

Praktyczna ewaluacja umiejętności. Badanie poprawności kodu.

Prowadzący

Dr inż. Zofia Kruczkiewicz

Forma zajęć - laboratorium		Liczba godzin
La1	Wprowadzenie do laboratorium. Zasady obowiązujące na zajęciach. Wprowadzenie do narzędzi wykorzystywanych podczas laboratorium	2
La2	Praktyczna ewaluacja umiejętności. Badanie poprawności kodu	2
La3-La4	Specyfikowanie wymagań – przypadki użycia	4
La5-La7	Projektowanie oprogramowania – diagramy klas i in.	6
La8-La9	Implementacja w oparciu o wzorce projektowe	4
La10-La11	Sprawdzanie poprawności oprogramowania	4
La12	Tworzenie dokumentacji systemu	2
La13-La14	Refaktoryzowanie systemu	4
La15	Zaliczenie	2
	Suma godzin	30

Sposób przekazywania prac laboratoryjnych

Wybrane wskazówki dotyczące dostarczania materiałów do oceny:

1. Materiały, związane z bieżącym laboratorium, można dokończyć później i wysłać przed następnym laboratorium (2 dni wcześniej) na adres: zofia.kruczkiewicz@pwr.wroc.pl. Należy wysyłać listy korzystając z poczty studenckiej Politechniki.
2. Poprawiane materiały należy wysłać w terminie nie przekraczającym 2 tygodnie – poprawa może być obowiązkowa (prowadząca do poprawy oceny) lub nieobowiązkowa (też w celu poprawy oceny).
3. Do jednego załącznika listu należy dodać wszystkie materiały, należące do laboratoriów (bieżącego i poprawianych), pakując je do formatu zip lub rar. np., numerlab_numerindeksu.zip, gdzie słowo „numer indeksu” należy zastąpić konkretnym numerem indeksu oraz słowo „numerlab” numerem laboratorium
4. List powinien być wysłany również do członka grupy dwuosobowej.

1. Uruchomienie programu Library_1IO - należy pobrać spakowany program ze strony: http://zofia.kruczkiewicz.staff.iar.pwr.wroc.pl/wyklady/INP002017/Lab2_Library_1IO_2019.rar, otworzyć w środowisku NetBeans 8.2 i uruchomić program

Lab2_Library_1IO - NetBeans IDE 8.2

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Search (Ctrl+I)

Projects Files Services

Lab2_Library_1IO

- Source Packages
 - sub_business_tier
 - TFacade.java
 - TFactory.java
 - sub_business_tier.entities
 - TBook.java
 - TTitle_book.java
 - TTitle_book_on_tape.java
- Test Packages
- Libraries
- Test Libraries

main - Navigator

Members

TFacade

- TFacade()
- Print_books() : String
- Print_title_books() : String
- Search_book(String[] data1, String[] data2) : String
- Search_title_book(String[] data) : ArrayList<String>
- add_book(String[] data1, String[] data2) : ArrayList<String>
- add_title_book(String[] data) : String
- getmTitle_books() : List<TTitle_book>
- gettitle_books() : Object[][]
- main(String[] t)
- search_title_book(TTitle_book title_book) : TTitle_book
- setmTitle_books(List<TTitle_book> title_books)
- mTitle_books : List<TTitle_book>

Client.java TFacade.java TTitle_book_on_tape.java Facade.j...

Source History

sub_business_tier.TFacade main if (searched_book_data != null) else

Test Results Output

IO_W11 - C:\Studia\IO_W11 Lab2_Library_1IO (run)

```
run:
{
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 Actor: Actor1,
Title: Title2 Author: Author2 ISBN: ISBN2 Publisher: Publisher2 Actor: Actor2,
Title: Title4 Author: Author4 ISBN: ISBN4 Publisher: Publisher4 Actor: Actor4}
{
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][
Title: Title1 Author: Author1 ISBN: ISBN1 Publisher: Publisher1 Actor: Actor1 Number: 1][
Title: Title4 Author: Author4 ISBN: ISBN4 Publisher: Publisher4 Actor: Actor4 Number: 2]

Titles of book
[Publisher1, ISBN1, Title1, Author1, ]
[Publisher2, ISBN2, Title2, Author2, ]
[Publisher3, ISBN3, Title3, Author3, ]
[Publisher1, ISBN1, Title1, Author1, Actor1]
[Publisher2, ISBN2, Title2, Author2, Actor2]
[Publisher4, ISBN4, Title4, Author4, Actor4]

Books
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: 2
Title: Title1 Author: Author1 ISBN: ISBN1 Publisher: Publisher1 Actor: Actor1 Number: 1
Title: Title4 Author: Author4 ISBN: ISBN4 Publisher: Publisher4 Actor: Actor4 Number: 2

Searching of a title with its books
No such title

Searching of a book of a title
No such book
BUILD SUCCESSFUL (total time: 0 seconds)
```

2. Należy określić, które fragmenty kodu metody main generują wynik widoczny w oknie Output np.

Fragment kodu metody main

```
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"};
String t4[] = {"3", "Author1", "Title1", "ISBN1", "Publisher1", "Actor1"};
String t5[] = {"3", "Author2", "Title2", "ISBN2", "Publisher2", "Actor2"};
String t6[] = {"3", "Author4", "Title4", "ISBN4", "Publisher4", "Actor4"};
ap.add_title_book(t1);
ap.add_title_book(t2);
ap.add_title_book(t2);
ap.add_title_book(t3);
ap.add_title_book(t4);
ap.add_title_book(t5);
ap.add_title_book(t5);
ap.add_title_book(t6);
String lan = ap.getmTitle_books().toString();
System.out.println(lan);
```

Wynik **System.out.println(lan)**; generowany w okienku Output

```
[
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 Actor: Actor1,
Title: Title2 Author: Author2 ISBN: ISBN2 Publisher: Publisher2 Actor: Actor2,
Title: Title4 Author: Author4 ISBN: ISBN4 Publisher: Publisher4 Actor: Actor4]
```

2 (cd). Należy określić, które fragmenty kodu metody main generują wynik widoczny w oknie Output np.

Kolejny fragment kodu metody main

```
ArrayList<String> pom = ap.add_book(d1, tr1);  
if (pom != null) {  
    System.out.print(pom);  
}
```

Wynik `System.out.println(pom)`; generowany w okienku Output

```
[  
Title: Title1 Author: Author1 ISBN: ISBN1 Publisher: Publisher1 Number: 1]
```

Należy dokończyć analizę kodu w pliku tekstowym, wykonując w podobny sposób analizę wyników działania programu

3. Należy wykonać kody metod `public ArrayList<String> Search_title_book(String data[])` w klasie TFacade `public String Search_book(String data1[],String data2[])` w klasach TFacade i `public String Search_book(String data[])` TTitle_book

The screenshot displays an IDE interface for a Java project named 'Lab2_Library_1IO'. The left sidebar shows the project structure with packages 'sub_business_tier' and 'sub_business_tier.entities', and classes 'TFacade.java', 'TFactory.java', 'TBook.java', 'TTitle_book.java', and 'TTitle_book_on_tape.java'. The 'Members' view for 'TFacade' lists several methods, with 'Search_title_book(String[] data) : ArrayList<String>' highlighted. The main editor shows the 'Test Results' for a run of 'Lab2_Library_1IO', displaying the following output:

```
run:
[
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 Actor: Actor1,
Title: Title2 Author: Author2 ISBN: ISBN2 Publisher: Publisher2 Actor: Actor2,
Title: Title4 Author: Author4 ISBN: ISBN4 Publisher: Publisher4 Actor: Actor4]
[
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][
Title: Title1 Author: Author1 ISBN: ISBN1 Publisher: Publisher1 Actor: Actor1 Number: 1][
Title: Title4 Author: Author4 ISBN: ISBN4 Publisher: Publisher4 Actor: Actor4 Number: 2]

Titles of book
[Publisher1, ISBN1, Title1, Author1, ]
[Publisher2, ISBN2, Title2, Author2, ]
[Publisher3, ISBN3, Title3, Author3, ]
[Publisher1, ISBN1, Title1, Author1, Actor1]
[Publisher2, ISBN2, Title2, Author2, Actor2]
[Publisher4, ISBN4, Title4, Author4, Actor4]

Books
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: 2
Title: Title1 Author: Author1 ISBN: ISBN1 Publisher: Publisher1 Actor: Actor1 Number: 1
Title: Title4 Author: Author4 ISBN: ISBN4 Publisher: Publisher4 Actor: Actor4 Number: 2

Searching of a title with its books
[
Title: Title4 Author: Author4 ISBN: ISBN4 Publisher: Publisher4 Actor: Actor4 Number: 2]

Searching of a book of a title
Title: Title2 Author: Author2 ISBN: ISBN2 Publisher: Publisher2 Number: 1

BUILD SUCCESSFUL (total time: 0 seconds)
```

3 (cd). Należy wykonać kod metody **public ArrayList<String> Search_title_book(String data[])** w klasie TFacade.

Należy zastosować podejście podobne **jak przy dodawaniu obiektów typu TTitle_book**

Obecna postać kodu metody **Search_title_book**

```
public ArrayList<String> Search_title_book(String data[]) {  
    return null;  
}
```

wywołanej w metodzie main klasy TFacade w następujący sposób:

```
System.out.println("\nSearching of a title with its books");  
ArrayList<String> searched_title_data=ap.Search_title_book(t6);  
if (searched_title_data!=null)  
    System.out.print(searched_title_data + "\n");  
else  
    System.out.print("No such title" + "\n");
```

Obecnie wynik w okienku Output jest następujący:

```
Searching of a title  
No such title
```

Po wykonaniu kodu metody **Search_title_book** wynik w okienku Output powinien być następujący:

```
Searching of a title with its books
```

```
[
```

```
Title: Title4 Author: Author4 ISBN: ISBN4 Publisher: Publisher4 Actor: Actor4 Number: 2]
```


3 (cd). Należy wykonać kod metody **public String Search_book(String data1[],String data2[])** w klasie TFacade, która po wyszukaniu obiektu typu TTitle_book wywołuje metodę **public String Search_book(String []data)** z klasy TTitle_book w celu wyszukania obiektu typu TBook. Należy zastosować podejście podobne **jak przy dodawaniu obiektów typu TBook**

Obecna postać kodu metody **Search_book**

```
public String Search_book(String data1[], String data2[]) {  
    return null;  
}
```

wywołanej w metodzie main klasy TFacade w następujący sposób:

```
System.out.print("\nSearching of a book of a title");  
String searched_book_data=ap.Search_book(d2, tr1);  
if (searched_book_data!=null)  
    System.out.println(searched_book_data + "\n");  
else  
    System.out.println("\n"+"No such book");
```

Obecnie wynik w okienku Output jest następujący:

```
Searching of a book of a title  
No such book
```

Po wykonaniu kodu metody **Search_book** wynik w okienku Output powinien być następujący:

```
Searching of a book of a title  
Title: Title2 Author: Author2 ISBN: ISBN2 Publisher: Publisher2 Number: 1
```

4. Należy rozszerzyć i/lub zmodyfikować kod programu

1. Należy rozszerzyć kod metody main w celu rozszerzenia przypadków testowania działania funkcji programu.
2. Należy dokonać ewentualnej modyfikacji kodu wyjaśniając w komentarzu zmienianego kodu powód zmiany.
3. Należy zmodyfikować nazwy klas, metod itp