

Jakub Balcerzak

## **Inżynieria Oprogramowania**

### ***Instrukcja laboratoryjna 1 - sprawozdanie***

Zapoznanie się z wybranym narzędziem UML i wprowadzenie do UML.

### 1. Poznane narzędzia:

- Visual Paradigm Community Edition (CE) – program ten jest narzędziem, które można zastosować m.in. w: architekturze przedsiębiorstw, zarządzaniu projektami, rozwoju oprogramowania, projektach grupowych.

### 2. Opis biznesowy „świata rzeczywistego” (punkt 1.):

Opis biznesowy „świata rzeczywistego” został zaprezentowany na przykładzie katalogu książek używanego w bibliotece. Podzielony został na trzy części:

- opis zasobów ludzkich – opis działania aplikacji przez pryzmat użytkownika końcowego np. bibliotekarza zarządzającego projektowanym katalogiem,
- przepisy – określenie, za co odpowiada użytkownik końcowy podczas wykorzystywania systemu,
- dane techniczne – techniczne aspekty wykorzystania systemu.

### 3. Wymagania programu opracowane na podstawie opisu „świata rzeczywistego” (punkt 2.):

Na podstawie analizy opisu utworzonego w punkcie 1. instrukcji zostały opisane wymagania dla danego projektu:

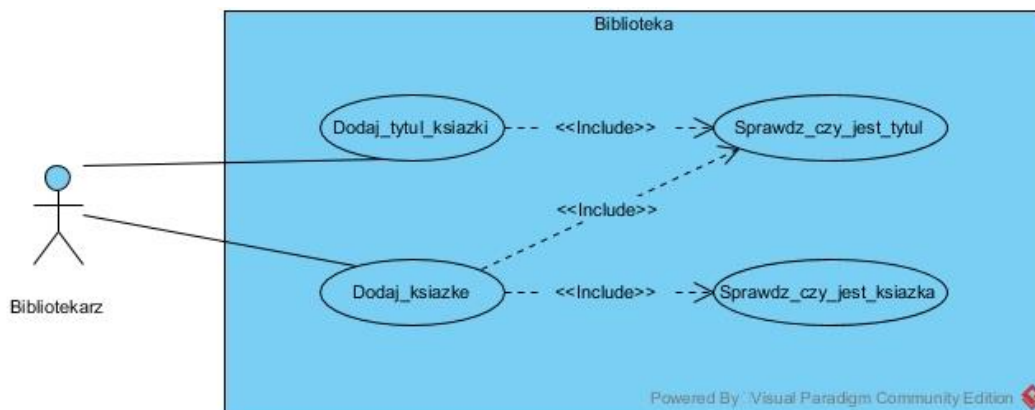
- wymagania funkcjonalne – konkretne aspekty funkcjonalności aplikacji, które definiują działania, które mogą być udostępniane dla użytkownika lub samego oprogramowania,
- wymagania niefunkcjonalne – aspekty funkcjonalności aplikacji, które nie definiują żadnych działań, lecz także wpływają na ostateczny kształt produktu.

### 4. Tworzenie projektu w Visual Paradigm (punkt 3.):

Nowy projekt tworzy się na karcie Project. Po kliknięciu Nowy Projekt wyskakuje okno wstępne, w którym użytkownik może wyspecyfikować parametry tworzonego projektu. Projekt jest zapisywany w domyślnej przestrzeni pracy (ang. workspace).

### 5. Diagram przypadków użycia (punkt 4.):

Diagram przypadków użycia prezentuje interakcję użytkownika z systemem, który składa się z przypadków użycia (PU). Diagram ten definiuje „co”, a nie „jak” robi system. Poniżej prezentowany jest diagram dla opisywanego w punkcie 1. instrukcji świata:

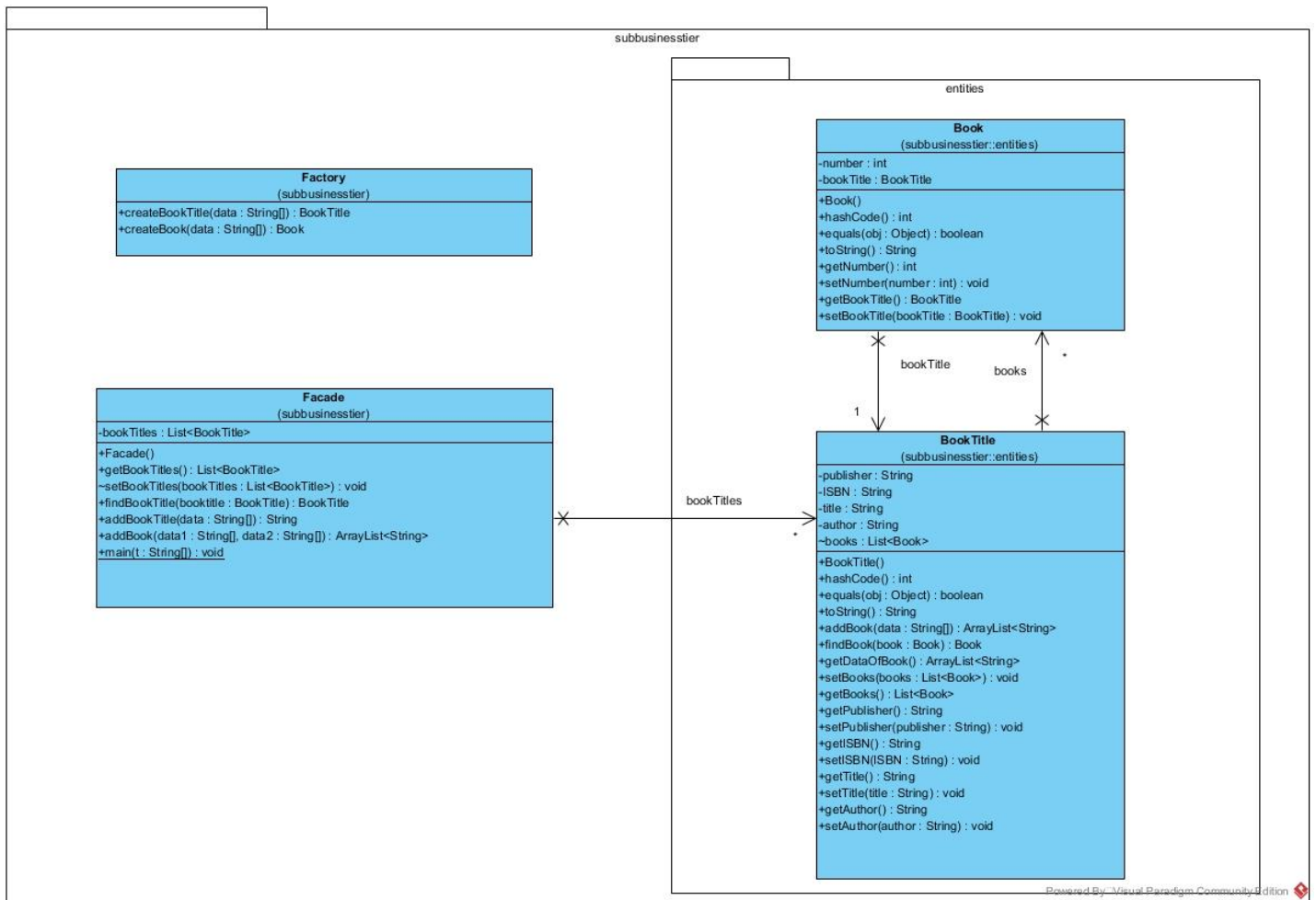


## 6. Diagram klas i diagram pakietów (punkt 5.2.):

Diagram klas opisuje strukturę systemu poprzez zdefiniowanie jego klas zawierających w sobie konkretne atrybuty oraz metody, a także związków pomiędzy obiektami. Diagram ten jest ściśle związany z podejściem zorientowanym obiektowo.

Diagram pakietów wykorzystywany przede wszystkim w średnich i dużych projektach ma za zadanie zdefiniować organizację oraz układ elementów w nim zawartych. Obrazuje on nie tylko strukturę, ale także zależności pomiędzy innymi podsystemami i innymi modułami.

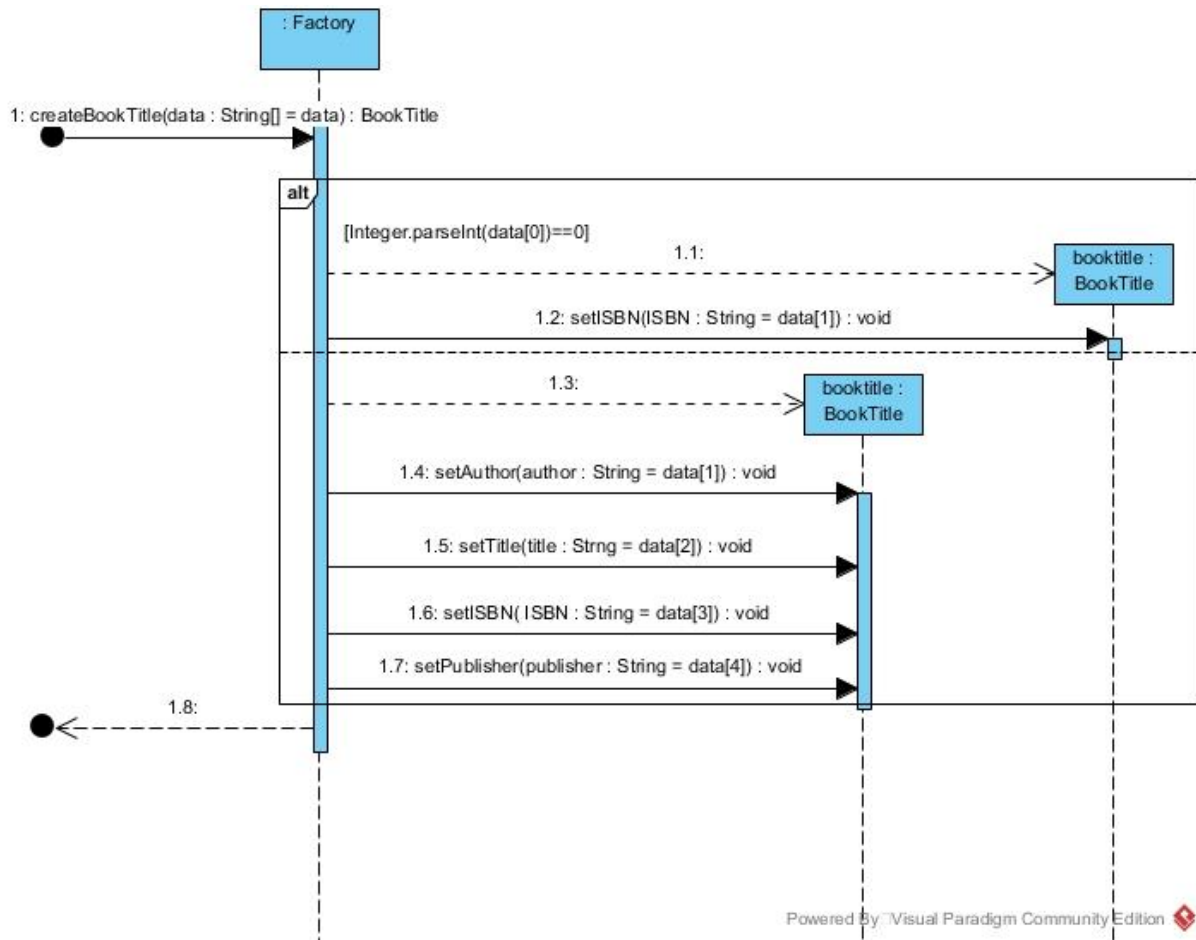
Poniższy rysunek zawiera dwa pakiety, w których osadzone zostały klasy:



## 7. Diagram sekwencji (punkty 5.4., 5.5.):

Diagram sekwencji należy do diagramów interakcji, które uszczegóławiają sposób, w jaki przeprowadzane są kolejne operacje. Diagramy te są zorientowane czasowo, co ma na celu zaprezentowanie przebiegu wykonywania się kolejnych akcji/interakcji. Pozioma oś pozwala odnosić się do niej, jako do osi czasu.

Poniższy diagram prezentuje działanie sekwencyjne metody createBookTitle pochodzącej z klasy Factory:



## 8. Uwagi i wnioski:

Ponieważ w środowisku Visual Paradigm CE nie jest możliwa tzw „inżyniera w przód” (ang. forward engineering) punkt 5.3. przerabianej instrukcji dot. wykonania projektu typu Java Class Library został pominięty. Skupiono się na wykonaniu pozostałych punktów instrukcji, które zostały powyżej opracowane w niniejszym sprawozdaniu.