

Tworzenie informacji przydatnej przy tworzeniu dokumentacji Laboratorium 12

Zofia Kruczkiewicz

Tworzenie dokumentacji dla zespołu wytwarzającego oprogramowanie. Wykorzystanie narzędzi

Cel laboratorium:

Definiowanie danych przydatnych przy tworzeniu dokumentacji

Należy w polach *Description* pozycji *Open Specification...* podanych dalej elementów w projekcie UML (wykonanym podczas lab3-10) - dodać informację wg podanych wskazówek:

Podanie informacji w polach **Description**, dostępnych na formularzu uruchomionym z pozycji **Open Specification...**, powiązany z otwartym diagramem lub elementem otwartego diagramu

1. Należy wybrać **diagram klas**, który należy do modelu danych zdefiniowanym podczas lab.5-7, zawierający logikę biznesową. Po kliknięciu na powierzchnię diagramu wybrać pozycję **Open Specification...**, a następnie pole **Description**. Należy tam wpisać informację dotyczącą roli diagramu klas w projekcie np wg. instrukcji do lab.5-7:

http://zofia.kruckiewicz.staff.iiar.pwr.wroc.pl/wyklady/INP002017/Lab_INP002017_5_2.pdf

str. 10, gdzie podano źródło identyfikacji klas, czyli przyjęta postać diagramu przypadków użycia

2. Na diagramie klas wybrać kolejno klasy i w ich polach **Description** wpisać informację dotyczącą przyczyny identyfikacji klasy, jej atrybutów i związków z innymi klasami. Należy podać nazwy przypadków użycia, których scenariusze stanowią źródłową informację dotyczącą identyfikacji danej klasy oraz jej właściwości – na podstawie podanej wyżej instrukcji do lab. 5-7, str. 11-15.

Podanie informacji w polach **Description**, dostępnych na formularzu uruchomionym z pozycji **Open Specification...**, powiązany z otwartym diagramem lub elementem otwartego diagramu - cd

3. Należy wybrać **diagram aktywności**. W polu **Description** tego diagramu wpisać informację dotyczącą informacji źródłowej związanej z modelowanymi procesami w poszczególnych torach, na podstawie instrukcji do lab. 8:

http://zofia.kruczkiewicz.staff.iiar.pwr.wroc.pl/wyklady/INP002017/Lab_INP002017_8.pdf

- 3.1. Który scenariusz przypadków użycia jest modelowany na diagramie aktywności
- 3.2. Jakie diagramy sekwencji stanowią źródłową informację związaną z modelowanym procesem w każdym z torów.
- 3.3. Należy podać, które wiadomości typu **Call Message** są odwzorowane na akcje,
- 3.4. Należy podać, które zbiory wiadomości typu **Call Message** są odwzorowane na aktywności,
- 3.5. Należy podać, które diagramy sekwencji są odwzorowane jedynie na aktywności w poszczególnych torach diagramu. Należy wyjaśnić, dlaczego nie są przedstawiane w postaci torów.

Podanie informacji w polach **Description**, dostępnych na formularzu uruchomionym z pozycji **Open Specification...**, powiązany z otwartym diagramem lub elementem otwartego diagramu - cd

3. Należy wybrać **diagram stanów**. W polu **Description** tego diagramu wpisać informację dotyczącą informacji źródłowej związanej z modelowanymi procesami obiektu wybranej klasy - w poszczególnych przejściach i stanach, na podstawie instrukcji do lab. 9:

http://zofia.kruczkiewicz.staff.iar.pwr.wroc.pl/wyklady/INP002017/Lab_INP002017_9.pdf

3.1. Które fragmenty scenariuszy przypadków użycia są modelowane na diagramie stanów wybranej klasy

3.2. Jakie diagramy sekwencji stanowią źródłową informację związaną z modelowanym zdarzeniem, jego dozorem i jego akcjami w każdym z fragmentów diagramu stanów, zawierającym podzbiór przejść i stanów. Należy również podać, z którego diagramu sekwencji został modelowany diagram sekwencji wywołany. Jest to informacja o miejscu wystąpienia zdarzenia na obiekcie wybranej klasy.

Informacje pomocnicze przy tworzeniu tekstu w polu **Description**

1. Wstawianie do odwołań do elementów modelu np w opisie klas przy odwołaniu do przypadków użycia, lub w opisach diagramu aktywności i diagramu stanów w odwołaniach do diagramów sekwencji – po kliknięciu na klasę należy wybrać pozycję **Add Model Element...**

The screenshot displays the Visual Paradigm Community Edition interface. The main window shows a UML class diagram with several classes: TitleBookRead, Client, TitleBook, Reservation, and Book. The TitleBook class is highlighted, and its specification dialog is open. The dialog has tabs for General, Attributes, Operations, and References. The General tab is active, showing the class name 'TitleBook', parent 'entities', and visibility 'public'. The Description field contains the text 'Scenariusze przywoadków użycia' followed by '1) PU'. A context menu is open over the text, with 'Add Model Element...' selected. The UML diagram in the background shows the following classes and relationships:

- TitleBookRead**: Properties: publisher: String, ISBN: String, title: String, author: String. Operations: books: Book = new ArrayList().
- Client**: Properties: reservationNumber: int, name: String. Operations: reservations: List<Reservation>, searchBook(book: Book): Book, searchFreeBook(date: LocalDate): boolean, toString(): String.
- TitleBook**: Properties: publisher: String, ISBN: String, title: String, author: String, books: Book = new ArrayList(). Operations: TitleBook(), getBooks(): List<Book>, setBooks(books: List<Book>): void, hashCode(): int, getActor(): String, getActor(val: String): void, getBooksModel(): ArrayList<String>, equals(obj: Object): boolean, addBook(data: String[]): ArrayList<String>, searchBook(book: Book): Book, searchFreeBook(date: LocalDate): boolean, toString(): String, toString(): String[].
- Reservation**: Properties: book: Book, client: Client, number: int, date: LocalDate. Operations: Reservation(), getReservations(): List<Reservation>, setReservations(reservations: List<Reservation>): void, hashCode(): int, equals(obj: Object): boolean, toString(): String, addReservation(reservation: Reservation): void, isFree(date: LocalDate): boolean.
- Book**: Properties: number: int, titleBook: TitleBook, reservations: Reservation = new ArrayList(). Operations: Book(), getReservations(): List<Reservation>, setReservations(reservations: List<Reservation>): void, hashCode(): int, equals(obj: Object): boolean, toString(): String, addReservation(reservation: Reservation): void, isFree(date: LocalDate): boolean.

Relationships: TitleBookRead has a directed association to TitleBook. Client has a directed association to TitleBook. Reservation has a directed association to Book. Book has a directed association to Reservation. There are also self-associations on Client and Book.

Informacje pomocnicze przy tworzeniu tekstu w polu **Description** (cd)

2. W formularzu **Select Model Elements** należy wyszukać i zaznaczyć właściwe przypadki użycia

The screenshot displays the Rational UML modeling environment. The main window shows a UML Class Diagram with classes: `TitleBookRead`, `Client`, `TitleBook`, and `Book`. The `TitleBook` class is highlighted, and its specification is shown in the **Class Specification** dialog box.

The **Class Specification** dialog box has the following fields:

- Name: `TitleBook`
- Parent: `entities`
- Visibili...: `public`
- Description: Scenariusze przywoadków użycia
1) PU

A context menu is open over the description field, with the following options:

- Add Model Element...
- Add Property Value...
- Add Diagram...
- Add Variable
- Insert Hyperlink...
- Open URL
- Cut
- Copy
- Paste
- Paste without format
- Undo
- Redo
- Format

The **Select Model Elements** dialog box is also open, showing a tree view of the model elements. The following elements are selected:

- Dodaj_Ksiazke
- Dodaj_Tytul_Ksiazki
- Modyfikacja_Ksiazki
- Usun_Ksiazke
- Rezerwacja

The `TitleBook` class specification is visible in the background, showing the following code:

```
<<Property>> -publisher : String
<<Property>> -ISBN : String
<<Property>> -title : String
<<Property>> -author : String
~books : Book = new ArrayList()
<<Property>> -book : Book

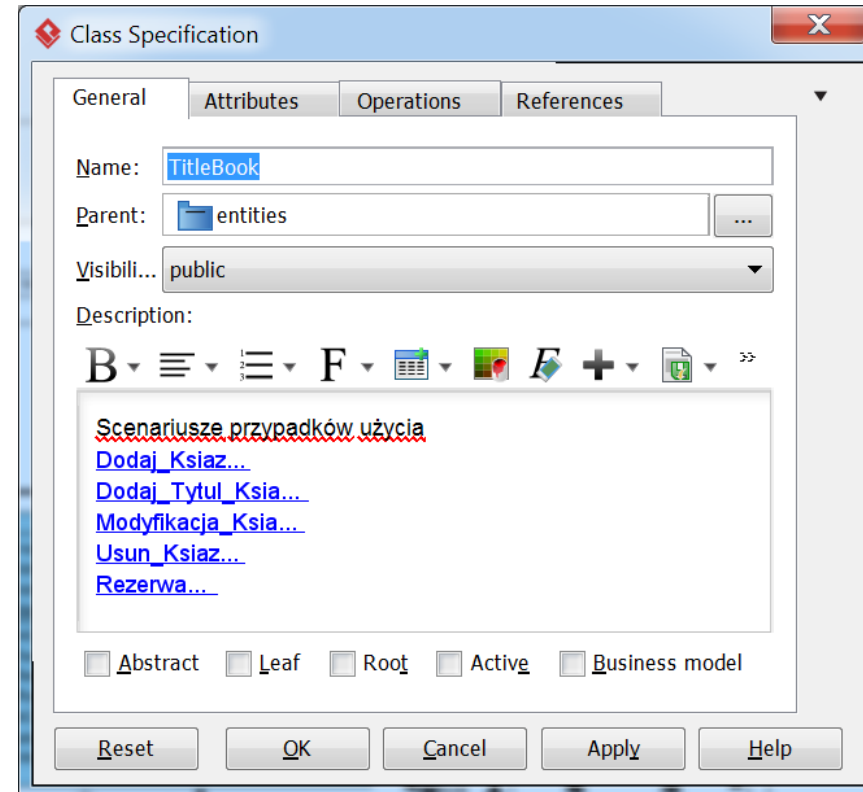
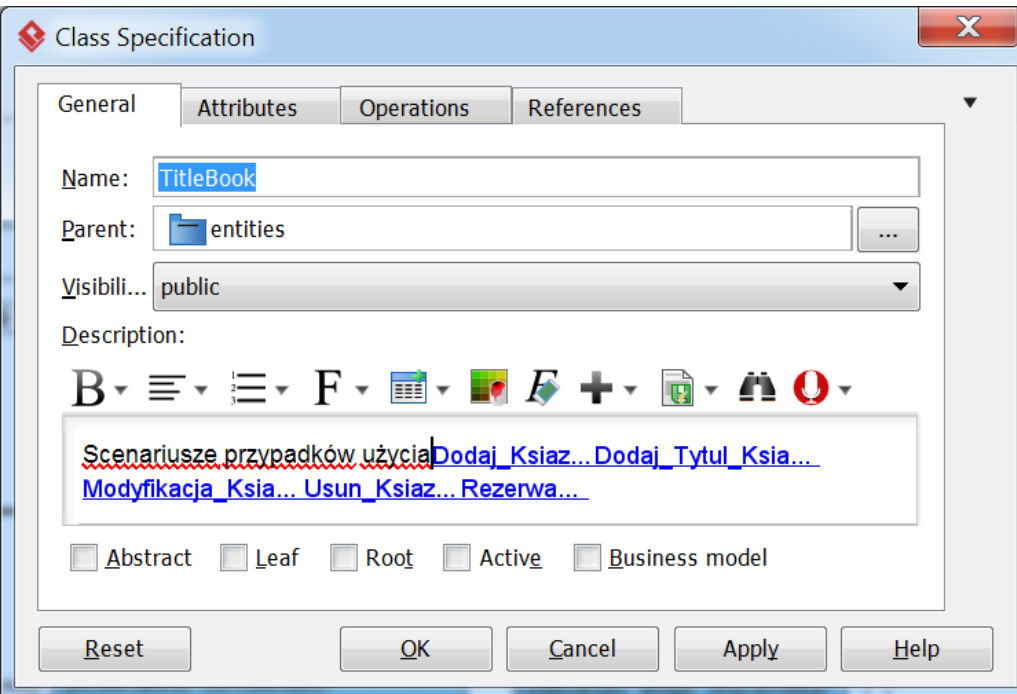
+TitleBook()
+getBooks() : List<Book>
+setBooks(books : List<Book>) : void
+hashCode() : int
+getActor() : String
+setActor(val : String) : void
+booksModel() : ArrayList<String>
+is(obj : Object) : boolean
+Book(data : String[]) : ArrayList<String>
+chBook(book : Book) : Book
+chFreeBook(date : LocalDate) : boolean
+ing() : String
+ing() : String[]

Book
number : int
eBook : TitleBook
reservation = new ArrayList()

() : List<Reservation>
(reservations : List<Reservation>) : void
act) : boolean
}
(reservation : Reservation) : void
calDate) : boolean
```

Informacje pomocnicze przy tworzeniu tekstu w polu **Description** (cd)

3. W ostatnim kroku można uporządkować uzyskane odwołania do scenariuszy przypadków użycia.



Informacje pomocnicze przy tworzeniu tekstu w polu **Description** (cd)

- Po wyborze nazwy elementu modelu (tutaj nazwy przypadku użycia) można uzyskać dostęp do scenariusza przypadku użycia

The screenshot displays the Visual Paradigm Community Edition interface. The main window shows a UML Class Diagram with a 'Factory' class and a 'Facade' class. The 'Class Specification' dialog is open, showing the 'General' tab for the 'TitleBook' class. The 'Description' field contains the following text:

Scenariusze przypadków użycia

- [Dodaj_Ksiaz...](#)
- [Dodaj_Tytul_Ksia...](#)
- [Modyfikacja_Ksia...](#)
- [Usun_Ksiaz...](#)
- [Rezerwa...](#)

Below this, a yellow box highlights the 'Scenariusz' (Scenario) for the 'Rezerwa...' use case:

Scenariusz

- Należy podać atrybuty tytułu: ISBN jako obowiązkowa i ewentualnie aktora. Jeśli książka jest nagrana, numer książki, datę rezerwacji oraz numer karty bibliotecznej klienta
- Należy wykonać scenariusz **PU Operacje_Na_Ksiazkach**
- W przypadku braku książki należy zakończyć PU
- Po wyszukaniu książki sprawdza się za pomocą **PU Wyszukaj_Rezerwacje** czy dany egzemplarz książki nie jest zarezerwowany w podanym terminie. Jeśli tak, należy zakończyć PU
- Należy wyszukać klienta wg numeru karty bibliotecznej za pomocą **PU Wyszukaj_Klienta** jeśli nie istnieje, należy zakończyć PU.
- Należy wykonać rezerwację, umieszczając na niej dane wyszukanej książki, dane klienta oraz datę rezerwacji. Dane rezerwacji należy umieścić w zbiorze rezerwacji zarezerwowanej książki i w zbiorze rezerwacji znalezionego klienta.

Informacje pomocnicze przy tworzeniu tekstu w polu **Description** (cd)

5. W polu **Description** diagramów aktywności i diagramu stanów można również „wkleić” dostęp do diagramów sekwencji.

The screenshot displays the Visual Paradigm Community Edition interface. The main workspace shows an activity diagram for 'addBook' with three swimlanes: GUI, Facade, and TitleBook. The GUI swimlane contains 'input data' and a thick black bar. The Facade swimlane contains 'data1 of searched title', 'create titleBook for searching', 'Search titleExist', 'data2 of new book', and a decision diamond. The TitleBook swimlane contains 'data2 of new book', 'create new book', 'Search Book', a decision diamond, 'Add new book', 'Prepare model of books', and 'Model of books'. The diagram shows data flows and control flows between these elements.

Two configuration windows are overlaid on the diagram:

- Activity Diagram Specification:** This window shows the 'General' tab with the following fields:
 - Diagram name: `addBook`
 - Parent model: `<No parent model>`
 - Zoom ratio: `90%`
 - Diagram background: `White`
 - Description: (empty)
- Select Model Elements:** This window shows a list of model elements with the following selected items:
 - `subbusnesstier.entities.TitleBook.addBook(String)`
 - `subbusnesstier.Facade.addBook(String, String)`

Informacje pomocnicze przy tworzeniu tekstu w polu **Description** (cd)

6. Dostęp do widoku diagramu sekwencji jest jednak możliwy w wersji *Model Explorer* narzędzia *Visual Paradigm*.

The image shows two instances of the 'Activity Diagram Specification' dialog box. The left instance shows the 'Description' field with the text: `subbusinessstier.entities.TitleBook.addBook(String, String)` and `subbusinessstier.Facade.addBook(String, String)`. The right instance shows the same dialog box with a context menu open over the text. The menu options are: Open Specification..., Show View..., Add Variable, Insert Hyperlink..., Open URL, Cut, Copy, Paste, Paste without format, Undo, Redo, and Format. A red arrow points from the 'Show View...' option in the menu to the 'Show View' dialog box.

Activity Diagram Specification

General | Grid Setting | References | Project Management | Comments

Diagram name:

Parent model:

Zoom ratio:

Diagram background:

Description:

B **F**

[subbusinessstier.entities.TitleBook.addBook\(String, String\)](#)
[subbusinessstier.Facade.addBook\(String, String\)](#)

Auto fit shapes size Follow parent

Reset OK Cancel Apply Help

Show View

Model: subbusinessstier.Facade.addBook(String, String)

Master view is shown on Model Explorer

"subbusinessstier.Facade.addBook(String, String)" has no view

Go to Vi... Close

Informacje pomocnicze przy tworzeniu tekstu w polu **Description** (cd)

Uwaga

Można również skorzystać z innych elementów edytora pola ***Description***