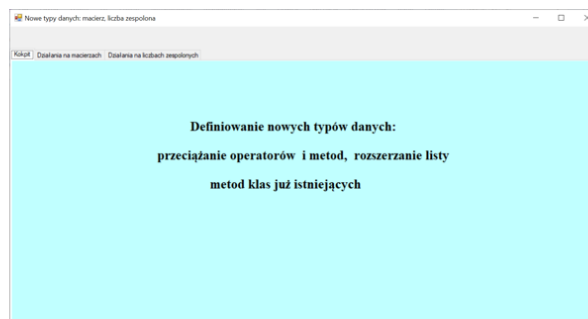


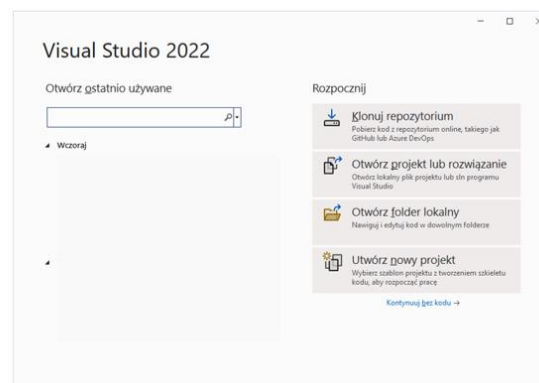
PROJEKT Nr 1

Zgodnie z treścią zadania projektowego Nr 1 mamy zaprojektować program formularzowy umożliwiający wykonywanie działań (kalkulatora) na nowo zdefiniowanych typach danych:

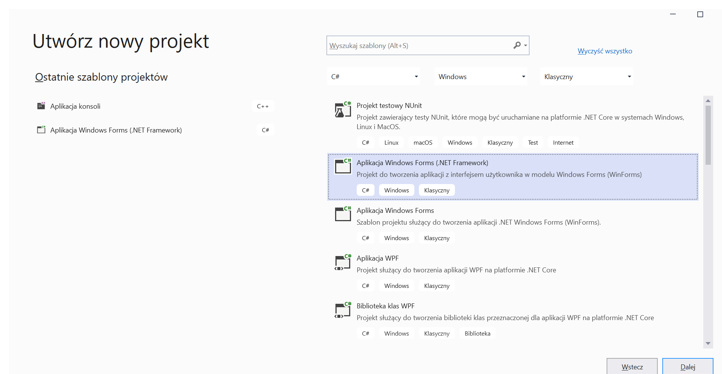
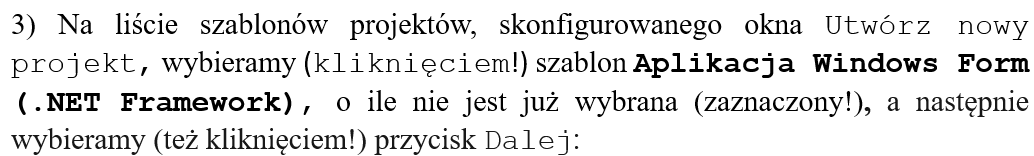


Otwarcie Projektu Nr 1

- 1) Uruchamiamy Visual Studio 2022 i wybieramy przycisk Utwórz nowy projekt:

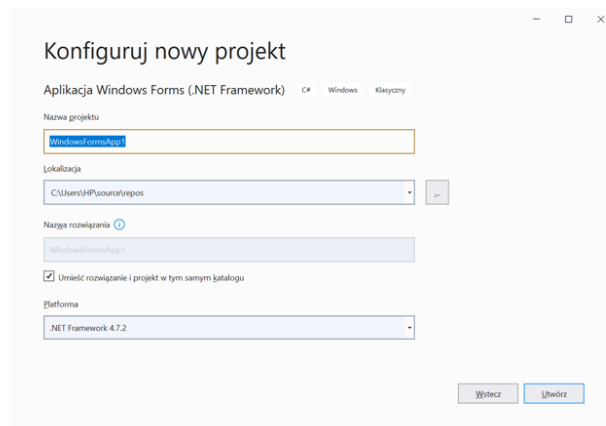


2) W wyświetlonym, nowym oknie Utwórz nowy projekt, w sekcji filtrowania **ustawiamy** rodzaj projektowanego programu na Klasyczny (Desktopowy) oraz język programowania C# i system operacyjny Windows:



Opracował: L. Jung

4) W nowym oknie Konfiguruj nowy projekt:

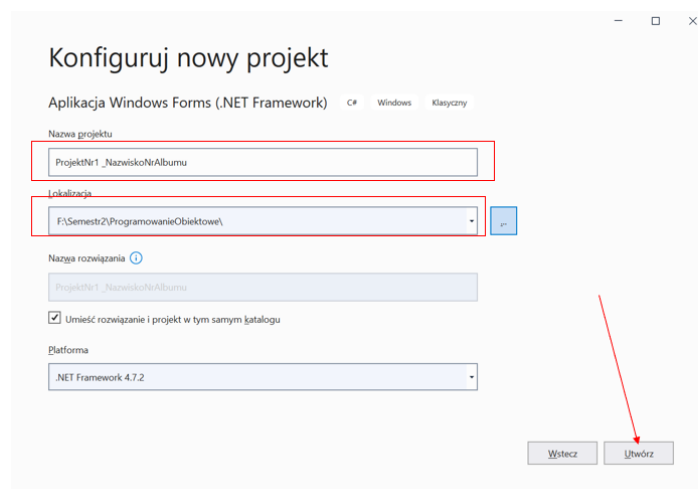


5) W nowym oknie Konfiguruj nowy projekt:

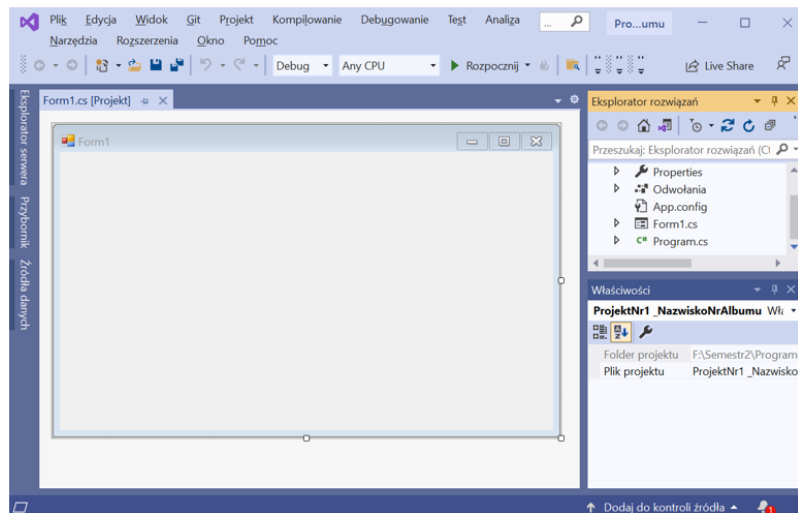
Wypełniamy tylko dwa pola i tak w polu:

- Nazwa projektu, wpisujemy nazwę projektowanego programu: ProjektNr1_NazwiskoNrAlbumu,
- Lokalizacja, wybieramy folder, w którym ma być zapisany projekt programu (dla przykładu): E:\Semestr2\ProgramowanieObiektowe,

Po wypełnieniu tych pól klikamy przycisk: Utwórz!



Przechodzimy do okna projektowego programu formularzowego ProjektNr1:



Formatowanie formularza

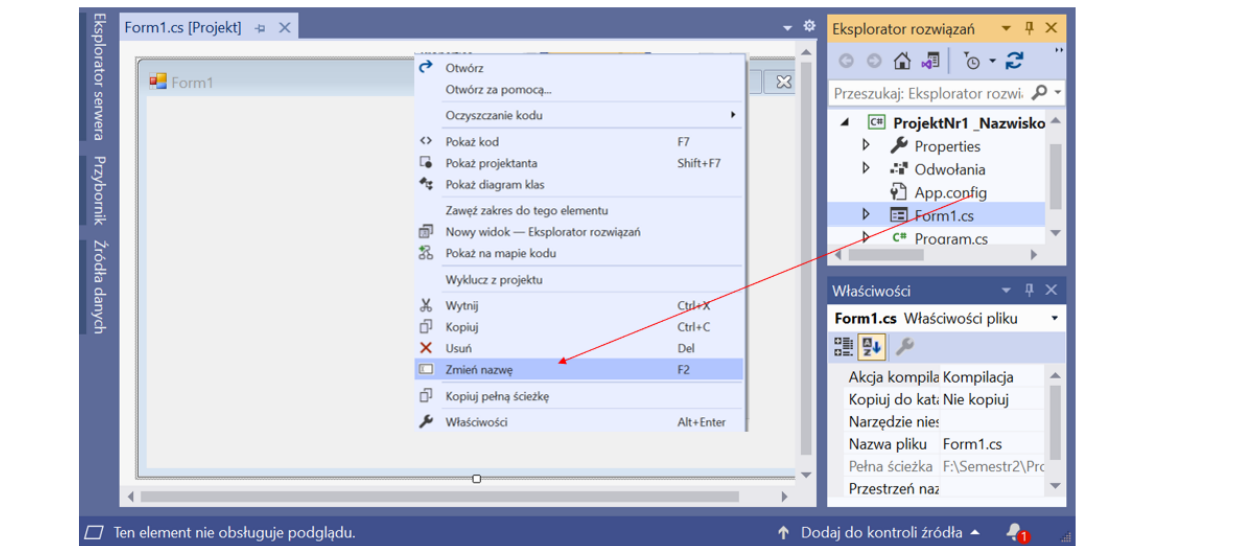
Formatowania (ustalanie właściwości, czyli atrybutów) formularza rozpoczniemy od ustalenia nazwy i tytułu formularza oraz jego rozmiarów.

Zmiana nazwy formularza

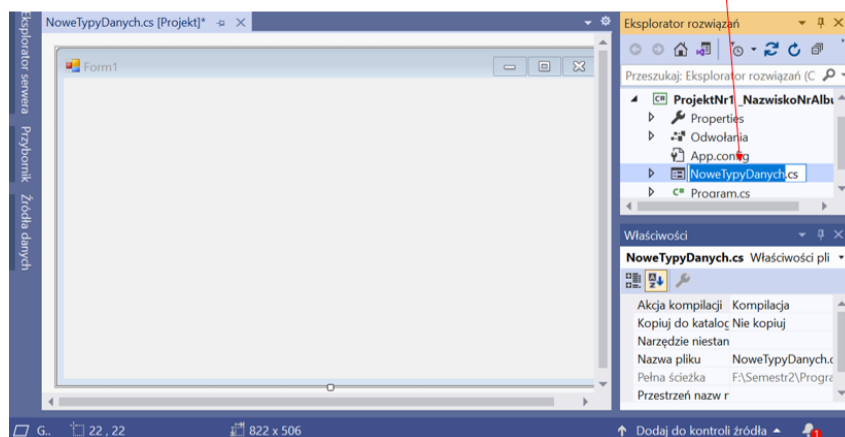
Wyświetlony w oknie projektowym Visual Studio 2022 formularz ma domyślną nazwę Form1, którą powinniśmy zmienić na własną (mnemoniczną) odpowiadającą projektowanemu programowi:

- w oknie Solution Explorer otwartego projektu (w naszym przypadku jest to projekt o nazwie ProjektNr1_NazwiskoNrAlbumu), klikamy prawym przyciskiem myszy nazwę formularza Form1 i z podmenu wybieramy polecenie Zmień nazwę (Rename):

Opracował: L. Jung

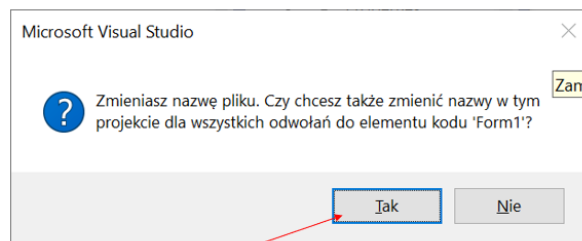


teraz w miejsce zaznaczonej nazwy `Form1` (rozszerzenia `.cs` nie zmieniamy!) wpisujemy nową nazwę formularza, którą będzie nazwa `NoweTypyDanych`, po czym naciskamy klawisz `Enter`:



Opracował: L. Jung

po naciśnięciu klawisza Enter lub kliknięciu w dowolne miejsce formularza `Form1`, zostanie wyświetlone okno dialogowe z zapytaniem czy mają być zmienione nazwy wszystkich plików umieszczonych w folderze `Form1`, którego nazwę teraz zmieniamy:



kliknięciem wybieramy przycisk **Tak**!

Zmiana tytułu formularza

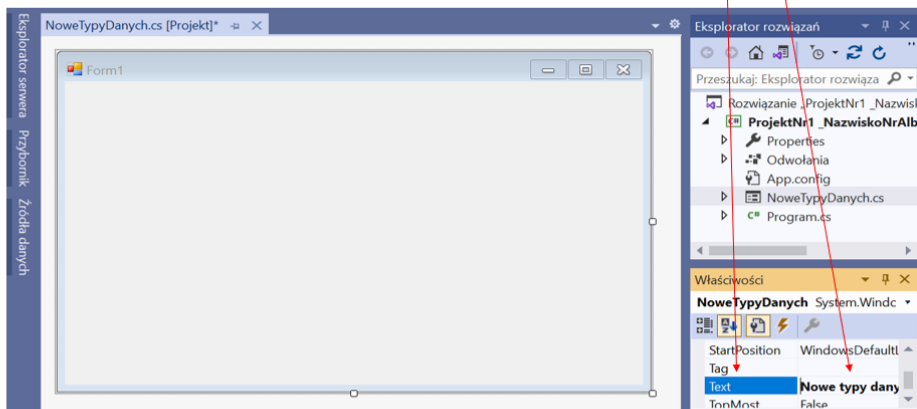
Po otwarciu nowego projektu w Visual Studio 2022 formularz ma domyślny tytuł `Form1`, którą też powinniśmy zmienić na tytuł odpowiadający (czytelniejszy dla Użytkownika programu) merytorycznie (funkcjonalnie) projektowanemu programowi:

- kliknięciem zaznaczamy formularz (powinny pojawić się uchwyty na dolnej i prawej krawędzi formularza):

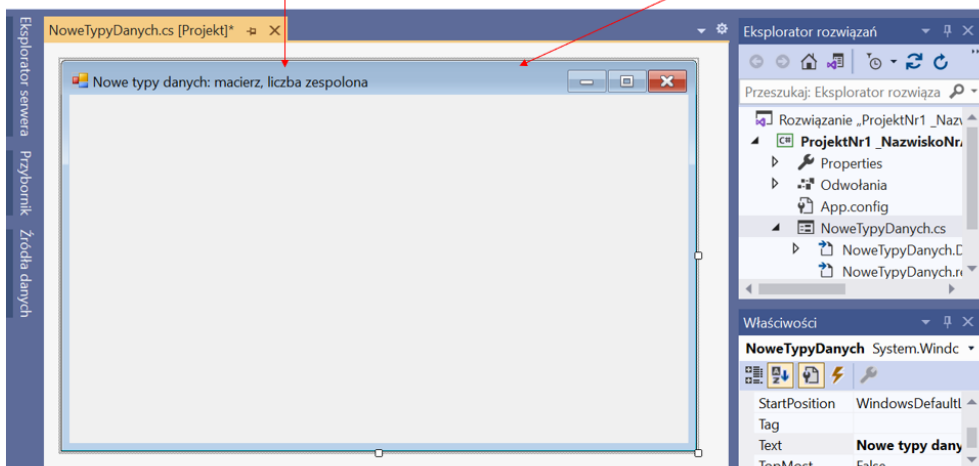


Opracował: L. Jung

Po zaznaczeniu formularza (białe uchwyty na jego krawędziach), w jego oknie właściwości odszukujemy pole edycyjne Text do którego wpisujemy tytuł naszego formularza Nowe typy danych: macierz, liczba zespolona :



następnie klikamy dowolne miejsce formularza i na jego pasku tytułowym będzie wyświetlony już nowy tytuł:

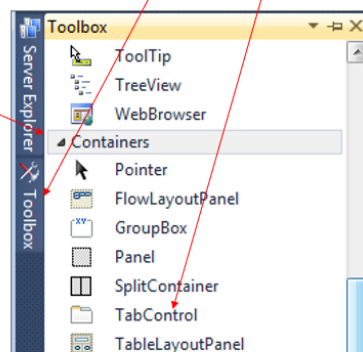


Opracował: L. Jung

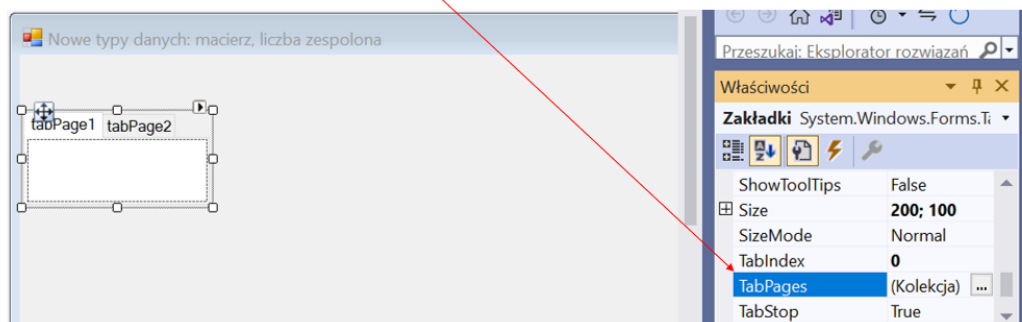
Projekt układu kart (stron) kontenera TabControl

Przystępujemy teraz do umieszczenia zakładek na naszym formularzu!

Na formularzu umieszczamy kontener TabControl (z zakładki Containers listwy narzędziowej Toolbox):

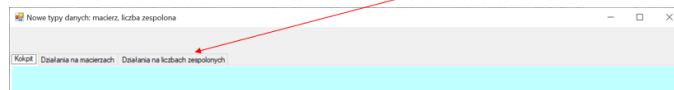


Kontener TabControl zawiera strony (karty) TabPage, które dodaje się za pośrednictwem właściwości TabPage:



Opracował: L. Jung

Kolejność stron (kart) w tej kolekcji [TabPage](#) odzwierciedla kolejność, w których strony (karty) te są wyświetlane w nagłówku kontenera [TabControl](#):

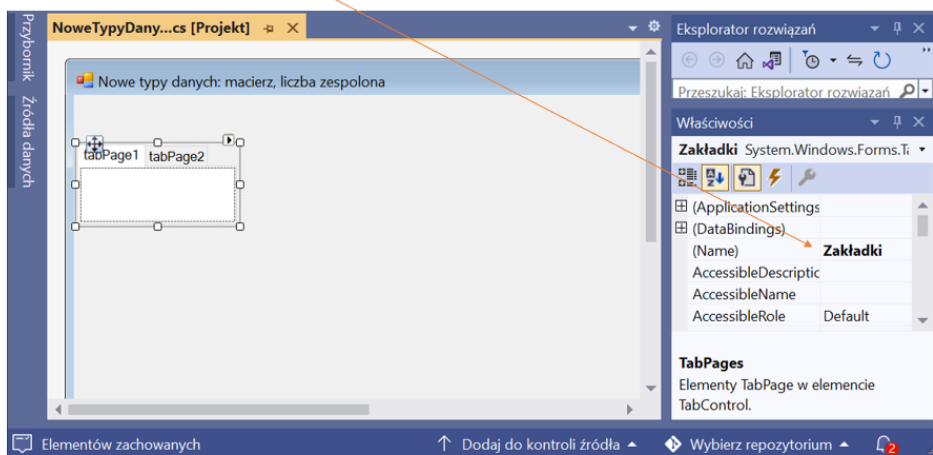


Bieżącą (otwartą, aktywną) stronę [TabPage](#) kontenera [TabControl](#) można zmienić klikając w inną jego stronę (kartę).

Programowa zmianę strony (karty) kontenera [TabControl](#) umożliwiają właściwości:

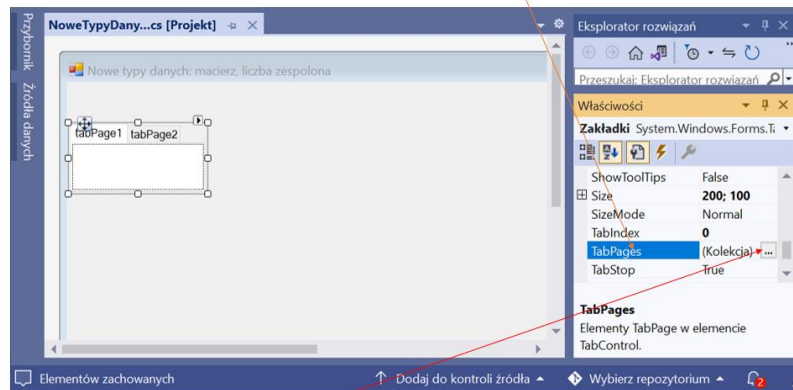
- [SelectedIndex](#)
- [SelectedTab](#)

Po umieszczeniu kontenera [TabControl](#) na formularzu, przypisujemy mu nazwę (np.) Zakładki:



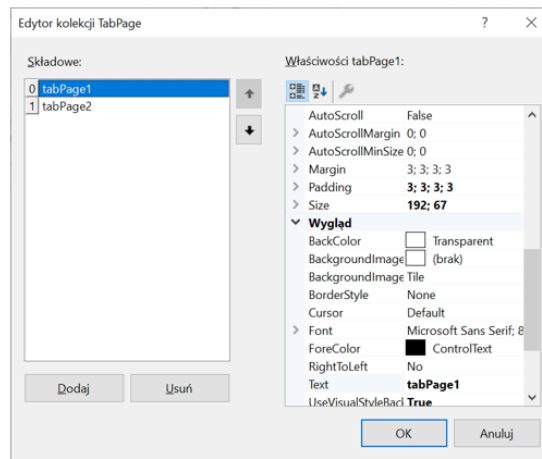
Opracował: L. Jung

W oknie właściwości odszukujemy atrybut TabPage:



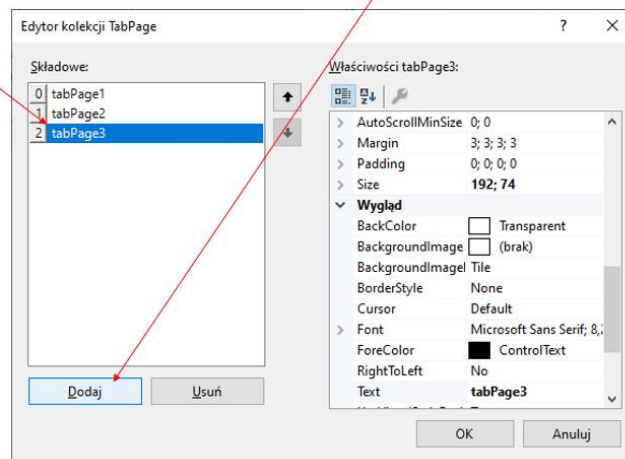
Wybieramy (kliknięciem) przycisk (...)

Kliknięcie przycisku (...) spowoduje wyświetlenie (odsłonięcie) okna dialogowego edycji stron (zakładek) kontenera TabControl:

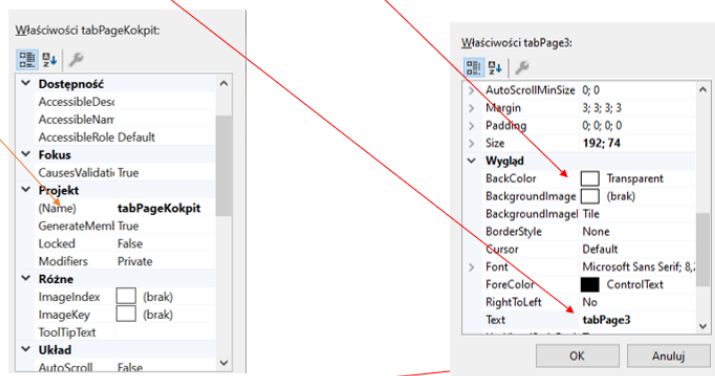


Opracował: L. Jung

W oknie dialogowym Edytor kolekcji TabPage (Collection Editor) klikamy przycisk poleceń Add (Dodaj) dla dodania nowej strony (zakładki) do kontenera TabControl:



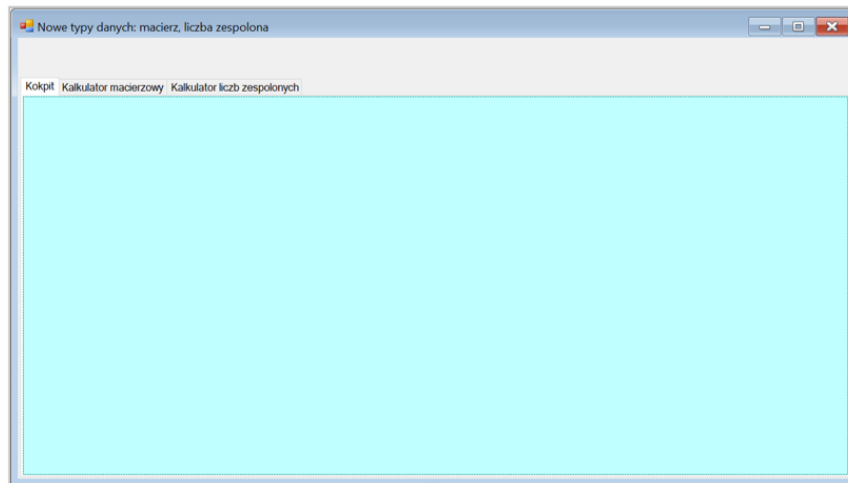
Teraz wybierając kolejno strony (zakładki) kontenera TabControl nadajemy im odpowiednie tytuł (w polu edycyjnym atrybutu Text), nazwę w polu (Name) oraz ustalamy kolor tła (atrybut BackColor):



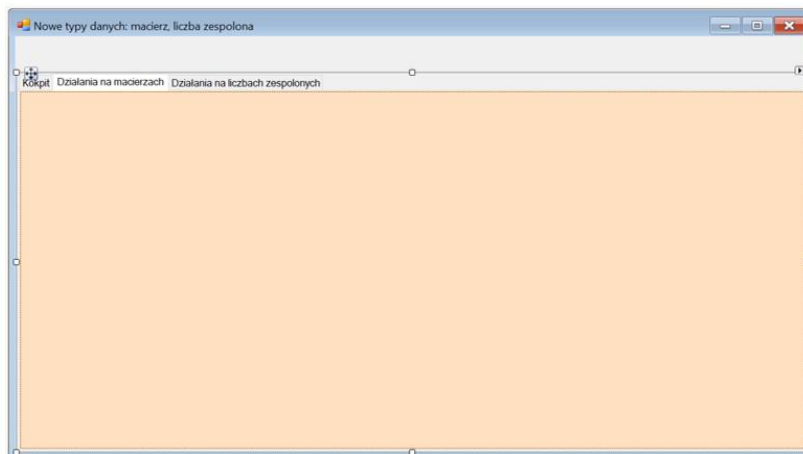
Po czym klikamy przycisk poleceń OK!

Opracował: L. Jung

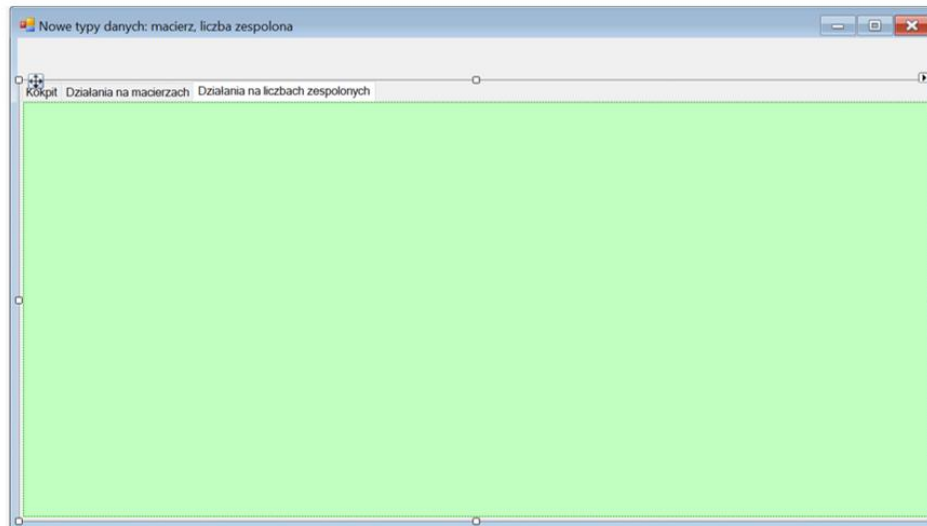
Zakładka Kokpit:



Zakładka Działania na macierzach:

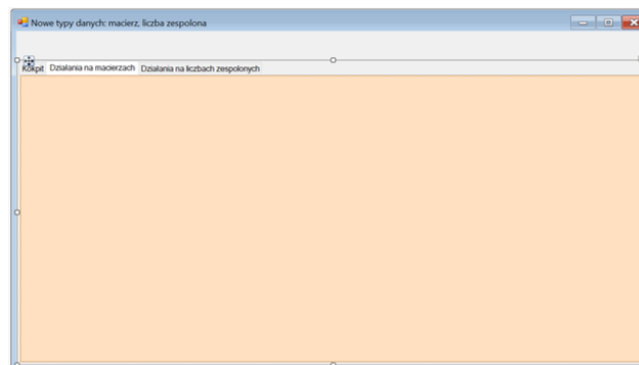


Zakładka Działania na liczbach zespolonych:



Nawigacja między stronami (kartami) kontenera TabControl

Umieszczony kontener TabControl na formularzu umożliwia (przy niezmiennych ustawieniach domyślnych) swobodne przechodzenie pomiędzy jego kartami (stronami) po kliknięciu nagłówka odpowiedniej karty:



Nadzorowana nawigacja między kartami Zakładki

Swobodnego przejścia między kartami uznajemy za niewskazane (to tak, jakbyśmy przechodzili z jednego pomieszczenia przez „ścianę”, a nie przez drzwi i korytarz), gdyż chcemy zapewnić swoistą „czystość” wykonywanych obliczeń: rachunku macierzowego lub rachunku na liczbach zespolonych.

Chcemy zapewnić przejście z jednej karty (strony) kontenera TabControl do drugiej, ale zawsze poprzez pierwszą kartę (stronę) Kokpit.

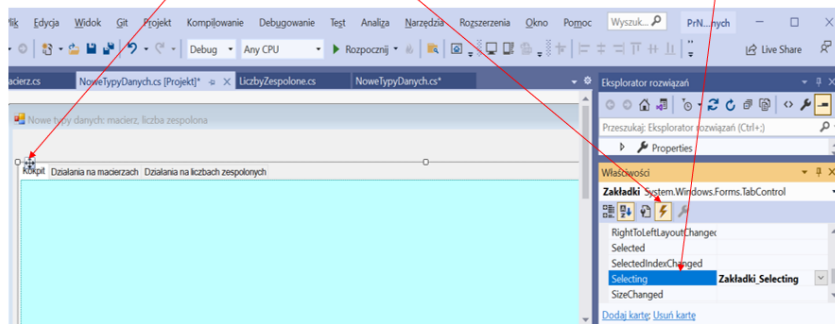
Jak to można zrealizować?

1. W widoku Code (Pokaż kod, klawisz F7), deklarujemy pomocniczą tablicę pełniącą funkcję opisu stanu zezwolenia do odpowiednich kart kontenera TabControl:

```
public partial class NoweTypyDanych : Form
{
    bool[] StanAktywnościStronZakładki =
        { true, true, true };
```

2. Dla kontenera TabControl projektujemy obsługę zdarzenia Selecting:

Zaznaczamy kontener Zakładki na formularzu i w oknie jego właściwości wybieramy zakładkę Zdarzenia, następnie odszukujemy zdarzenie Selecting, które klikamy dwukrotnie:



Projektujemy obsługę zdarzenia `Selecting`, tak, że po przejściu do zakładki (karty, strony):

- Kokpit (pełni funkcje „pulpitu nawigacyjnego” dla przejścia do innych kart `TabControl`) będzie możliwe następnie przejście do dowolnej innej zakładki (stron kontenera `TabControl`):

```
if ((e.TabPage == Zakładki.TabPages[0]) &&
    (StanAktywnościStronZakładki[0]))
{
    StanAktywnościStronZakładki[1] = true;
    StanAktywnościStronZakładki[2] = true;
    Zakładki.SelectedTab = tabPageKokpit;
    // lub: Zakładki.SelectedIndex = 0;
}
```

- **Działania na macierzach** nie będzie już możliwe przejście do zakładki **Działania na liczbach zespolonych** (ale będzie możliwe przejście do zakładki **Kokpit**):

```
if ((e.TabPage == Zakładki.TabPages[1]) &&
    (StanAktywnościStronZakładki[1]))
{
    StanAktywnościStronZakładki[2] = false;
    Zakładki.SelectedTab = tabPageMacierze;
    // lub: Zakładki.SelectedIndex = 1;
}
```

- Działania na liczbach zespolonych nie będzie już możliwe przejście do zakładki Działania na macierzach (ale będzie możliwe przejście do zakładki Kokpit):

```

if ((e.TabPage == Zakładki.TabPages[2]) &&
    (StanAktywnościStronZakładki[2]))
{
    StanAktywnościStronZakładki[1] = false;
    Zakładki.SelectedTab = tabPageLiczbyZespolone;
    // lub: Zakładki.SelectedIndex = 2;
}
else e.Cancel = true; // kasowanie przejścia do zakładki
}

```

Przy braku zezwolenia (w tablicy `StanAktywnościStronZakładki`) na przejście do wymaganej zakładki, zdarzenia `Selecting` będzie kasowane.

Pełny zapis metody obsługi zdarzenia `Selecting` przyjmie postać:

```

private void Zakładki_Selecting(object sender, TabControlCancelEventArgs e)
{
    if ((e.TabPage == Zakładki.TabPages[0]) &&
        (StanAktywnościStronZakładki[0]))
    {
        StanAktywnościStronZakładki[1] = true;
        StanAktywnościStronZakładki[2] = true;
        Zakładki.SelectedTab = tabPageKokpit; // lub: Zakładki.SelectedIndex = 0;
    }
}

```



```

else
if ((e.TabPage == Zakładki.TabPages[1]) && (StanAktywnościStronZakładki[1]))
{
StanAktywnościStronZakładki[2] = false;
Zakładki.SelectedTab = tabPageMacierze; // lub: Zakładki.SelectedIndex = 1;
}
else
if ((e.TabPage == Zakładki.TabPages[2]) && (StanAktywnościStronZakładki[2]))
{
StanAktywnościStronZakładki[1] = false;
Zakładki.SelectedTab = tabPageLiczbyZespolone;
// lub: Zakładki.SelectedIndex = 2;
}

else
e.Cancel = true; // kasowanie przejścia do zakładki
}

```

Projekt i implementacja zakładki Działania na macierzach

Projektujemy interfejs graficzny zakładki Działania na macierzach:

