

Platformy Technologiczne

Laboratorium 4

Java: JAXB

Należy zaimplementować aplikację pozwalającą na zarządzanie katalogiem przechowywanym w pliku xml. Struktura katalogu zostanie podana przez prowadzącego na zajęciach.

Przykład:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<catalog xmlns="http://www.eti.pg.gda.pl/kask/pt/catalog">
    <book title="Na ostrzu noża" pages="300"/>
    <bool title="Norweski dziennik" pages="200"/>
</catalog>
```

Encjami wykorzystywanymi w katalogu będą dwie klasy. Korzeń (np. *catalog*) przechowujący kolekcję elementów (np. *book*).

Wczytywanie/zapisywanie z/do pliku xml realizuje się przez obiekty *Unmarshaller* i *Marshaller* stworzone za pomocą obiektu *JAXBContext*.

Przykład:

```
JAXBContext context = JAXBContext.newInstance("pakiet_z_klasami_encyjnymi");
Unmarshaller unmarshaller = context.createUnmarshaller();
Catalog catalog = (Catalog) unmarshaller.unmarshal(file);

JAXBContext context = JAXBContext.newInstance("pakiet_z_klasami_encyjnymi");
Marshaller marshaller = context.createMarshaller();
marshaller.marshal(catalog, file);
```

Wszystkie klasy encyjne powinny znajdować się w jednym pakiecie wraz z klasą *ObjectFactory* i plikiem *package-info.java*. Klasa *ObjectFactory* musi dostarczać metody wytwórcze dla wszystkich klas encyjnych, natomiast plik *package-info.java* powinien dostarczać powiązanie pakietu z nazwą schemy.

Przykład:

```
@XmlRegistry
public class ObjectFactory {

    public ObjectFactory() {
    }

    public Tower createCatalog() {
        return new Catalog();
    }
}
```

```
@XmlSchema(namespace = "http://www.eti.pg.gda.pl/kask/pt/catalog", elementFormDefault =
XmlNsForm.QUALIFIED)
package pl.gda.pg.eti.kask.pt.catalog;
```

Aplikacja musi korzystać z komponentu `TableView` z JavaFX. Zarówno kolumny jak i sposób ich wyświetlania można zdefiniować w pliku `fxml` korzystając z odpowiednich obiektów ***TableColumn*** i ***PropertyValueFactory*** oraz odpowiednich własności ***columns*** i ***cellValueFactory***.

Przykład:

```
<TableView fx:id="tableView">
    <columns>
        <TableColumn fx:id="titleColumn">
            <cellValueFactory>
                <PropertyValueFactory property="title"/>
            </cellValueFactory>
        </TableColumn>
    </columns>
</TableView>
```

Wszystkie atrybuty (np. ***title***) muszą być edytowalne z poziomu tabelki. W tym celu obiekt tabeli musi być ustawiony jako edytowalny a do każdej z kolumn należy podpiąć odpowiednie obiekty ***TableCell*** poprzez własność ***cellFactory***. Jako obiektu ***TableCell*** należy skorzystać ze standardowego ***TextFieldTableCell*** renderującego komórkę tabeli jako zwykłe pole tekstowe. Należy zwrócić uwagę że ustawienie obiektu ***TextFieldTableCell*** pozwala jedynie na edycję zawartości komórki. Należy dopisać odpowiedni handler (klasa implementująca interfejs ***EventHandler***) obsługujący przepisanie nowej wartości do odpowiedniego obiektu.

Przykład:

```
TableView.setEditable(true);

titleColumn.setCellFactory(TextFieldTableCell.forTableColumn());
titleColumn.setOnEditCommit(new EventHandler<CellEditEvent<Book, String>>() {
    @Override
    public void handle(CellEditEvent<Mage, String> t) {
        Book book = t.getRowValue();
        String newTitle = t.getNewValue();
        book.setTitle(newTitle);
    }
});
```

W przypadku pól typu innego niż ***String*** (np. ***int***, ***double***, itd.) należy skorzystać z odpowiedniego konwertera (np. ***IntegerStringConverter***, ***DoubleStringConverter***, itd.).

Przykład:

```
IntegerStringConverter conv = new IntegerStringConverter();
pagesColumn.setCellFactory(TextFieldTableCell.forTableColumn(conv));
```

Aplikacja musi pozwalać (np. za pomocą przycisków) na dodawanie nowej pozycji w katalogu, usuwanie, zapisanie katalogu do pliku oraz jego wczytanie.

Punktacja:

- wczytanie katalogu z pliku xml i wyświetlenie go w tabeli: 1 pkt,
- zapisanie zawartości tabeli do pliku xml: 1 pkt,
- usuwanie elementów z tabeli: 1 pkt,
- dodawanie nowych elementów do tabeli: 1 pkt,
- edycja elementów w tabeli: 1 pkt.