

## Ćwiczenie 4 (za 10 lub 10 + 3 punktów, środowisko jednomaszynowe)

Dla grup: wt/TP07, wt/TP09

1. Przestudiuj opis i zrealizuj w domu (**przed zajęciami wirtualnymi**) aplikację opisaną w **Ćwiczenie 4, WCF – kontrakty danych, operacje jednokierunkowe i zwrotne (callbacks), Zadanie - część I, plik: Cw4-WCF-2a.pdf**.
2. Podczas zajęć zdalnych (labo wirtualne) zbuduj serwis WCF i klienta tego serwisu opisane w **Zadanie – Część II**.
3. Aplikacja kliencka - spełnia wymagania podane w **Zadanie – część II, B.1**.
4. Serwis (usługa) – spełnia wymagania podane w **Zadanie – część II, B.2**.

### **Uwaga:**

- a. aplikacja/serwis zapewnia funkcje podręcznej biblioteki – *minimalne wymagania poniżej*.
  - a. Oprócz kontraktu danych zdefiniuj/implementuj także **Fault Contract**
5. Serwis + klient konsolowy z prostym menu => ocena max. 10 punktów.
  6. Serwis + klient graficzny => ocena max. 10 + 3 punktów.
  7. Udokumentuj – w pliku *nazwisko.docx* i/lub *nazwisko.pdf* - kolejne kroki kompilacji i uruchamiania/testowania serwisu WCF i klienta tego serwisu środowisku jednomaszynowym.

**Bardzo ważne:** dokumentacja - w postaci kroków - musi zawierać krótkie komentarze/polecenia i screenshoty wprowadzonych komend i uzyskiwanych wyników. Należy również podać użytą wersję .NET Framework.

8. Na zakończenie prześlij=wgraj rozwiązania (*solution*) **Zadanie – część I, Zadanie – część II** wraz dokumentacją dot. **Zadanie – część II** na **portal**. Obowiązują następujące reguły:

- wszystkie pliki mają być skompresowane do jednego pliku
- nazwa pliku ma mieć format: **X-Nazwisko-Imie.zip**  
gdzie **X** - numer zadania (taki jak na witrynie),  
- bez użycia spacji - zamiast nich są myślniki (!),  
- bez polskich znaków diaktrycznych (!)  
np.: **3-Ziolkowski-Michal.zip**
- wgrany **projekt powinien być wyczyszczony** ("odchudzony")!

### **Uwaga, ważne:**

1. Pliki z kodem i załączona dokumentacja musi być tak napisana, że szybko i łatwo będę w stanie powtórzyć wszystkie kroki kompilacji, uruchamiania i testowania aplikacji.

2. Kompletne rozwiązanie (*solution*) wgrywane na ePortal musi być oczyszczone ze zbędnych plików/folderów, konieczne usuń podfoldery: **bin**, **obj**, **.vs**

#### Minimalna funkcjonalność aplikacji/serwisu („podręcznej” biblioteki).

1. Dodaj książkę; *parametry*: tytuł, autor, ISBN, rokWydania, cena
2. Szukaj po ISBN; *parametr*: dokładna wartość ISBN
3. Szukaj po tytule; *parametr*: tytuł lub kilka początkowych znaków tytułu
4. Modyfikuj tytuł; *parametr*: ISBN, nowy tytuł
5. Pobierz opisy (lista: tytuł, autor)
6. Aktualna wartość wszystkich książek
7. Pobierz książki – *operacja asynchroniczna* (zwraca wynik po **n** sekundach, **n**-hardkodowane,  $n > 10$  sekund)
8. Operacja **info**

#### *Uwaga*

Bezparametrowa operacja **info** typu *callback* zwraca klientowi napis zawierający:

- nr albumu i imię autora (-ów) aplikacji,
- datę i czas wywołania jej w serwisie,
- datę i czas wysłania callback (odczytany w serwisie)

Sekundowe opóźnienie wysłania callbacku jest hardkodowane i wynosi nie mniej niż 10 sekund.