

Przykład 1

Iteracja 2 tworzenia oprogramowania

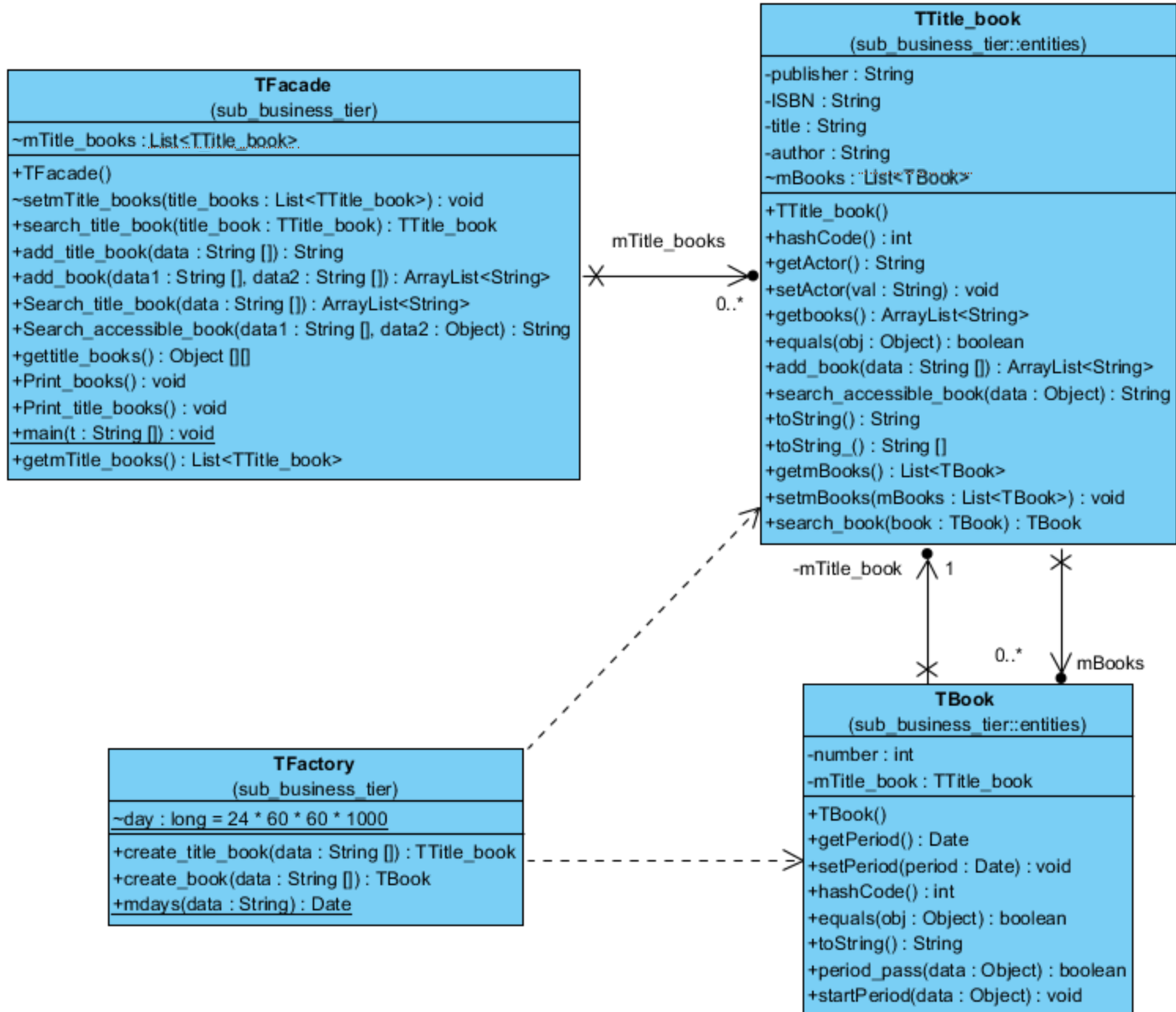
Diagramy klas i sekwencji:

Relacja 1 do 0..* między klasami
modelu obiektowego

Cele drugiej iteracji

- 2-i etap projektu.
 - Instrukcja zawiera przykłady powiązań 1:* (jeden do wiele) między klasami reprezentującymi dane: z 1-go etapu (TTitle_book oraz nową klasą TBook, która została zidentyfikowana po zaprojektowaniu kolejnego przypadku użycia PU Dodaj_ksiazke).
 - Wykonaj kopię projektu UML z etapu 1 (laboratorium 2). Uzupełnij diagramy przypadków użycia, klas oraz uzupełnij i/lub dodaj nowe diagramy sekwencji. Należy wykorzystać te dwa typy powiązań 1:* oraz 1:0..1 w wykonywanym 2-im etapie projektu i jego implementacji na podstawie instrukcji do laboratorium 3.
 - Wykonaj prosty program stanowiący realizację 2-go etapu wybranego projektu. Program ten stanowi rozwinięcie programu z etapu 1.

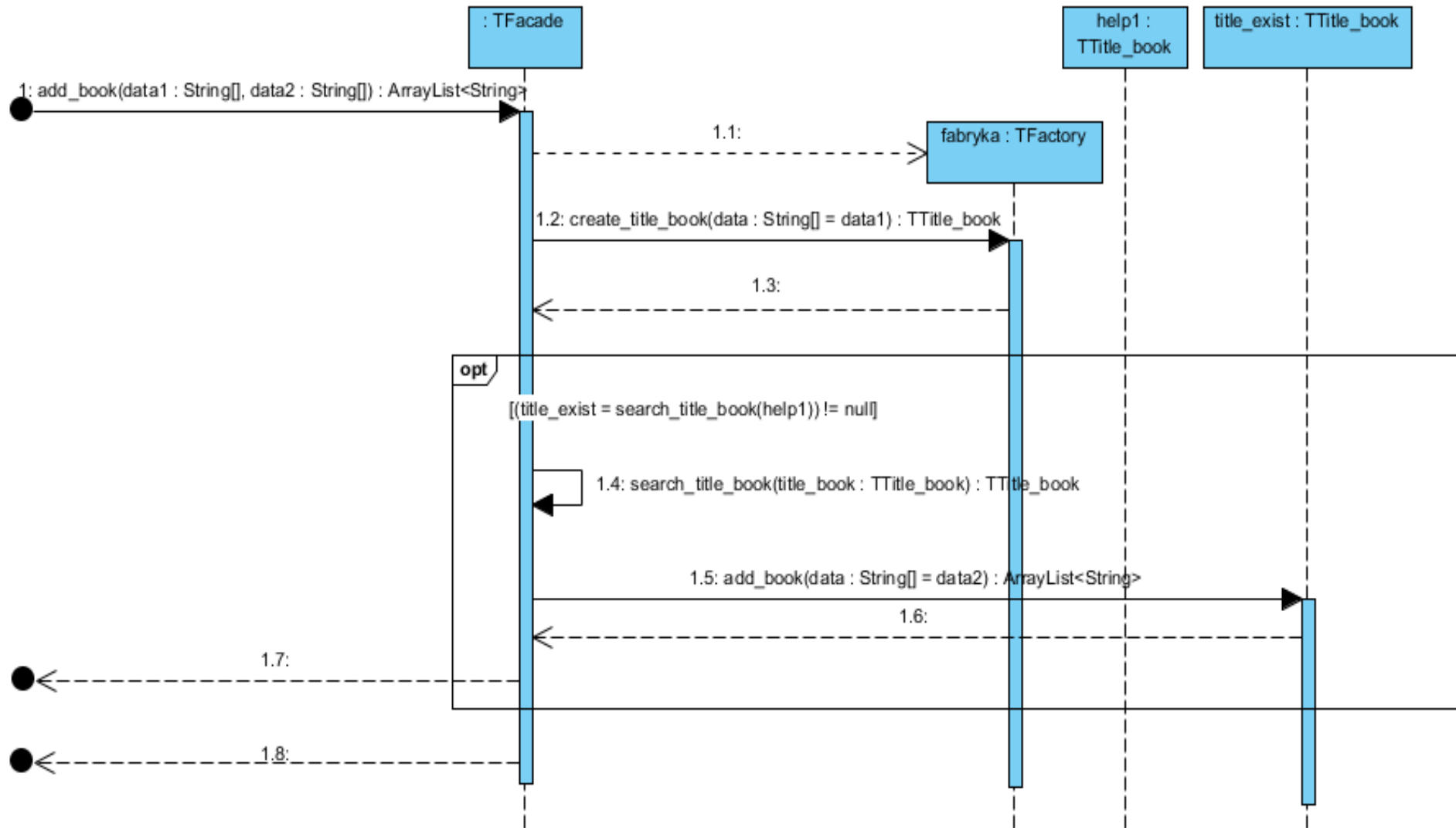
1. Diagram klas



2.Iteracja 2

Dodawanie książki

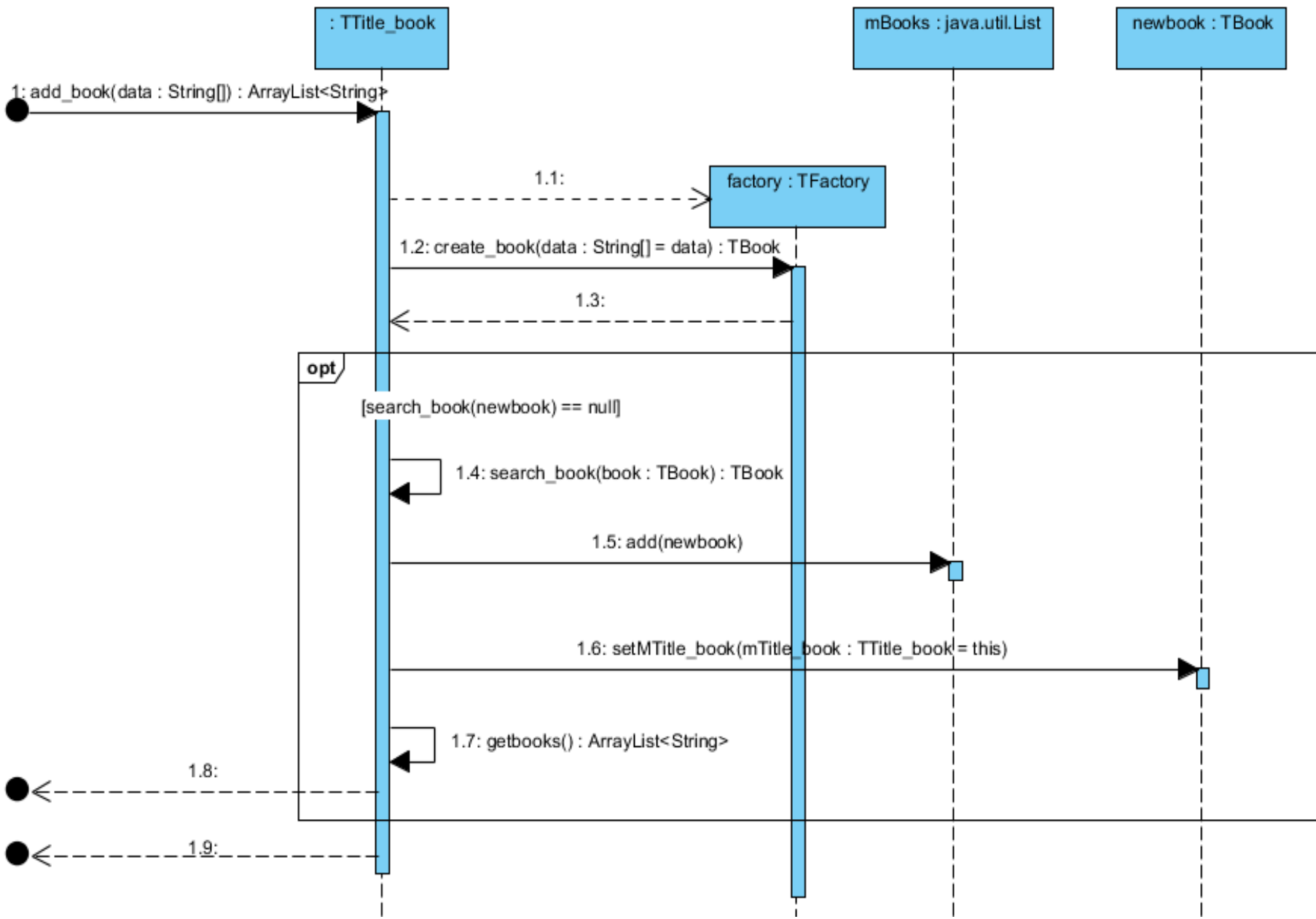
2.1. Diagram sekwencji metody add_book klasy TFacade



2.2. Kod metody: add_book klasy TFacade

```
public ArrayList<String> add_book(String data1[], String data2[]) {  
    TTitle_book help1, title_exist;  
    TFactory fabryka = new TFactory();  
    help1 = fabryka.create_title_book(data1);  
    if ((title_exist = search_title_book(help1)) != null) {  
        return title_exist.add_book(data2);  
    }  
    return null;  
}
```

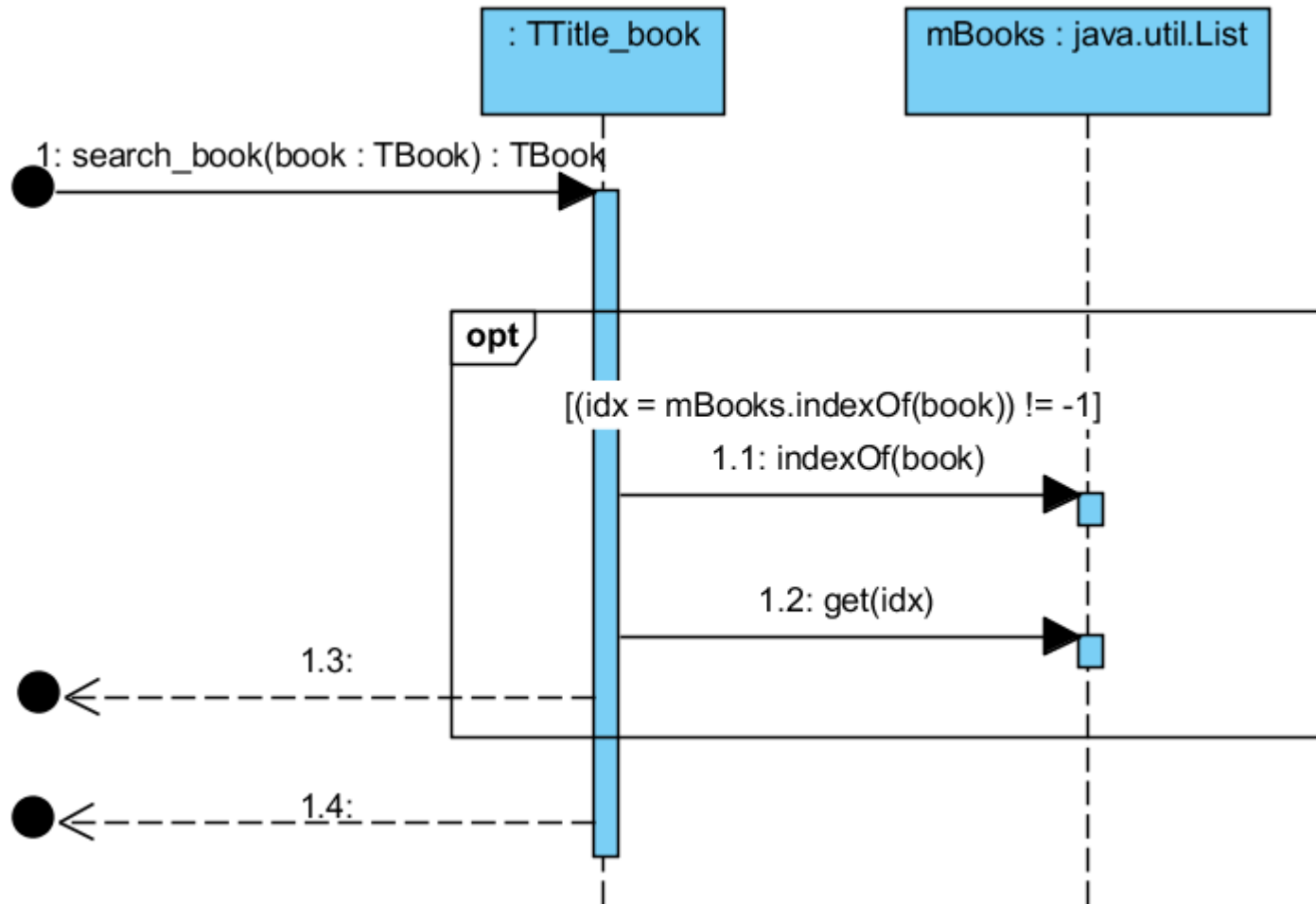
2.3. Diagram sekwencji metody add_book klasy TTitle_book



2.4. Kod metody: add_book klasy TTitle_book

```
public ArrayList<String> add_book(String data[]) {  
    TFactory factory = new TFactory();  
    TBook newbook;  
    newbook = factory.create_book(data);  
    if (search_book(newbook) == null) {  
        mBooks.add(newbook);  
        newbook.setmTitle_book(this);  
        return getbooks();  
    }  
    return null;  
}
```

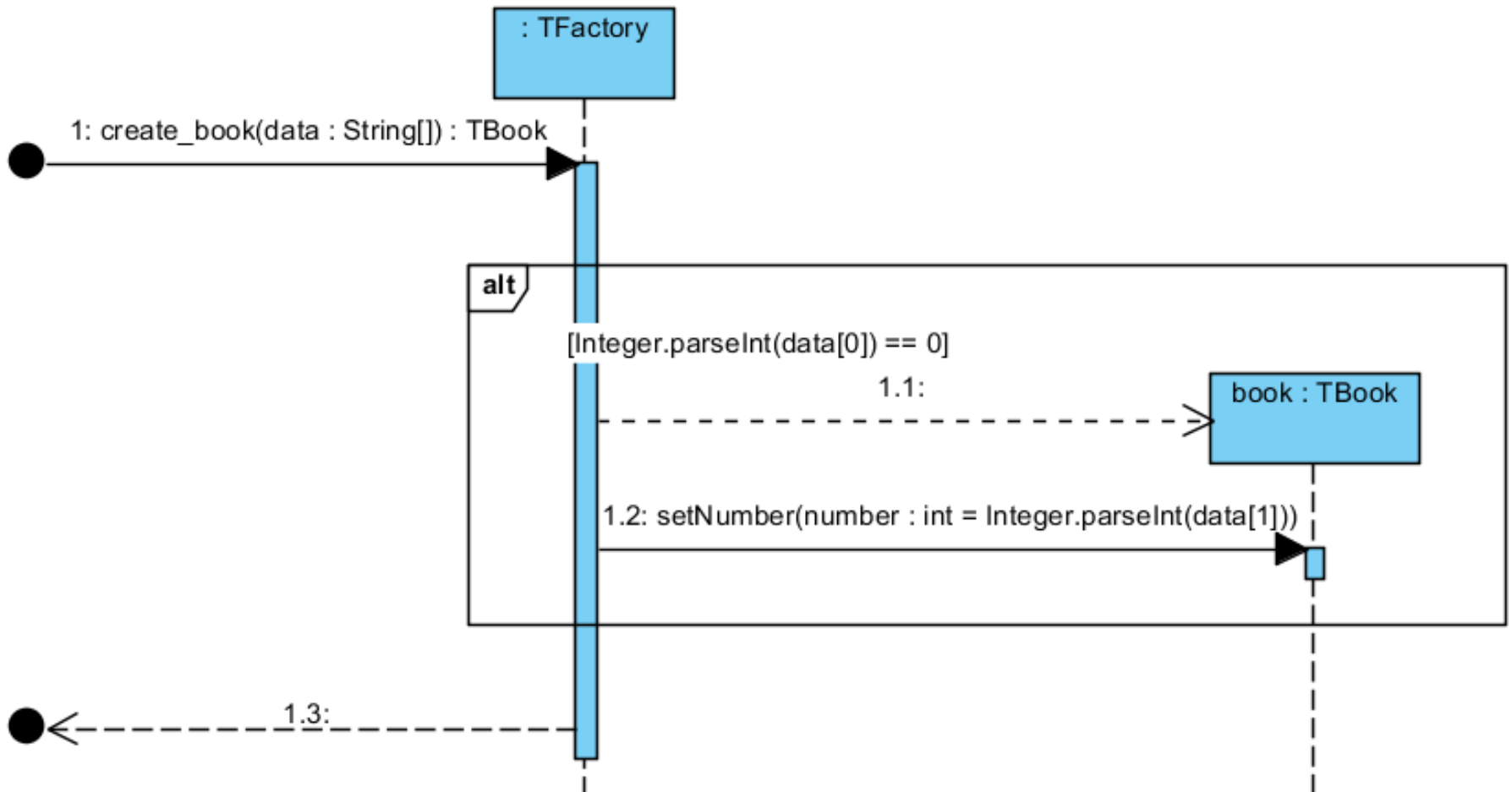

2.5. Diagram sekwencji metody search_book klasy TTitle_book



2.6. Kod metody: search_book klasy TTitle_book

```
public TBook search_book(TBook book) {  
    int idx;  
    if ((idx = mBooks.indexOf(book)) != -1) {  
        book = (TBook) mBooks.get(idx);  
        return book;  
    }  
    return null;  
}
```

2.7. Diagram sekwencji metody create_book klasy TFactory



2.8. Kod metody: create_book klasy TFactory

```
public TBook create_book(String data[]) {  
    TBook book = null;  
    switch (Integer.parseInt(data[0])) //what_book_type  
    {  
        case 0:  
            book = new TBook(); //TBook object for persisting  
            book.setNumber(Integer.parseInt(data[1]));  
            break;  
    }  
    return book;  
}
```

2.9./2.9.1. Przykład kodu sprawdzającego działanie dodawanie tytułów książki oraz książek

```
public static void main(String t[]) {  
    TFacade ap = new TFacade();  
    String t1[] = {"1", "Author1", "Title1", "ISBN1", "Publisher1"};  
    String t2[] = {"1", "Author2", "Title2", "ISBN2", "Publisher2"};  
    String t3[] = {"1", "Author3", "Title3", "ISBN3", "Publisher3"};  
    ap.add_title_book(t1);  
    ap.add_title_book(t2);  
    ap.add_title_book(t2);  
    ap.add_title_book(t3);  
    String lan = ap.getmTitle_books().toString();  
    System.out.println(lan);  
    String d1[] = {"0", "ISBN1"};  
    String d2[] = {"0", "ISBN2"};  
    String d3[] = {"0", "ISBN5"};  
    String tr1[] = {"0", "1"};  
    String tr2[] = {"0", "2"};
```

2.9.2. Przykład kodu sprawdzającego działanie dodawanie tytułów książki oraz książek

```
String d1[] = {"0", "ISBN1"};
String d2[] = {"0", "ISBN2"};
String d3[] = {"0", "ISBN5"};
String tr1[] = {"0", "1"};
String tr2[] = {"0", "2"};
ArrayList<String> pom = ap.add_book(d1, tr1);
if (pom != null) { System.out.print(pom);    }
pom = ap.add_book(d2, tr1);
if (pom != null) { System.out.print(pom);    }
pom = ap.add_book(d2, tr1);
if (pom != null) { System.out.print(pom);    }
pom = ap.add_book(d2, tr2);
if (pom != null) { System.out.print(pom);    }
pom = ap.add_book(d3, tr2);
if (pom != null) { System.out.print(pom);    }
}
```

2.9.2. Przykład kodu sprawdzającego działanie dodawanie tytułów książki oraz książek – wynik działania dodawania tytułów książek i książek

```
[  
Title: Title1 Author: Author1 ISBN: ISBN1 Publisher: Publisher1,  
Title: Title2 Author: Author2 ISBN: ISBN2 Publisher: Publisher2,  
Title: Title3 Author: Author3 ISBN: ISBN3 Publisher: Publisher3]
```

```
[  
Title: Title1 Author: Author1 ISBN: ISBN1 Publisher: Publisher1 Number: 1][  
Title: Title2 Author: Author2 ISBN: ISBN2 Publisher: Publisher2 Number: 1][  
Title: Title2 Author: Author2 ISBN: ISBN2 Publisher: Publisher2 Number: 1,  
Title: Title2 Author: Author2 ISBN: ISBN2 Publisher: Publisher2 Number: 2]
```