

# Platformy Technologiczne

## Laboratorium 2

### Java: JavaFX

Należy zaimplementować aplikację pełniącą funkcję menadżera plików. Aplikacja powinna pozwalać na wybranie katalogu z systemu plików za pomocą obiektu klasy *FileChooser* a następnie wyświetlić go i jego zawartość rekurencyjnie za pomocą kontrolki *TreeView*. Do wypełnienia kontrolki *TreeView* należy wykorzystać obiekty klasy *TreeItem*. Zarówno *TreeView* jak i *TreeItem* muszą być sparametryzowane klasą obiektów przechowywanych w drzewku. Może to być typ *File* lub własna klasa go opakowująca.

Przykład:

```
@FXML
private TreeView<File> treeView;
```

```
TreeItem<File> root = new TreeItem<>();
treeView.setRoot(root);
```

```
FileChooser fileChooser = new FileChooser();
File file = fileChooser.showOpenDialog(null);
```

Wygląd GUI musi być zdefiniowany w pliku fxml i wczytany przy starcie aplikacji. Wszystkie akcje udostępniane przez aplikację muszą zostać zrealizowane w kontrolerze powiązanym z plikiem fxml.

Przykład:

```
public class MainController implements Initializable {
    @Override
    public void initialize(URL url, ResourceBundle rb) {
    }
}
```

```
<AnchorPane id="AnchorPane" xmlns:fx="http://javafx.com/fxml"
    xmlns="http://javafx.com/javafx/2.2" fx:controller="MainController">
</AnchorPane>
```

```
public class App extends Application {

    public static void main(String[] args) {
        launch(args);
    }

    @Override
    public void start(Stage stage) throws Exception {
        Parent root = FXMLLoader.load(getClass().getResource("main.fxml"));
        Scene scene = new Scene(root, 600, 400);
        stage.setScene(scene);
        stage.show();
    }
}
```

Elementy drzewka muszą być wyświetlone według instrukcji prowadzącego (np. sama nazwa pliku, prawa dostępu, data modyfikacji). Przy tym zadaniu nie trzeba wykorzystywać wiązań JavaFX a jedynie zwykłe własności. Definicja wyglądu elementów drzewka wymaga stworzenia fabryki komórek implementującej interfejs **Callback** deklarujący metodę **call** zwracającą obiekt **TreeCell**.

Przykład:

```
private class FileCell extends TreeCell<File> {  
  
    @Override  
    protected void updateItem(File file, boolean empty) {  
        super.updateItem(file, empty);  
        if (file != null) {  
            setText(file.getName());  
        } else {  
            setText(null);  
        }  
    }  
}
```

```
private class FileCellFactory implements Callback<TreeView<File>, TreeCell<File>> {  
    @Override  
    public TreeCell<File> call(TreeView<File> p) {  
        return new FileCell();  
    }  
}
```

```
treeView.setCellFactory(new FileCellFactory());
```

Prócz możliwości wczytania katalogu aplikacja musi pozwalać na usunięcie/dodanie elementu z/do drzewka co wiąże się z odpowiednią modyfikacją w systemie plików. W przypadku dodawania elementu użytkownik musi mieć wybór czy będzie to plik czy katalog (np. za pomocą różnych przycisków). Tworzenie plików i katalogów można zrealizować za pomocą metod **createNewFile** i **mkdir** udostępnione przez klasę **File**. W celu usunięcia pliku/katalogu można wykorzystać metodę **delete**. W przypadku usuwania katalogu musi on być pusty, dlatego potrzebna będzie metoda rekurencyjna usuwająca wpięrow wszystkie elementy z wybranego katalogu.

Przykład:

```
TreeItem<File> selectedItem = treeView.getSelectionModel().getSelectedItem();  
File selectedFile = selectedItem.getValue();  
  
File newFile = new File(selectedFile, "new_file_name");  
newFile.createNewFile();
```

```
TreeItem<File> selectedItem = treeView.getSelectionModel().getSelectedItem();  
selectedItem.getParent().getChildren().remove(selectedItem);  
selectedItem.getValue().delete();
```

Akcje powinny być dostępne w interfejsie użytkownika za pomocą przycisków podpiętych pod odpowiednie akcje (metody) w kontrolerze.

Przykład:

```
<Button mnemonicParsing="false" onAction="#deleteFileAction" text="Delete file" />
```

```
@FXML
private void deleteFileAction(ActionEvent event) {
}
```

Punktacja:

- wczytanie katalogu i wyświetlenie go i jego zawartości w drzewku: 1 pkt,
- sformatowanie wyświetlania elementów drzewka: 1 pkt,
- usuwanie plików i pustych katalogów z drzewka i systemu plików: 1 pkt,
- usuwanie niepustych katalogów z drzewka i systemu plików: 1 pkt,
- tworzenie nowych plików i katalogów w drzewku i systemie plików: 1 pkt.