Pojęcia

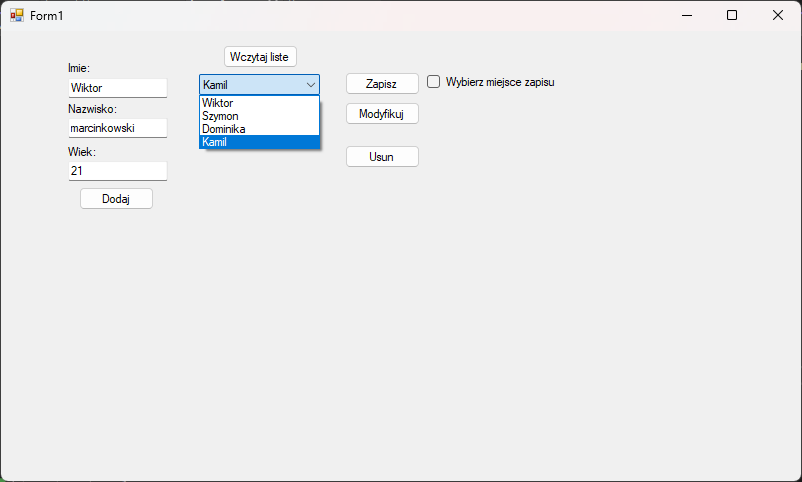
**Serializacja –** proces konwersji struktury danych lub obiektu w do formatu strumienia bajtów, XML lub JSON (np. do pliku), w celu przechowywania lub przesyłania danych między różnymi systemami lub procesami.

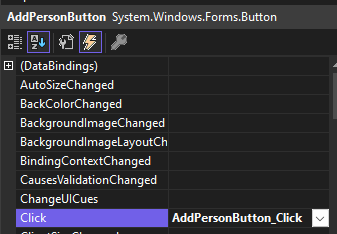
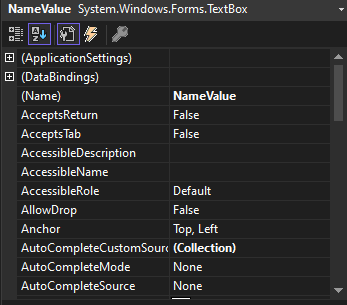
**Deserializacja –** proces konwersji danych, które zostały wcześniej zserializowane spowrotem do pierwotnej formy tj. odczyt np.z xml i wczytanie do pamięci obiektu wraz z jego wlasciwosciami.

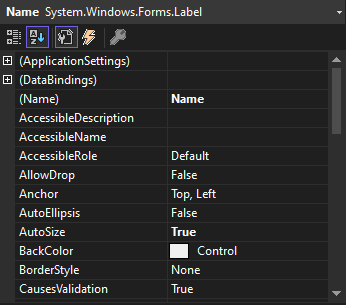
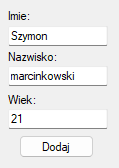
Zastosowanie Programu

Program służy do serializacji i deserializacji \*(listy osób)\* Listy obiektów klasy *Person*, która składa się z pól *ID*, *Name*, *Surname* oraz *Age* za pomocą serializacji XML.

Wygląd Programu



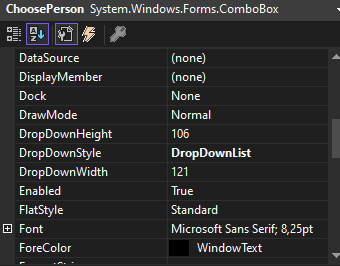
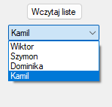
Opis Programu

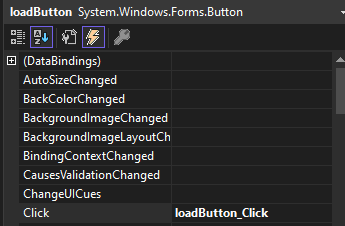


**Lewa sekcja**

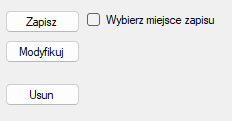
**Każdy „TextBox” jest opisany za pomocą etykiety „Label”.**

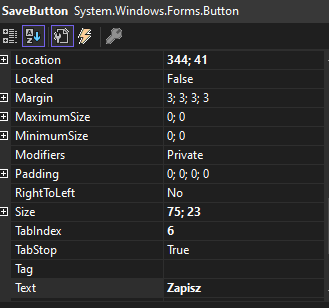
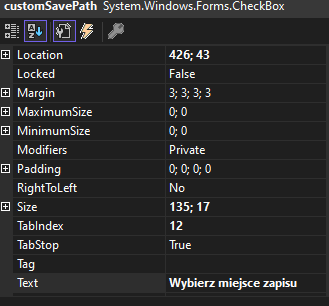
1. **Okna tekstowe (TextBox):  
   wprowadzanie informacji o osobie:**
   * **Imię**
   * **Nazwisko**
   * **Wiek**
2. **Przycisk „Dodaj” (Button):**
   * **W zakładce „events” zostało przypisane wywoałanie metody „*AddPersonButton\_Click*”.**
   * **Przycisk tworzy z wpisanych danych obiekt *Person* i dodaje go do listy obiektów.**

**Środkowa sekcja**

****

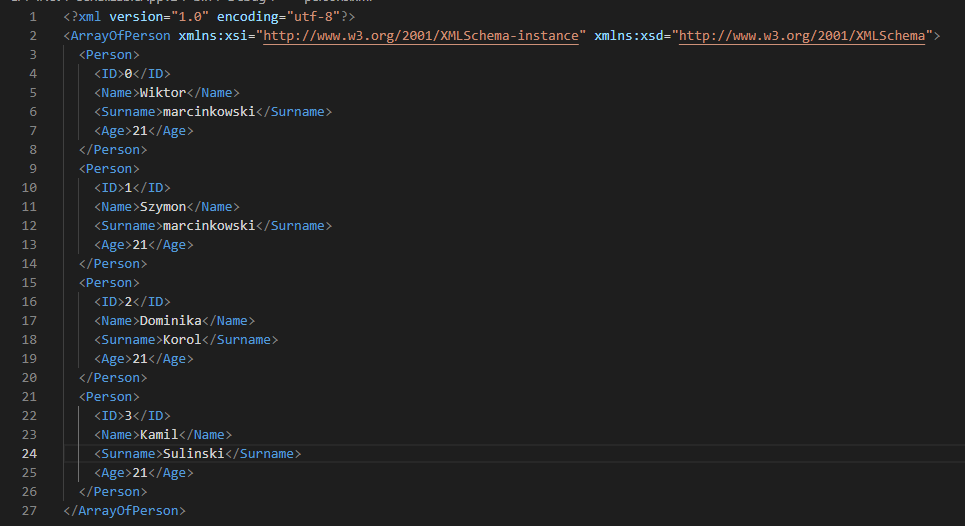
1. **Przycisk „Wczytaj liste” (Button):**
   * **Pozwala wybrać inny zserializowany plik XML przechowujący listę obiektów klasy *Person* niż domyślnie przechowywany plik „Person.xml” w plikach programu.**
   * **Deserializuje plik i wczytuje go do listy**
2. **Kontrolka „ComboBox” z ustawioną właściwością „DropDownStyle” na *DropDownList.***
   * **Element listy wyświetla właściwość *Imię* z listy obiektów klasy Person. Po wybraniu wczytuje wszystkie właściwości obiektu do Lewej Sekcji**

**Prawa sekcja**

****

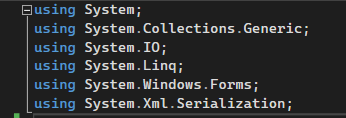
1. **Przysk „Zapisz”:**
   * **serializuje listę obiektów klasy do XML i zapisuje ją do pliku Person.xml, lub do wybranego przez użytkownika pliku.**
2. **CheckBox „Wybierz miejsce zapisu”:**
   * **jeśli jest zaznaczony, to po kliknięciu przyisku „Zapisz” otwiera nam się okienko, które pozwala wybrać miejsce zapisu pliku.**
3. **Przycisk „Modyfikuj”:**
   * **Pozwala zmodyfikować wybrany w liscie rozwijanej obiekt.**
4. **Przycisk usun:**
   * **Usuwa wybrany z listy obiekt klasy *Person***

Przykładowa struktura pliku XML

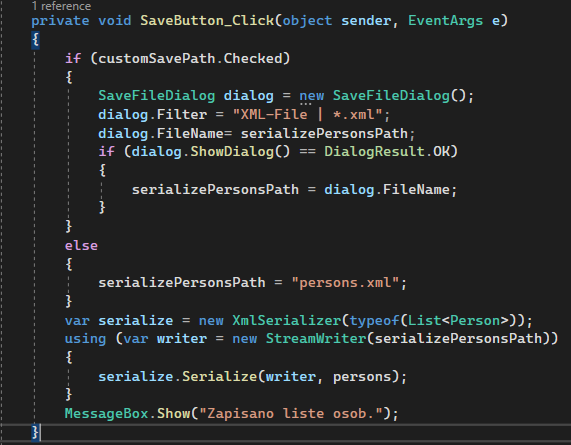


Opis kodu

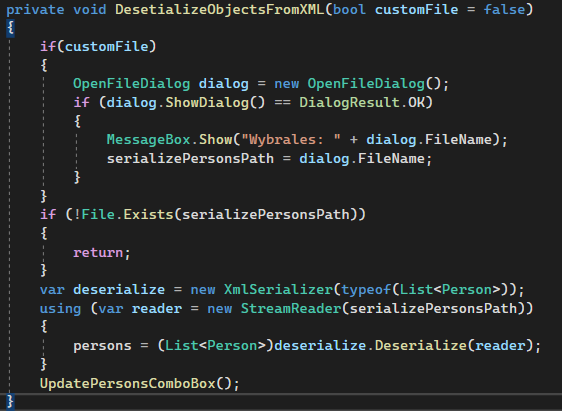
**Biblioteki:**

****

Biblioteki podane wyżej są niezbędne do poprawnego działania programu.

**Serializacja:**

Ten kod służy do zapisania listy obiektów typu Person do pliku XML. Jeśli użytkownik wybierze niestandardową ścieżkę zapisu, otwierane jest okno dialogowe, w którym użytkownik może wybrać plik i ścieżkę zapisu. W przeciwnym razie, plik zostanie zapisany w domyślnej ścieżce "persons.xml". Następnie tworzony jest obiekt XmlSerializer, który jest odpowiedzialny za serializację listy obiektów do pliku XML. Na końcu, wyświetlany jest komunikat informujący o pomyślnym zapisaniu danych do pliku.

**Deserializacja:**

Ten kod służy do wczytania i deserializacji danych z pliku XML do listy obiektów typu Person. Jeśli użytkownik wybrał niestandardowy plik do wczytania, otwiera się okno dialogowe, które umożliwia użytkownikowi wybór pliku. Następnie, jeśli plik nie istnieje, funkcja kończy działanie. W przeciwnym razie tworzony jest obiekt XmlSerializer, który jest odpowiedzialny za deserializację danych z pliku XML do listy obiektów typu Person. Następnie plik jest odczytywany przy użyciu StreamReadera, a dane są deserializowane. Na koniec, funkcja UpdatePersonsComboBox() aktualizuje ComboBox z danymi o osobach.