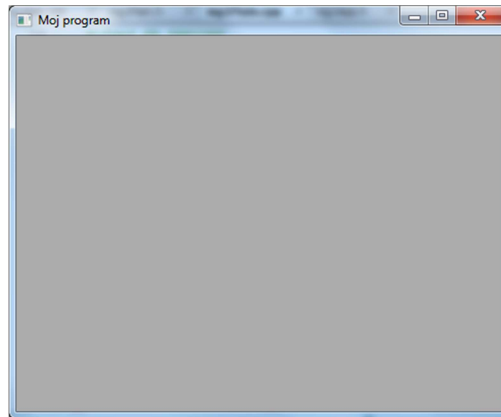
- dyrektywy dołączenia plików nagłówkowych;
- definicję klasy aplikacji, która dziedziczy po klasie *wxApp*;
- wywołanie makra *IMPLEMENT\_APP*;
- definicję przeciążonej metody *OnInit()*.

## 3. Zadania do wykonania

1. Zainstalować bibliotekę *wxWidgets* dla środowiska *Code Blocks*, po uprzednim sprawdzeniu, które wersje bibliotek obsługuje *IDE*, dołączyć ścieżkę dostępu do

katalogu kompilatora *MingW* oraz skompilować pliki źródłowe biblioteki z użyciem wiersza poleceń.

2. Przetestować poprawność realizacji zadania nr 1, tworząc nowy, pusty, domyślny projekt z wykorzystaniem biblioteki *wxWidgets*.
3. Napisać prosty program służący do wyświetlenia przykładowego okna, będącego obiektem klasy bibliotecznej *wxFrame*. Okno powinno posiadać lewy górny róg w punkcie ekranu o współrzędnych 50x50 oraz wymiary 200x200.



Wskazówka:

```
#include "wx/wxprec.h"

#ifdef WX_PRECOMP
#include "wx/wx.h"
#endif

class Aplikacja: public wxApp

{   public:

    virtual bool OnInit();};

IMPLEMENT_APP(Aplikacja);

bool Aplikacja::OnInit(){

wxFrame* MainWindow = new wxFrame(NULL,-1,_T("Moj
program"),wxPoint(50,50), wxSize(200,200),
wxDEFAULT_FRAME_STYLE); MainWindow->Show(TRUE); return TRUE;}
```