Trong phần này trình bày cách để thực hiện các controller khác nhau và controller này sẽ có thể lấy và xử lý dữ liệu từ một controller khác. (giải thích code được comment trong code).

các bước B1, B2 giống như ở bên trong "C:\Users\user\Downloads\GitHub\My\_Java\javaFX\Khóa học của bro code\switch\_scenes.docx"

**B3: tạo controller2 để điều khiển FXML2.fxml**

|  |
| --- |
| /\*  \* Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license  \* Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template  \*/  package javafxapplication6;  import java.net.URL;  import java.util.ResourceBundle;  import javafx.fxml.FXML;  import javafx.fxml.Initializable;  import javafx.scene.control.Label;  /\*\*  \*  \* @author user  \*/  public class controller2 implements Initializable{  @FXML  private Label lb;  public void display\_layble(String s)  {  lb.setText("Hello" + s);  }  @Override  public void initialize(URL location, ResourceBundle resources) {  }      } |

**B4: Tạo ra controller1 để điều khiển FXML1.fxml và liên kết đến controller2 để có thể áp dụng dữ liệu trong controller1 vào bên trong controller2.**

|  |
| --- |
| /\*  \* Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license  \* Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template  \*/  package javafxapplication6;  import java.io.IOException;  import java.net.URL;  import java.util.ResourceBundle;  import java.util.logging.Level;  import java.util.logging.Logger;  import javafx.event.ActionEvent;  import javafx.fxml.FXML;  import javafx.fxml.FXMLLoader;  import javafx.fxml.Initializable;  import javafx.scene.Node;  import javafx.scene.Parent;  import javafx.scene.Scene;  import javafx.scene.control.TextField;  import javafx.scene.input.MouseEvent;  import javafx.stage.Stage;  /\*\*  \*  \* @author user  \*/  public class controller1 implements Initializable{  @FXML  private TextField tf;  private Parent root;  private Scene scene;  private Stage stage;  public void login(ActionEvent e)  {  String s = tf.getText();  try {  FXMLLoader loader = new FXMLLoader(getClass().getResource("FXML2.fxml")); // load toàn bộ dữ liệu của FXML2.fxml vào loader.  root = loader.load();// load dữ liệu vào root.  controller2 obj\_controller2 = loader.getController(); // dòng này có tác dụng giống như việc chúng ta gán controller class vào bên trong FXML trong scene builder, nhưng với việc sử dụng dòng này chúng ta có thể định nghĩa, sử dụng dữ liệu từ lớp này với các hàm từ lớp khác.  obj\_controller2.display\_layble(s); // sử dụng hàm từ controller khác với dữ liệu từ lớp này.  scene = new Scene(root);  stage = (Stage)((Node)e.getSource()).getScene().getWindow();  stage.setScene(scene);  stage.show();  } catch (IOException ex) {  Logger.getLogger(controller1.class.getName()).log(Level.SEVERE, null, ex);  }  }  @Override  public void initialize(URL location, ResourceBundle resources) {  }      } |