

RAD w aplikacjach mobilnych i enterprise

C++ Builder – wprowadzenie

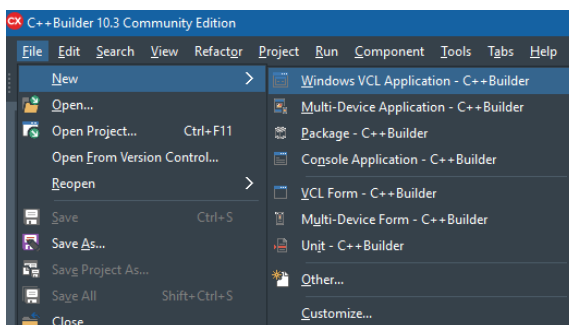
opracował: dr inż. Andrzej Czajkowski, a.czajkowski@issi.uz.zgora.pl

1 Cel ćwiczenia

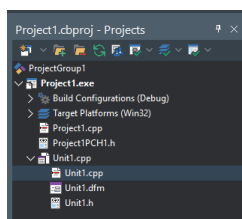
Celem ćwiczenia jest zapoznanie się z podstawową funkcjonalnością środowiska C++ Builder – aplikacje VCL (Visual Component Library).

2 Przebieg zajęć – pierwsza aplikacja.

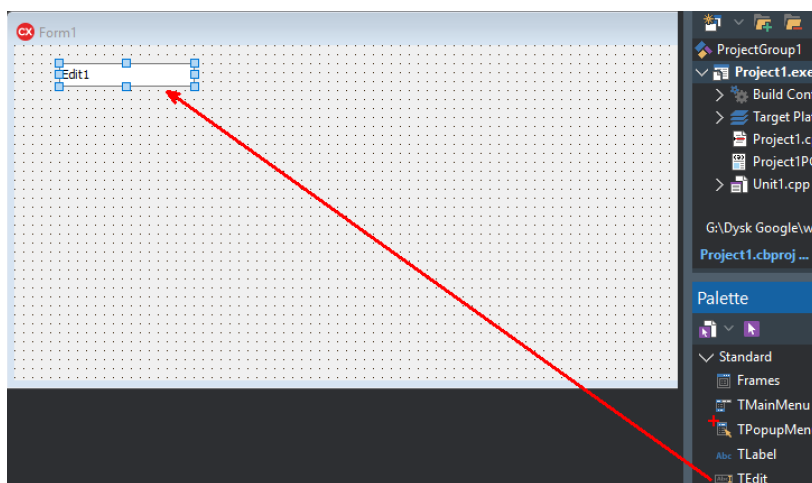
1. Po uruchomieniu środowiska utworzyć nowy projekt jako Windows VCL Application - C++ Builder i zapisać go w wybranej lokalizacji (Save all).



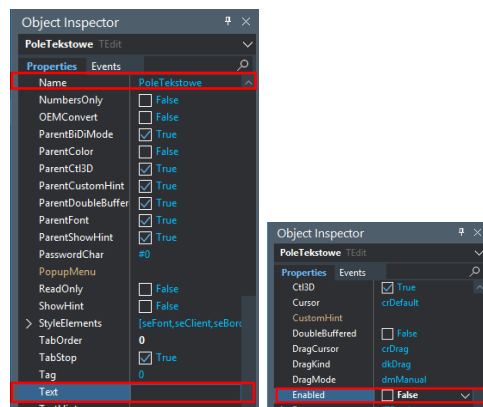
2. Przeanalizuj okno plików projektu, zauważ, że po rozwinięciu pliku *.cpp, podwójne kliknięcia na plikach *.h, *.dfm i *.cpp umożliwia edycję różnych elementów aplikacji:



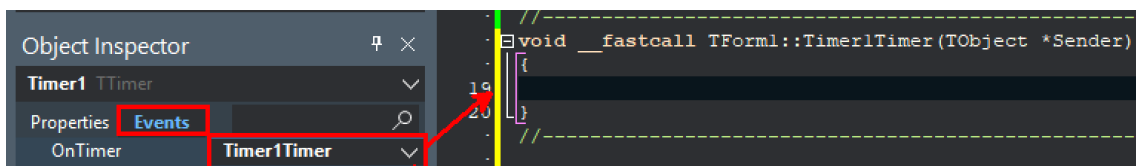
3. Do formularza dodaj TEdit z palety komponentów:



4. Zmień nazwę komponentu, wyczyść wyświetlany domyślny napis oraz zablokuj edycję w Object Inspectorze (pola Name, Enabled, Text):



5. !!! Uwaga !!! powyższa zmiana nazwy komponentu (jak i używanych w późniejszym czasie metod) musi być realizowana w inspektorze a nie w pliku nagłówkowym. Zmiany w pliku nagłówkowym nie są refaktoryzowane w całym projekcie.
6. Dodaj do formularza w dowolnym miejscu nierenderowany komponent TTimer.
7. Upewnij się, że pole **Interwał** w inspektorze jest ustawione na 1000 (wartość w milisekundach dzięki której cyklicznie jest wywoływane zdarzenie OnTimer).
8. W inspektorze obiektów przełącz się na zakładkę **Events** i kliknij dwukrotnie zdarzenie **OnTimer** co pozwoli na jego edycję:



9. Zmodyfikuj kod aby licznik co 1 sekundę aktualizował stan an polu TEdit.

```
void __fastcall TForm1::Timer1Timer(TObject *Sender)
{
    static unsigned long count=0;

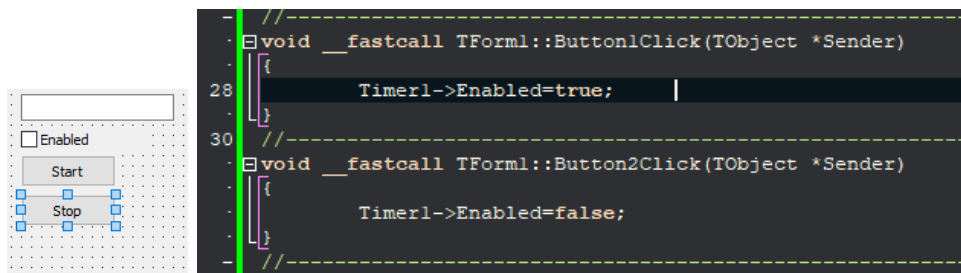
    PoleTekstowe->Text=count++;
}
```

10. Skompiluj i uruchom aplikację skrótem F9. W polu tekstowym, co sekundę, powinna się pojawiać inkrementowana wartość zmiennej `count`.
11. Kolejno dodaj do formularza komponent TCheckBox.
12. Zmień wyświetlany napis (pole `Caption`) na "Enabled".
13. Zmodyfikuj kod metody `Timer1Timer` aby uwzględniał aktualizowanie pola tekstonego dodanym checkbox'em:

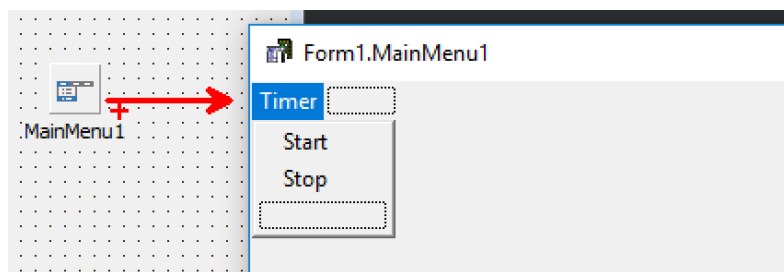
```
if (CheckBox1->Checked) {
    PoleTekstowe->Text=count++;
}
```

14. Dodaj do formularza dwa przyciski TButton i zmień ich napisy na start i stop.

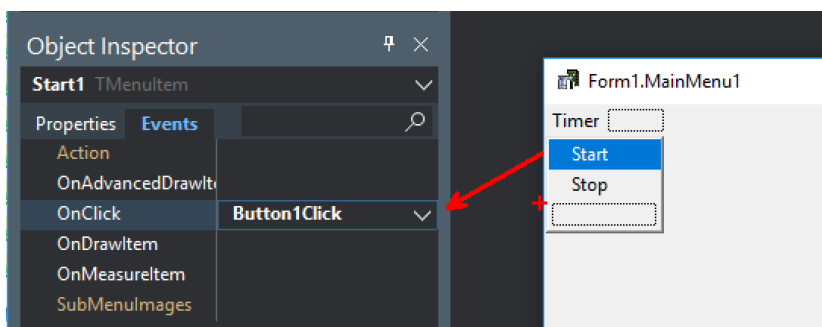
15. Dla obu przycisków dodaj metody obsługujące zdarzenie `OnClick` poprzez kliknięcie ich bezpośrednio na formularzu (w przypadku przycisków metoda ta jest szybsza niż dodawanie przez zakładkę `Events` w inspektorze). Zmodyfikuj kod aby przyciskowi uruchamiały i zatrzymywały licznik czasu:



16. Kolejno dodaj do formularza w dowolnym miejscu komponent `TMainMenu` który pełni funkcję paska menu dla tworzonego okna aplikacji. Po jego dwukrotnym kliknięciu możliwe jest zaprojektowanie menu (komponenty typu `TMenuItem` tworzone są automatycznie):

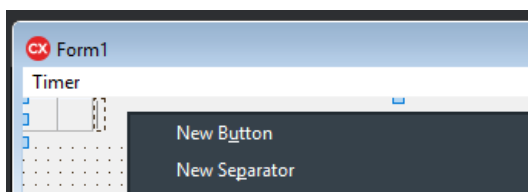


17. Podwójne kliknięcie komponentów `TMenuItem` umożliwia automatyczne tworzenie kodu uruchamianego po ich kliknięciu w aplikacji. Jednak aby niedublować funkcji uruchamiania jak zatrzymywania licznika możliwe jest odwołanie się do już utworzonych metod dla przycisków:



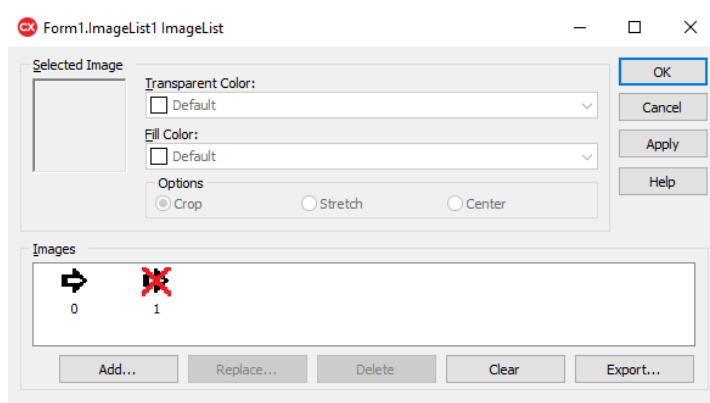
analogicznie utworzyć powiązanie dla opcji menu stop.

18. Kolejny dodawany komponent to `TToolBar`. Po jego dodaniu możliwe jest za pomocą prawego przycisku myszki dodawać do niego kolejne komponenty. Dodaj dwa przyciski na pasku i zakończ je separatorem:

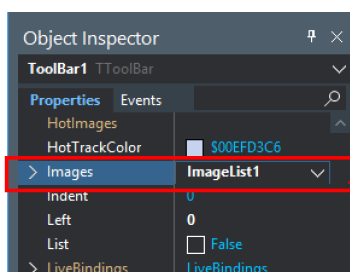


19. Dodane przycisku powiąż z metodami uruchamiającymi licznik.

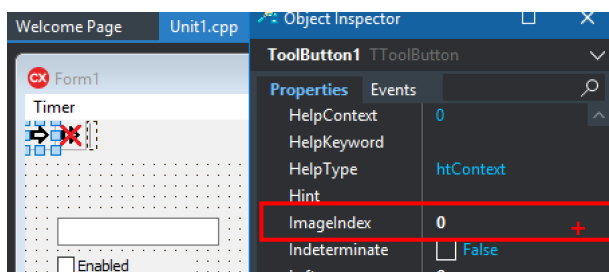
20. W celu zmiany ikon na tych przyciskach należy utworzyć `TImageList` i po jego dwukrotnym kliknięciu można dodać do listy pliki graficzne (należy je utworzyć w dowolnym programie graficznym):



21. Następnie należy dodać `ImageList` do pola `Image` paska narzędzi.



22. Ikonki powinny pojawić się na przyciskach automatycznie, ich kolejność można zmieniać z wykorzystaniem pola `ImageIndex`:



3 Zadania do samodzielnego wykonania

1. (obowiązkowe do zaliczenia) Wykonaj aplikację kalkulatora z podstawową funkcjonalnością (resetowanie, dodawanie, odejmowanie, dzielenie i mnożenie) operującego na liczbach całkowitych i zmiennoprzecinkowych. Zaimplementuj podstawową obsługę błędów (dzielenie przez zero, przekroczenie dostępnego zakresu na wyświetlaczu). Możliwość wpisywania cyfr i działań poprzez przyciski i klawiaturę. (Wykorzystać komponenty `TButton`, `TEdit`).
2. (zadanie dodatkowe) Zaimplementuj edytor tekstu z możliwością scrollowania, prostego formatowania (styl i rodzaj czcionki) i możliwością zapisu/odczytu do/z pliku.