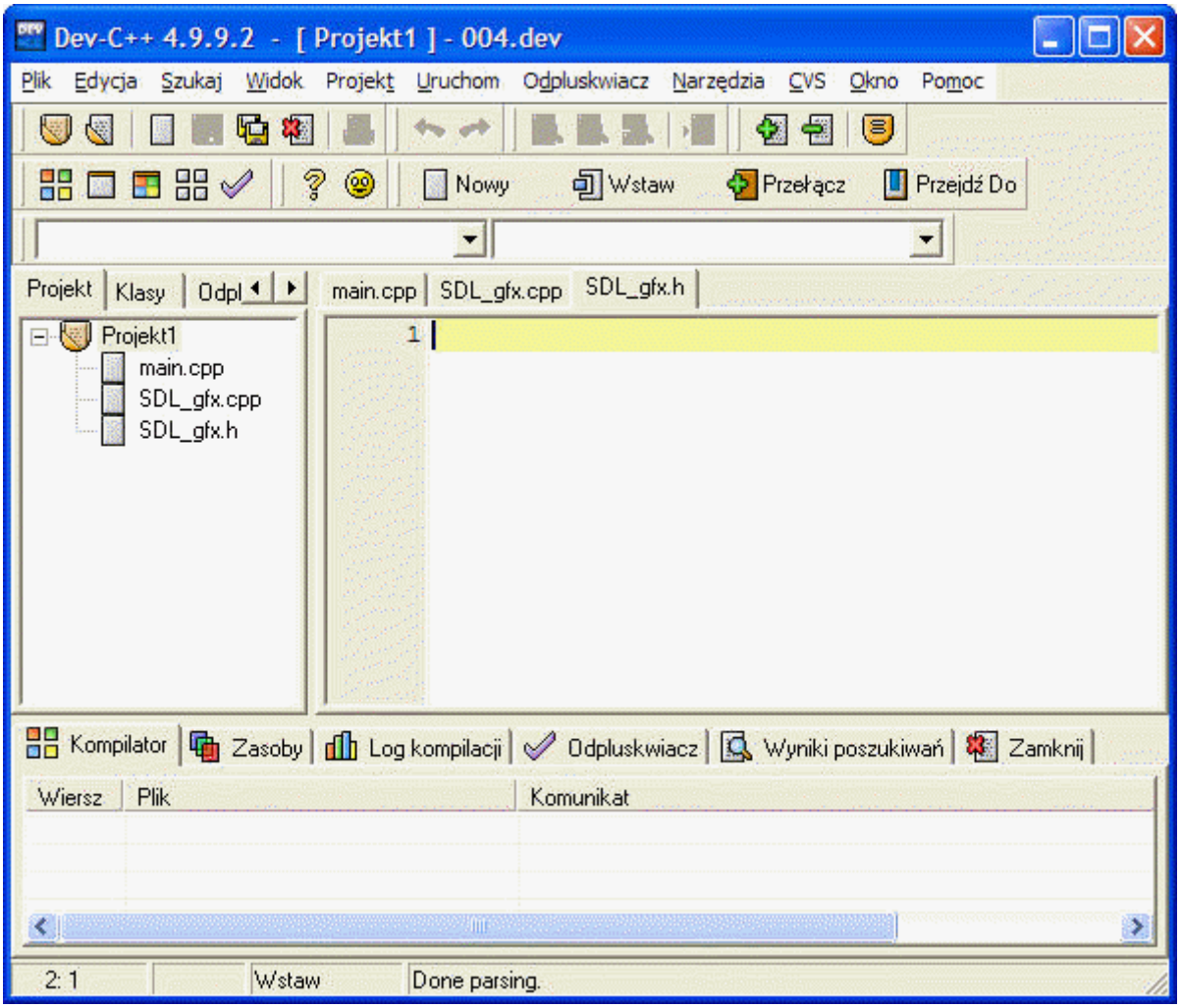
Przygotowanie plików dla własnej biblioteki graficznej w SDL

Biblioteka SDL udostępnia nam środowisko graficzne, lecz nie zawiera funkcji rysunkowych - musimy je stworzyć sami. Zatem na kolejnych zajęciach koła informatycznego będziemy zajmować się pisaniem podstawowych funkcji graficznych - tzw. prymitywów, które zbierzemy w pliku SDL_gfx.cpp. Plik ten będzie można dołączać do projektu tworzonej aplikacji, co udostępni nam wszystkie zawarte w nim funkcje. Dodatkowo utworzymy plik nagłówkowy z definicjami funkcji bibliotecznych o nazwie SDL_gfx.h. Oba pliki będziemy przechowywać w katalogu c:\Dev-Cpp\include\SDL.

Wykonaj następujące kroki:

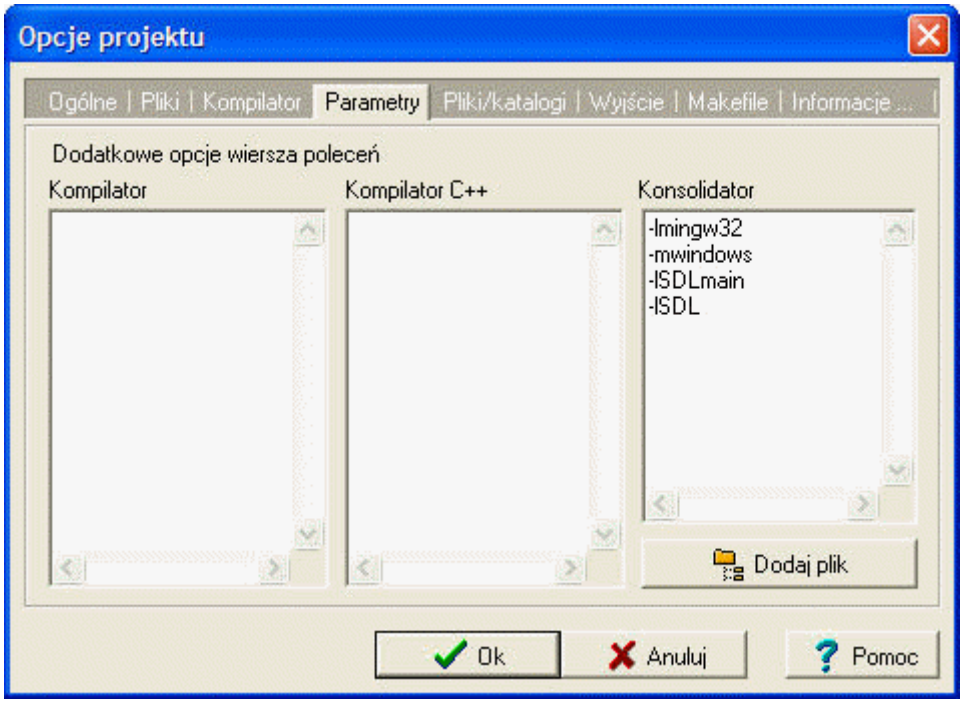
1. Uruchom środowisko Dev-C++. Utwórz nowy projekt Aplikacji Windows i zapisz go w odpowiednim katalogu na swoim dysku twardym. W edytorze zostanie umieszczony szablon aplikacji Windows. Ponieważ programując w SDL nie korzystamy z tego, możesz wszystko wykasować. Plik zapisz na dysku pod nazwą main.cpp w swoim katalogu - tzn. w tym katalogu, w którym będziesz tworzył swoją aplikację.
2. Wybierz z menu Projekt/Nowy plik. W oknie edytora powstanie nowa zakładka BezNazwy1. Zapisz ten plik w katalogu c:\Dev-Cpp\include\SDL pod nazwą SDL_gfx.cpp.
3. Ponownie wybierz opcję menu Projekt/Nowy plik. W oknie edytora pojawi się kolejna zakładka BezNazwy2. Plik również zapisz w katalogu c:\Dev-Cpp\include\SDL pod nazwą SDL_gfx.h. Z obu plików będziemy korzystać w późniejszych programach.

Jeśli poprawnie wykonałeś opisane dotąd kroki, to okno programu powinno wyglądać tak:



4. Wybieramy opcje menu **Projekt/Opcje** projektu, w okienku dialogowym klikamy w zakładkę Parametry i do pola tekstowego Konsolidator wprowadzamy poniższy wpis:

-lmingw32
-mwindows
-lSDLmain
-lSDL



5. Teraz w okienku edytora kliknij zakładkę z nazwą **main.cpp** (możesz również kliknąć nazwę pliku main.cpp w okienku Projekt, które znajduje się po lewej stronie okna edytora - różnica jest taka, iż zakładki istnieją tylko wtedy, gdy dane pliki zostały wczytane do edytora, w w oknie projektu widzisz WSZYSTKIE pliki dołączone do swojego projektu). Przełączysz się na główny plik źródłowy naszej aplikacji. Przenieś do tego pliku poniższy szablon aplikacji SDL (zwróć uwagę, iż dołączamy plik nagłówkowy SDL_gfx.h, w którym będą prototypy tworzonych przez nas funkcji - plik ten zawiera już dołączenie SDL.h, więc nie musimy tego robić w naszym programie):

```
//
// P004 - test blików biblioteki
SDL_gfx
//-------------------------------------
-

#include <windows.h>
#include <SDL/SDL_gfx.h>

int main(int argc, char *argv[])
{
  MessageBox(0, gfxLibVersion(), "OK,
koniec", MB_OK);
  return 0;
}
```

6. Kliknij w zakładkę SDL_gfx.cpp (lub w nazwę pliku w okienku Projekt). Następnie przekopiuj do edytora poniższy tekst programu:

```
//
// Biblioteka procedur graficznych dla
trybów 32 bitowych
//-------------------------------------
-----------------

#include <SDL/SDL_gfx.h>

//------------------
// DEFINICJE FUNKCJI
//------------------

// Zwraca tekst z numerem wersji
biblioteki
//-------------------------------------
----

char * gfxLibVersion(void)
{
  return "1.01\0";
}
```

W bibliotece jest na razie tylko jedna funkcja - gfxLibVersion(), która zwraca tekst z opisem numeru wersji biblioteki. Funkcję tę wykorzystuje program główny do wyświetlenia numeru wersji w końcowym oknie wiadomości. Zapisz program na dysku (Ctrl+S).

7. Kliknij w zakładkę SDL_gfx.h. Przekopiuj do okna edytora poniższy tekst:

```
//
// Biblioteka procedur graficznych dla
trybów 32 bitowych
//-------------------------------------
-----------------

#include <SDL/SDL.h>

char * gfxLibVersion(void);
```

Jest to plik nagłówkowy (h = heading, czyli nagłówek). Plik definiuje tzw. prototypy funkcji, dzięki którym kompilator będzie mógł sprawdzać poprawność użycia funkcji w

programie. Prototyp funkcji gfxLibVersion() określa, iż zwraca ona wskaźnik do tekstu - ciągu znaków. Ciąg znaków zgodnie z konwencjami w C++ musi kończyć się kodem zero ("\0"). Tak właśnie jest - spójrz do poprzedniego pliku SDL_gfx.cpp.

Pliki nagłówkowe dołączamy do tekstów źródłowych tworzonych programów za pomocą dyrektywy preprocesora #include.

8. Teraz sprawdzimy poprawność konfiguracji naszej aplikacji. Wróć na zakładkę main.cpp i skompiluj swój program. Jeśli będą błędy, to sprawdź, czy postępowałeś zgodnie z opisanymi krokami. Jeśli nie będzie błędów, uruchom program. Jako wynik powinieneś otrzymać poniższe okienko wiadomości z numerem wersji biblioteki: