

Napisać aplikację „Wybór owoców/warzyw do dostarczenia” .

Aplikacja polega na tym, że użytkownikowi wyświetlane jest okno dialogowe ze zdjęciami 4-ech owoców/warzyw (banany, jabłka, marchewka, czereśnie) oraz czterema polami wyboru. Użytkownik może wybrać dowolną kombinację tych rodzajów owoców lub nie wybrać żadnego. Użytkownik klikając przycisk „Zatwierdź wybór” dokonuje potwierdzenia swojego wyboru. Przykładowy wygląd aplikacji przedstawiają Rys. 1 i Rys. 2.

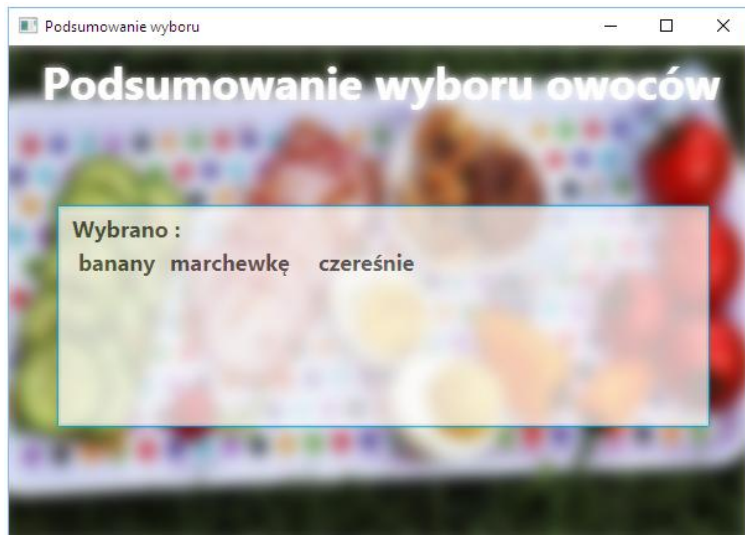


Rys. 1 Przykładowy wygląd aplikacji, użytkownik nie wybrał żadnego rodzaju dostępnych owoców/warzyw.

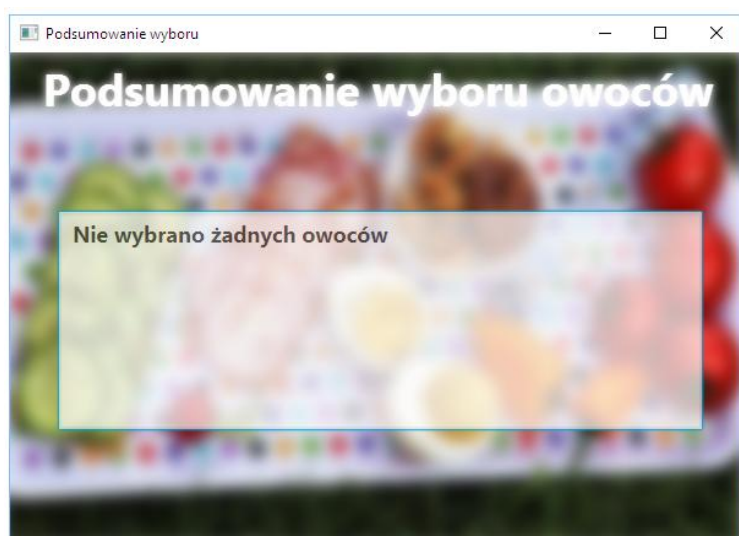


Rys. 2 Przykładowy wygląd aplikacji, użytkownik wybrał 3 rodzaje spośród dostępnych owoców/warzyw – banany, marchewkę, czereśnie.

Po kliknięciu na przycisk „Zatwierdź wybór” okno z możliwością wyboru owoców/warzyw znika a pojawia się okno z informacją podsumowującą dokonany wybór (Rys. 3 i Rys. 4).



Rys. 3 Okno z podsumowaniem dokonanego wyboru w przypadku, gdy użytkownik wybrał którąś z dostępnych kombinacji.



Rys. 4 Okno z podsumowaniem dokonanego wyboru w przypadku, gdy użytkownik nie wybrał żadnego rodzaju z dostępnych owoców.

Do zbudowania aplikacji zastosować **wzorzec projektowy MVC** (ang. Model View Controller – model widok kontroler). Należy stworzyć dwa widoki – pierwszy związany z wyborem dostępnych rodzajów owoców (jak na Rys. 1 i Rys. 2), drugi związany z podsumowaniem dokonanego wyboru (jak na Rys. 3 i Rys. 4). Należy również stworzyć dwie klasy kontrolerów odpowiedzialnych za powiązanie widoków z modelem.

Obrazki owoców oraz obrazek tła w oknie podsumowania wyboru są dostępne w pliku `obrazki_app_owoce.zip`.

Do stworzenia aplikacji wykorzystać JavaFX. Do stworzenia widoków można wykorzystać narzędzie Scene Builder dostępne pod adresem <http://gluonhq.com/products/scene-builder/>. Stworzyć widoki w plikach FXML.

Do zaimportowania plików FXML można wykorzystać instrukcje:

```
Parent root = FXMLLoader.load(getClass().getResource("nazwa_pliku.fxml"));
```

lub instrukcje:

```
FXMLLoader loader = new FXMLLoader();
```

```
loader.setLocation(getClass().getResource("nazwa_pliku.fxml"));
```

```
try {
```

```
    loader.load();
```

```
    } catch (IOException e) {  
        e.printStackTrace();  
    }
```

Obsługa zdarzeń odbywać się może za pomocą kodu:

```
kontrolka.setOnAction(new EventHandler<ActionEvent>() {  
    public void handle(ActionEvent event) {  
        /**obsługa zdarzenia **/  
    }  
});
```

umieszczonego wewnątrz metody **initialize** kontrolera.

Może być konieczne zaimplementowanie m.in. następujących zasobów:

```
import javafx.event.ActionEvent;  
import javafx.event.EventHandler;  
import javafx.fxml.FXML;  
import javafx.fxml.FXMLLoader;  
import javafx.fxml.Initializable;  
import javafx.scene.Parent;  
import javafx.scene.Scene;  
import javafx.scene.control.Button;  
import javafx.scene.control.CheckBox;  
import javafx.scene.control.TextArea;  
import javafx.scene.paint.Color;  
import javafx.stage.Stage;  
import java.io.IOException;  
import java.net.URL;  
import java.util.ResourceBundle;
```

Przykład prostej aplikacji w JavaFX

```
package sample;
```

```
import javafx.application.Application;  
import javafx.scene.Group;  
import javafx.scene.Scene;  
import javafx.scene.paint.Color;  
import javafx.scene.shape.Rectangle;  
import javafx.stage.Stage;
```

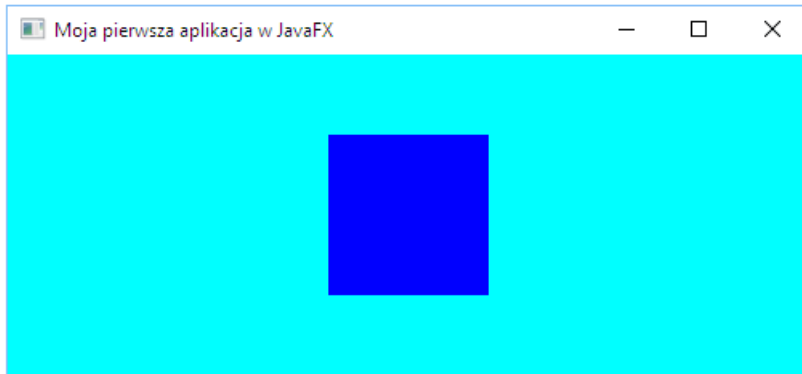
```
public class Main extends Application {
```

```
    public void start(Stage stage) throws Exception{  
        Group root = new Group();  
        stage.setTitle("Moja pierwsza aplikacja w JavaFX");  
        Scene scene = new Scene(root, 500, 200, Color.AQUA);  
        stage.setScene(scene);  
        Rectangle r= new Rectangle(200,50,100,100);  
        r.setFill(Color.BLUE);
```

```
root.getChildren().add(r);
stage.show();
}

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

Efektem działania tej aplikacji będzie następujący widok:



Więcej o JavaFX:

[Informacje ogólne o JavaFX](#)

[Hello World w JavaFX](#)

[JavaFX Scene Graph API \(Application Programming Interface\)](#)

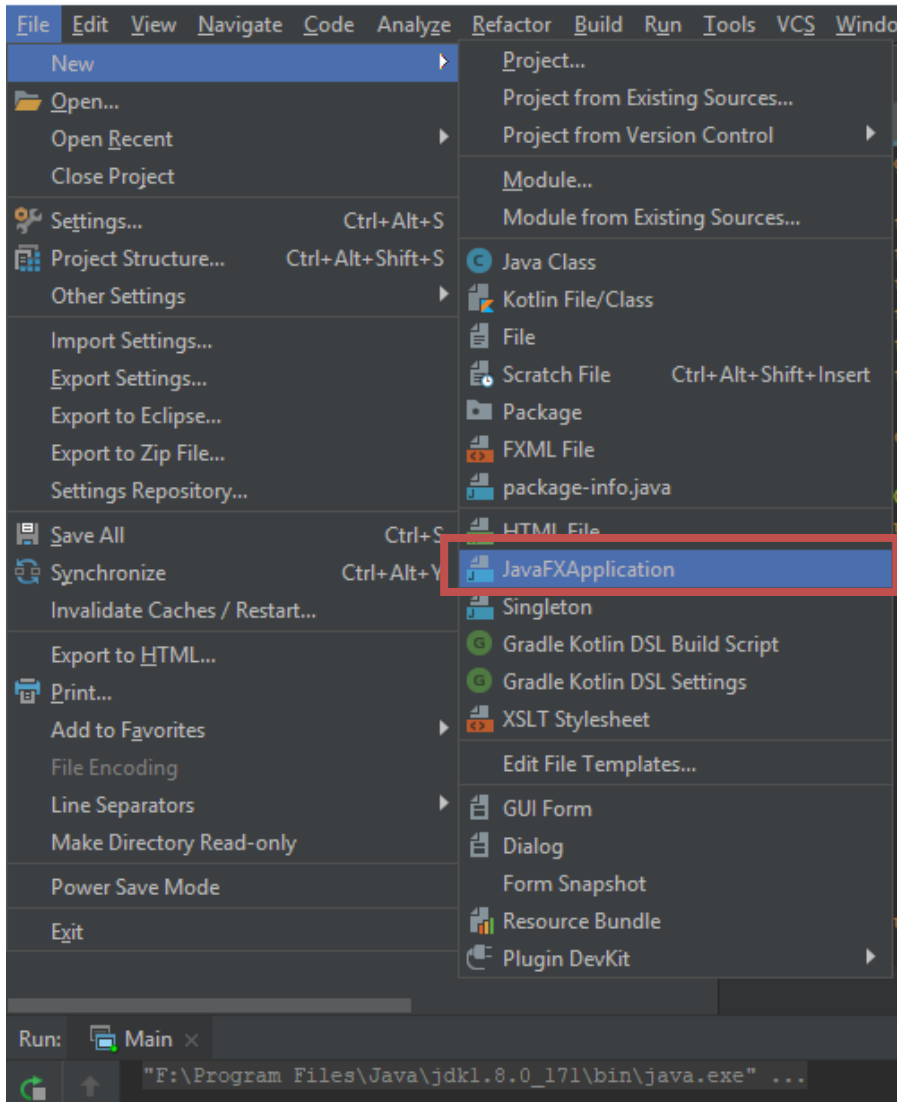
[JavaFX Architecture](#)

Przykład klasy kontrolera:

```
public class Controller implements Initializable {
    ...
}
```

Do stworzenia wizualnie atrakcyjnego interfejsu graficznego można wykorzystać [Material Design](#) . Biblioteka Material Design dla JavaFX jest dostępna [tutaj](#).

Do stworzenia aplikacji w JavaFX można wykorzystać [IDE IntelliJ w darmowej wersji Community](#) oferujące wsparcie dla JavaFX:



Innym możliwym do wykonania zadaniem na laboratorium jest napisanie aplikacji, która pozwala użytkownikowi podać lokalizację i wyświetla dla tej lokalizacji informacje pogodowe, podobnie jak poniżej:



Wrocław, PL  **Sky is Clear**

13.2°C temperature from 13.2 to 13.2 °C, wind 0.83 m/s. clouds 0 %, 1001.82 hpa

Geo coords [51.109, 17.0327]

źródło: <https://openweathermap.org/find?q=Wroc%C5%82aw>

API które można wykorzystać do tego celu jest dostępne [tutaj](#) .