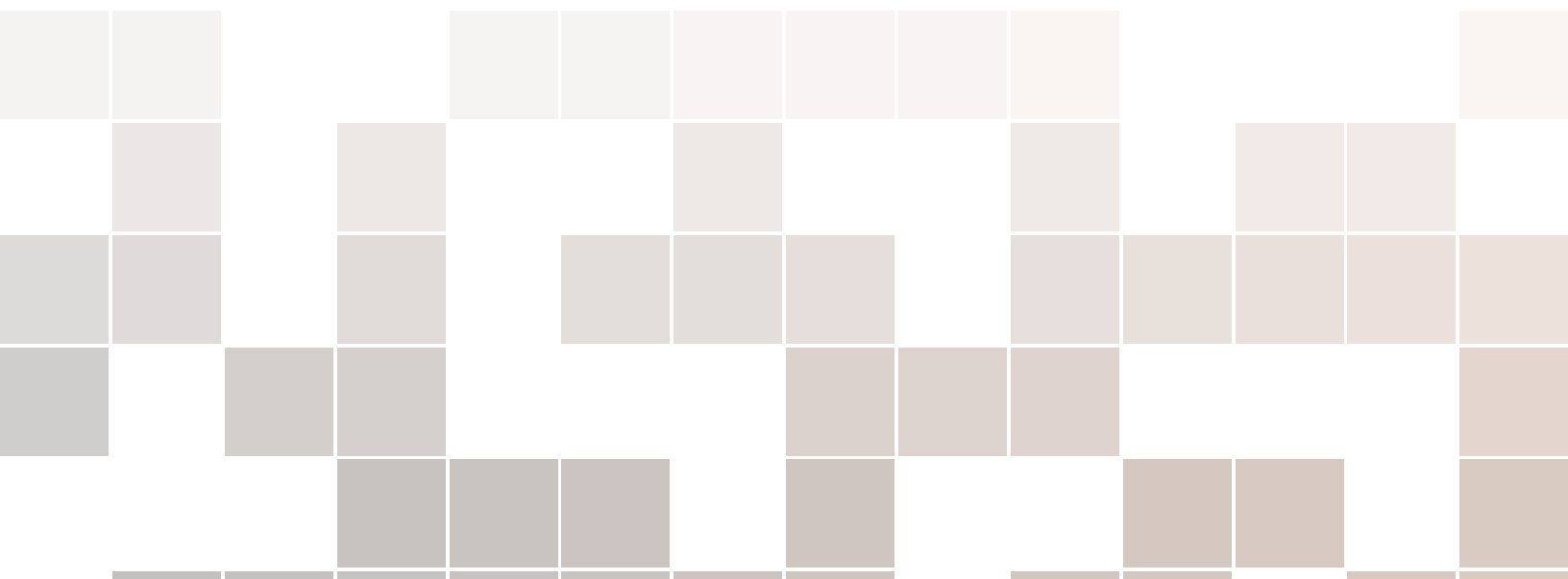


Programowanie funkcyjne

Materiały z wykładu i laboratorium

Łukasz Bartczuk



7. Podstawy programowania funkcyjnego w C#

7.1 Zadania do samodzielnego wykonania

Zadanie 7.1 Napisz klasę niemutowalną, która będzie reprezentowała książkę. Powinna mieć następujące pola:

1. ID - identyfikator książki w formacie: B-00-001
2. Autorzy - kolekcja zawierająca imię i nazwisko autora (musi być co najmniej jeden)
3. Tytuł (do 50 znaków)
4. Rok publikacji - rok nie mniejszy niż 2015, i nie większy niż rok następujący po obecnym
5. Cena książki - kwota i waluta

Zastanów się, jakie typy wartościowe zdefiniować. Jak zamodelować dodawanie autora, skoro klasa jest niemutowalna.

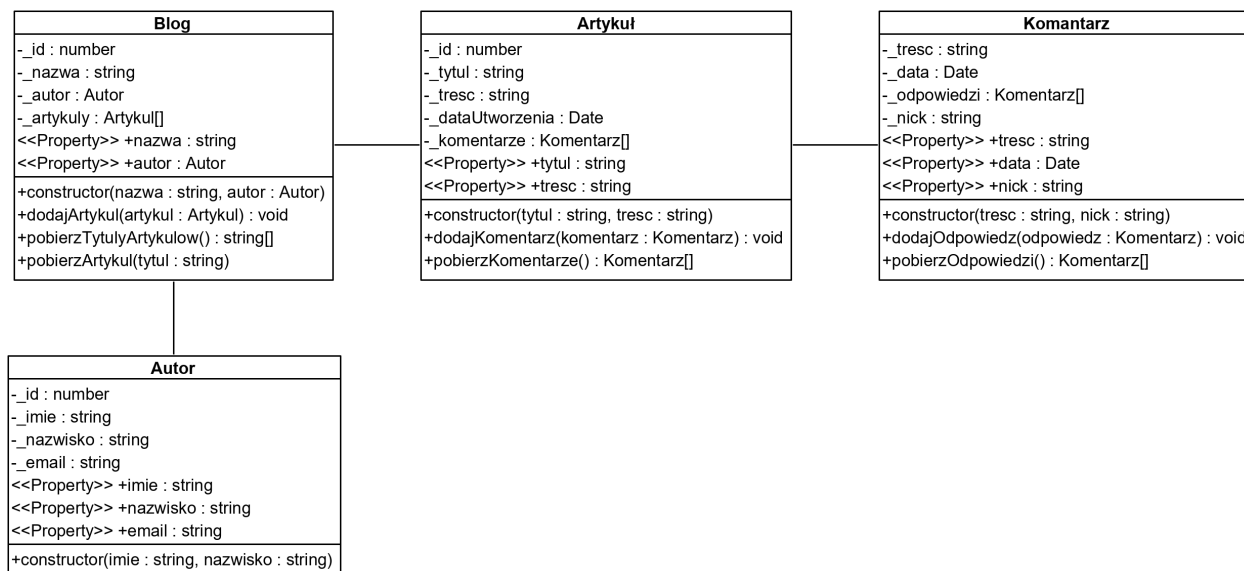
Zadanie 7.2 Zapisz klasę Książka jako klasę tylko do odczytu. Właściwości pojedynczych wartości powinny być zamodelowane jako tylko do odczytu. Do zdefiniowania kolekcji wykorzystaj interfejs IReadOnlyList.

Zadanie 7.3 Zapisz klasy z zadania 1 za pomocą rekordów.

Zadanie 7.4 Zaimplementuj klasy zgodnie ze schematem przedstawionym na rys. 7.1.

Zadanie 7.5 Proszę zmodyfikować poprzednie zadanie w taki sposób, aby utworzenie obiektów niezgodnych z poniższymi wymaganiami było niemożliwe:

1. Dla klasy Autor:
 - pola imię, nazwisko, email są wymagane
 - pole email powinno zawierać poprawny adres (pod względem składniowym)
2. Dla klasy Blog:
 - pole nazwa nie może być puste



Rysunek 7.1: Diagram klas do zadania 7.4

- autor musi być podany

3. Dla klasy Artykuł:

- pole tytuł nie może być puste
- w momencie tworzenia data powinna być ustalona na aktualną

Jakie są możliwości rozwiązania tego zadania.

Zadanie 7.6 Napisz program, który będzie pozwalał przechowywać następujące informacje o osobie: imię, nazwisko, wiek. Przygotuj odpowiedni typ danych. Napisz program, który będzie pozwalał na wprowadzenie informacji o osobie, modyfikowanie ich oraz wyświetlenie. Po uruchomieniu programu użytkownik powinien zobaczyć menu:

WYBIERZ OPCJE

- 1 – utworzenie rekordu
- 2 – modyfikacja rekordu
- 3 – pokaz rekord
- 4 – zakończenie programu

Po wybraniu odpowiedniej opcji użytkownik powinien mieć możliwość wprowadzenia danych (program powinien wyświetlać informacje o jakie dane prosi użytkownika), lub ich zobaczenia. Podczas modyfikowania danych jeżeli użytkownik wprowadzi łańcuch pusty (w przypadku imienia lub nazwiska) lub 0 (w przypadku wieku) program powinien zachowywać wcześniejsze dane. Po zakończeniu wprowadzania lub przeglądania danych oraz w przypadku, gdy użytkownik wybierze inną opcję niż od 1 do 4 program powinien wracać do menu głównego.

Zadanie 7.7 Napisz program, który będzie pozwalał użytkownikowi na wprowadzanie z klawiatury liczb oraz operatorów i funkcji matematycznych w odwrotnej notacji polskiej (przy czym my zakładamy, że każdy element jest podawany w nowej linii i

zatwierdzany osobno). Po podaniu symbolu '=' program powinien obliczać wartość podanego wyrażenia. Przykładowo:

```
10.0  
20.0  
+  
sin  
=
```

jest innym sposobem zapisania wyrażenia $\sin(10.0 + 20.0)$ i jego wykonanie powinno na ekranie wyświetlić: -0.9880316241 . Jeżeli użytkownik poda za mało wartości do przeprowadzenia danej operacji na ekranie należy wyświetlić stosowny komunikat np.

```
10.0  
+  
=
```

powinno wyświetlić komunikat: "Operator + wymaga podania dwóch wartości". Wykorzystaj mutowalny stos z przestrzeni nazw `System.Collections.Generic`

Zadanie 7.8 Rozwiąż poprzednie zadanie z wykorzystaniem niemutowalnego stosu z przestrzeni nazw `System.Collections.Immutable`.