Politechnika Poznańska
Wydział Budowy Maszyn i Zarządzania

Programowanie systemów informatycznych – laboratorium

Ćw. 1: Programowanie obiektowe

1) Cel ćwiczenia

Celem ćwiczenia jest zapoznanie się z podstawowymi elementami obiektowymi systemu Windows wykorzystując Visual Studio 2008 takimi jak: przyciski, pola tekstowe, okna pobierania danych itd.

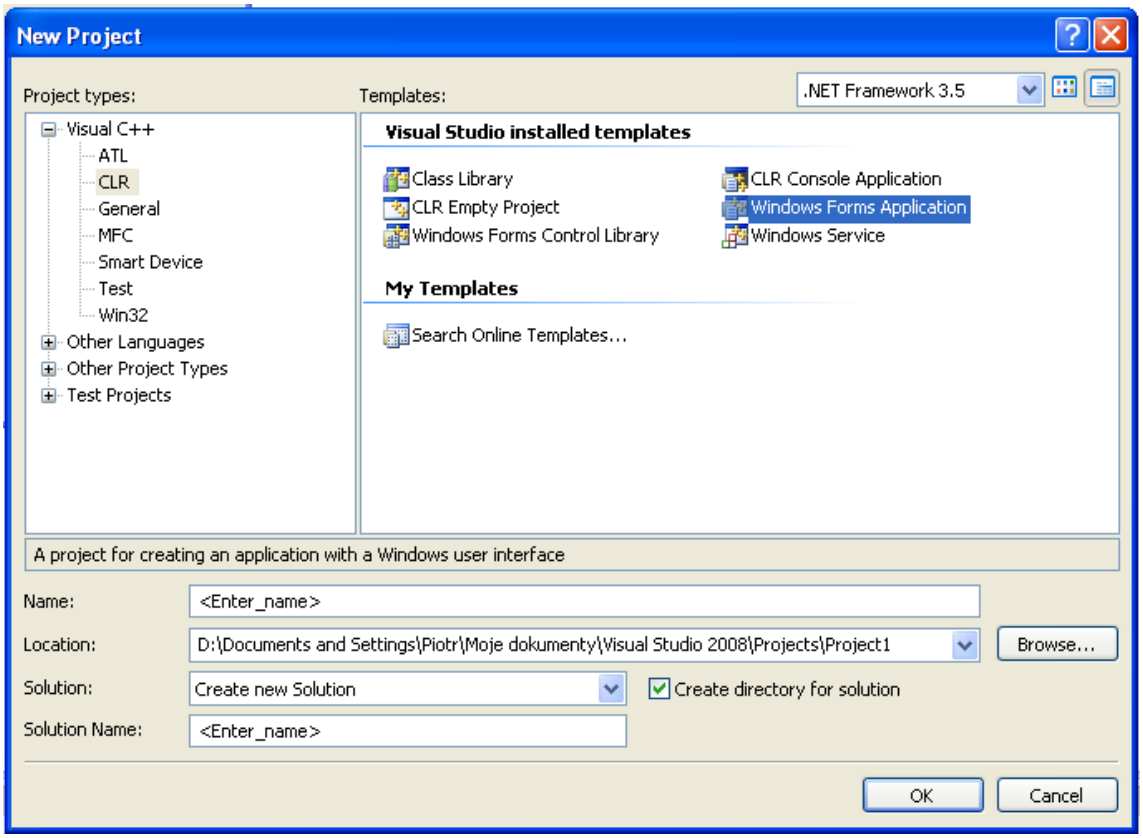
2) Zakres niezbędnych wiadomości

- podstawy języka C
  - konstruowanie funkcji w C
  - rodzaje zmiennych w C
  - struktury
  - klasy
  - wskaźniki

3) Wstęp

Aplikacja tworzona na zajęciach wykorzystuje środowisko programistyczne Visual Studio 2008 w języku C++. Tworząc projekt wybieramy kolejno:

- Visual C++
- CLR
- Windows Form Application

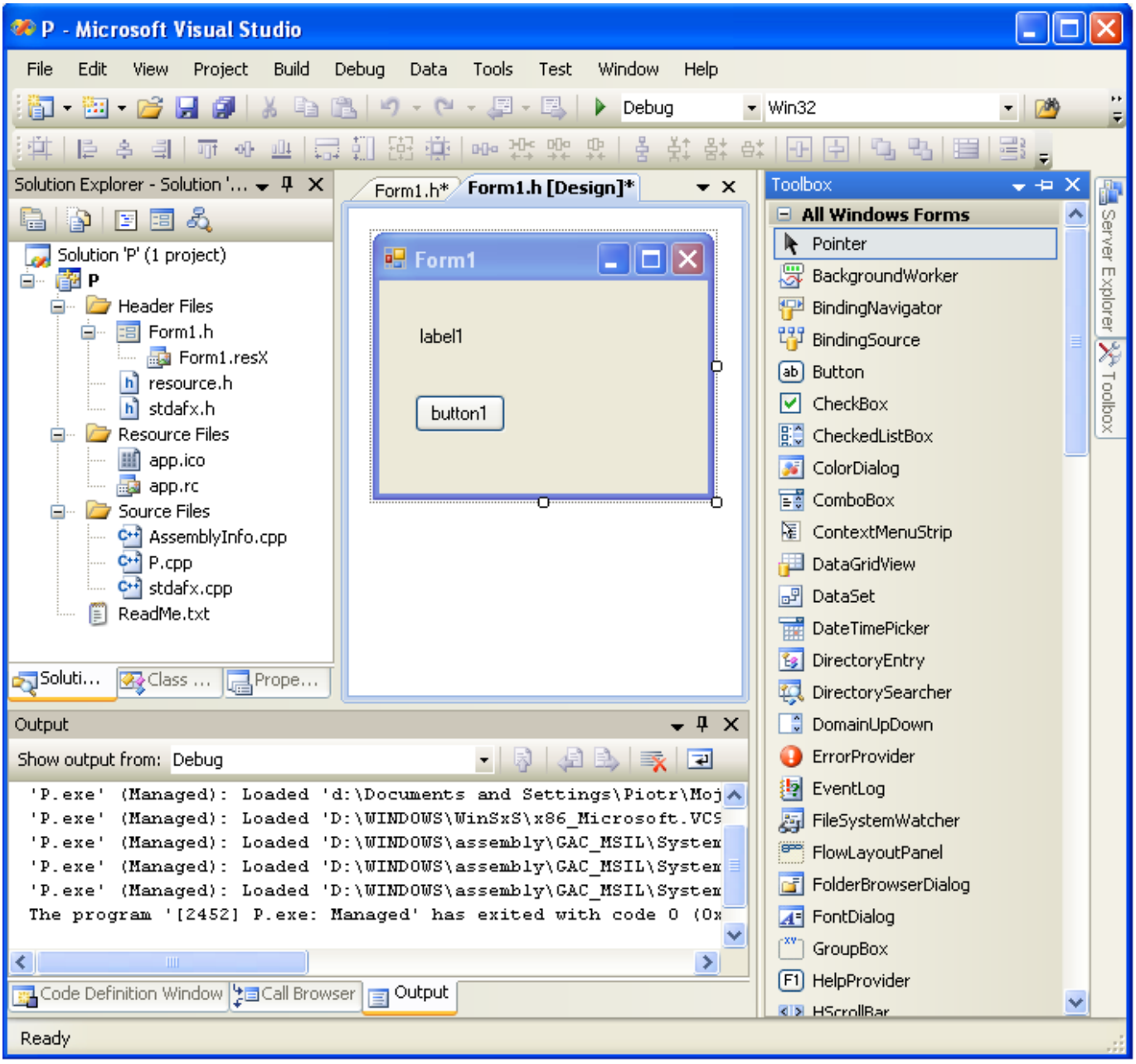


Po utworzeniu projektu zostanie wyświetlone puste okno o nazwie Form1. Tak utworzony projekt jest w pełni działającą aplikacją, która zostanie wykorzystana do dalszej pracy. Aplikację można uruchomić na dwa sposoby:

1. Build -> Build Solution (F7) -> Debug -> Start Debugging.
2. Po przebudowaniu aplikacji zostanie utworzony plik *.exe w folderze z projektem.

-label, button

System Windows operuje na obiektach, są to klasy. Pierwszym ważnym obiektem jest „Label”. Jest to pole teksowe służące do opisywania informacji znajdujących się w oknie oraz do wyprowadzania wyników. Wszystkie obiekty wykorzystywane w aplikacji można znaleść w zakładcę „Toolbox”.



Na oknie można umieszczać wszystkie dostępne kontrolki. Parametry obiektu zmienia się w opcji „Properties” dostępnej po zaznaczeniu obiektu. Do podstawowych parametrów każdego obiektu należy: rozmiar czcionki, rozmiar obiektu, nazwa obiektu oraz aktywność.
Po dwukrotnym kliknięciu na obiekt automatycznie klasa okna zostaje uaktualniona o funkcję obsługującą daną kontrolkę.

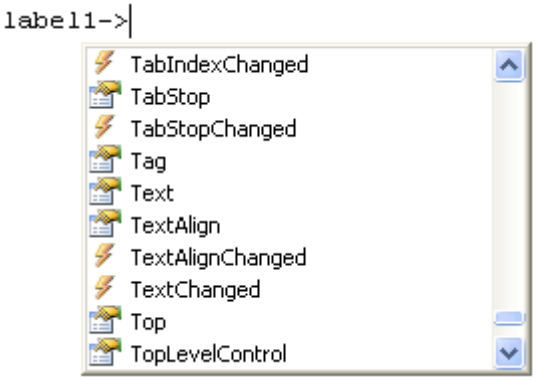
```
private: System::Void button1_Click(System::Object^  sender, System::EventArgs^  e) {

//****************************//
                }
```

Wewnątrz funkcji można umieszczać podstawowe operacje arytmetyczne dostępne w C++ oraz co najważniejsze można obsługiwać pracę głównego okna i obiektów na nim się znajdujących. Prosty przykład przedstawiono poniżej.

Zad1. Po kliknięciu na przycisk „button1” należy zmienić wartość pola „Text” obiektu „label1”.

```
label1->Text="Zad1";
```

Po wpisaniu komendy odwołania się do obiektu „label1” (pamiętając, że aplikacja operuje na wskaźnikach) zobaczymy zmianę napisu. Używając operatora „->” odwołujemy się do pól danej klasy. Pomocą jest wyświetlana lista z możliwościami wyboru elementów danej klasy.



Wyświetlenie napisu jest trywialnie proste, sytuacja się komplikuje gdy użytkownik chce wyświetlić zmienną. Należy wtedy wykonać konwersję, do tego celu należy wykonać poniższą funkcję:

```
int k =100;
label1->Text=String::Format("Wartość int: {0:n2}", k);
```

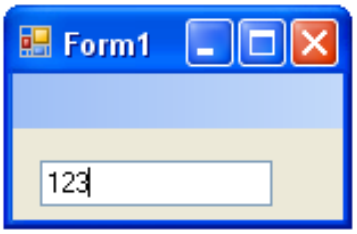
Przykład:

```
// przybliżenia
String::Format("{0:0.00}", 123.4567);       // "123.46"
String::Format("{0:0.00}", 123.4);          // "123.40"
String::Format("{0:0.00}", 123.0);          // "123.00"

// przybliżenie z zaokrągleniem
String::Format("{0:00.0}", 123.4567);       // "123.5"
String::Format("{0:00.0}", 23.4567);        // "23.5"
String::Format("{0:00.0}", 3.4567);         // "03.5"
String::Format("{0:00.0}", -3.4567);        // "-03.5"
```

```
// przybliżenie do cyfry znaczącej
String::Format("{0:0.##}", 123.4567);       // "123.46"
String::Format("{0:0.##}", 123.4);          // "123.4"
String::Format("{0:0.##}", 123.0);          // "123"
```

-TextBox

Pole tekstowe służy do wyświetlania oraz wprowadzania informacji.



Należy pamiętać, że pola tekstowe są to ciągi znaków, dlatego chcą wprowadzać parametry liczbowe należy taki ciąg konwertować do odpowiedniego typu liczbowego.

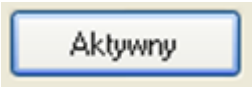
```
zmienna = Double::Parse (textBox1->Text);
```

Przykład:

```
label1->Text="123";
int zmienna= System::Int16::Parse(label1->Text);
label1->Text=""; // "wyzerowanie pola tekstowego"
label1->Text=String::Format("{0:0.00}", zmienna); // "wyświetlenie danych na
kontrolce"
```

Gdy chcemy wyświetlić pewien ciąg znaków lub licz w polu „TextBox” należy odnieść się do niego jak w przypadku obiektu „Label”. Dodawanie ciagów znaków następuje poprzez operator „+”.

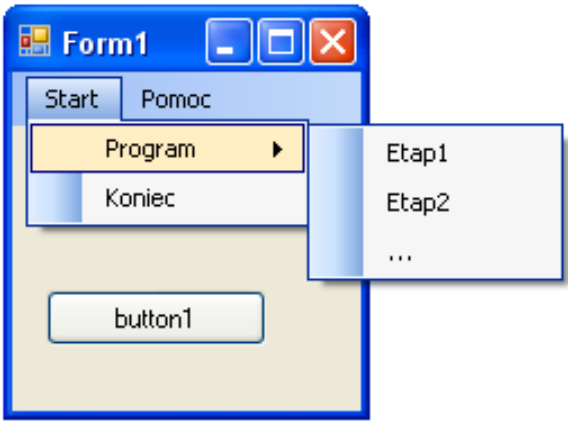
Aktywacja/dezaktywacja kontrolki:



button1->Enable = true; **button2->Enable = false;**

-MenuStrip1

Jest to klasyczne menu wykorzystywane w podstawowych aplikacjach obiektowych. Przykładowy wygląd przedstawiono poniżej:



Aby odnieść się do danego elementu „menu” należy na niego dwa razy kliknąć myszą w celu dodania funkcji obsługującej dany element. Analogicznie jak w przypadku obiektu „button”. Przykładowa funkcja ponież:

```
private: System::Void programToolStripMenuItem_Click(System::Object^  sender,
System::EventArgs^  e) {
   textBox1->Text="etap1";}
```

-Zmienne globalne

Globalne zmienne umieszczamy nad funkcjami obsługującymi dane zdarzenia po dyrektywie „`#pragma endregion”` , lecz przed funkcjami obsługi okna głównego. Z takich zmiennych można korzystać w wielu miejscach w programie zachowując ich przypisane wartości w przeciwieństwie do zmiennych lokalnych które usuwane są po zakończeniu funkcji. Przykład poniżej:

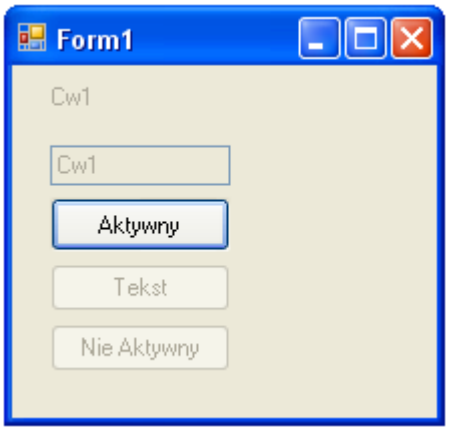
```
#pragma endregion

private : int liczba1;  //zmienna globalna
private : int liczba2;  //zmienna globalna


private: System::Void button1_Click(System::Object^  sender, System::EventArgs^  e)
{
      int liczba3;      //zmienna lokalna
}
```

4) Zadania.

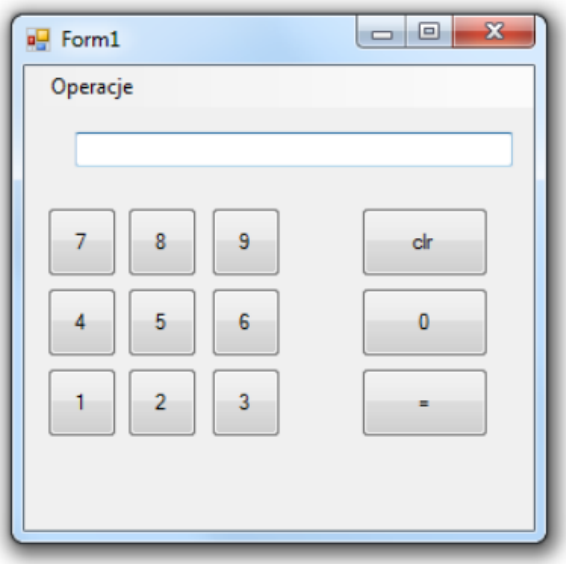
Ćw. 1 Wykorzystując obiekty typu „button”, „TextBox” oraz „Label” utworzyć aplikację, która ma za zadanie uaktywnić okno oraz przyciski umożliwiające wpisanie danych:

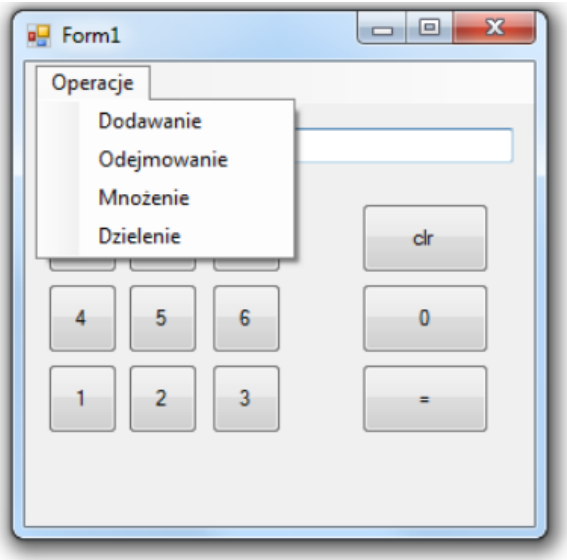
-„Aktywny” – przycisk, który uaktywni pozostałe dwa oraz pola tekstowe, a sam ma zostać zmieniony na nie aktywny

-„Text” – przycisk, który ma pobrać informacje z obiektu „TextBox” oraz wyświetlić je na obiekcie „Label”
-„Nie aktywny” – przycisk, który ma ponownie wyłączyć wszystkie kontrolki, a kontrolkę „Aktywny” ustawić na aktywną.



Ćw. 2 Wykorzystując wiedzę poznaną na laboratoriach utworzyć aplikację kalkulatora z wykorzystaniem obiektu „MenuStrip”. Należy utworzyć:
-podstawowe przyciski numeryczne,
-operację dodawania odejmowania oraz mnożenia i dzielenia mają być wywoływane z menu,
-Wykorzystać jedno pole tekstowe do pobierania i wyświetlania informacji.





Wygląd programu dowolny z zachowanie wymienionych obiektów oraz poprawności działania.

5) Sprawozdanie

Politechnika Poznańska
Wydział Budowy Maszyn i Zarządzania

Programowanie systemów informatycznych – laboratorium

Ćw. 1: Programowanie obiektowe

Sprawozdanie powinno zawierać:
- fragmenty funkcji wykonanych przykładów ze wstępu do ćwiczenia
- program, jaki grupa napisała na zajęciach
- za każdą linijką programu, jeżeli będzie taka potrzeba, należy umieścić bieżący komentarz na temat danej linii kodu
- pod każdym programem należy umieścić własny opis działania programu i wnioski.
- ogólne wnioski i spostrzeżenia na temat wykonanego ćwiczenia i zdobytej wiedzy