IUT2 Université Grenoble Alpes
Département INFO

TD01 - MVC

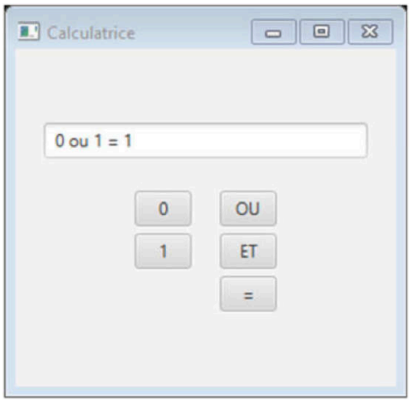
Exercice 1

Soit l'interface suivante qui est une application simple de simulation de retrait et dépôt d'argent.

Vous trouverez en annexe le code correspondant à cette application pour le retrait d'un certain montant. Il suit le modèle MVC.

1) Faites le diagramme de classes de l'application.
2) Quelle classe correspond à la vue, au contrôleur et au modèle ?
3) Faites le diagramme de séquences correspondant à l'enchainement des méthodes lors d'un clic sur le bouton Retirer.
4) Compléter le code correspondant au clic sur le bouton Deposer

Exercice 2 - Préparation TP4



On désire réaliser la calculatrice binaire dont l'interface est donnée ci-contre, en suivant le modèle MVC.

1) Faites le modèle de classes de l'application. Précisez leur rôle '(Modèle,Vue ou Controleur).
2) Faites le diagramme de séquences correspondant à l'enchainement des méthodes lors d'un clic sur le bouton 0.
3) Faites le diagramme de séquences correspondant à l'appel de méthodes lors d'un clic sur le bouton =.

IUT2 Université Grenoble Alpes
Département INFO

Annexes

MainView.fxml

```
<?xml version="1.0" encoding="UTF-8"?>

<?import javafx.geometry.*?>

<?import javafx.scene.control.*?>

<?import javafx.scene.layout.*?>


<VBox stylesheets="@../css/styles.css" xmlns="http://javafx.com/javafx/16" xmlns:fx="http://javafx.com/fxml/
1" fx:controller="com.edencoding.MVCExample">

   <Label styleClass="title" text="Compte en Banque" />

   <GridPane>

      <Label text="Détenteur :" />

      <Label fx:id=« detenteurCompte" GridPane.columnIndex="1" />

      <Label text="Numéro :" GridPane.rowIndex="1" />

      <Label fx:id="numeroCompte" GridPane.columnIndex="1" GridPane.rowIndex="1" />

      <Label text="Solde : " GridPane.rowIndex="2" />

      <Label fx:id="soldeCompte" GridPane.columnIndex="1" GridPane.rowIndex="2" />

   </GridPane>

   <HBox>

      <Button onAction="#handleRetrait" text="Retirer" />

      <TextField fx:id="montantTextField" promptText="Number" text="Montant" />

      <Button layoutX="10.0" onAction="#handleDepot" text="Déposer" />

   </HBox>

</VBox>
```

MVCExampleApp.java

```
package com.edencoding;


import …

public class MVCExampleApp extends Application {

   @Override

   public void start(Stage primaryStage) throws Exception{

      Parent root = FXMLLoader.load(getClass().getResource("/fxml/mainView.fxml"));

      primaryStage.setTitle("MVC Example App");

      primaryStage.getIcons().add(new Image(getClass().getResource("/img/
EdenCodingIcon.png").toExternalForm()));
```

```
        primaryStage.setScene(new Scene(root, 300, 275));
        primaryStage.show();
    }
 public static void main(String[] args) {
        launch(args);
    }
}
```

MVCExample.java

```
package…
import …

public class MVCExample {

    CompteEnBanque compte;
    @FXML private Label detenteurCompte;
    @FXML private Label numeroCompte;
    @FXML private Label soldeCompte;
    @FXML private TextField montantTextField;

    public void initialize(){
        compte = new CompteEnBanque("Zora Labria", 6626, 1000d);
        detenteurCompte.textProperty().bind(compte.detenteurCompteProperty());
        soldeCompte.textProperty().bind(compte.soldeCompteProperty().asString());
        numeroCompte.textProperty().bind(compte.numeroCompteProperty().asString());

        // ensure only numeric input (integers) in text field
        montantTextField.setTextFormatter(new TextFormatter<>(change -> {
            if (change.getText().matches("\\d+") || change.getText().equals("")) {
                return change;
            } else {
                change.setText("");
                change.setRange(
                        change.getRangeStart(),
                        change.getRangeStart()
                );
```

```
            return change;
        }
    }));
  }

  @FXML private void handleRetrait(Event event) {
    compte.retrait(getMontant());
    event.consume();
  }

  private double getMontant(){
    if (montantTextField.getText().equals("")) return 0;
    return Double.parseDouble(montantTextField.getText());
  }
}
```

CompteEnBanque.java

```
package …
import javafx.beans.property.*;

public class CompteEnBanque {

  private final StringProperty detenteurCompte;
  private final IntegerProperty numeroCompte;
  private final DoubleProperty soldeCompte;

  public CompteEnBanque(String detenteurCompte, Integer numeroCompte, Double soldeCompte) {
    this.detenteurCompte = new SimpleStringProperty(detenteurCompte);
    this.numeroCompte = new SimpleIntegerProperty(numeroCompte);
    this.soldeCompte = new SimpleDoubleProperty(soldeCompte);
  }

  public String getDetenteurCompte() {
    return detenteurCompte.get();
  }
```

```
    public StringProperty detenteurCompteProperty() {
        return detenteurCompte;
    }

    public int getNumeroCompte() {
        return numeroCompte.get();
    }

    public IntegerProperty numeroCompteProperty() {
        return numeroCompte;
    }

    public double getSoldeCompte() {
        return soldeCompte.get();
    }

    public DoubleProperty soldeCompteProperty() {
        return soldeCompte;
    }

    public void retrait(double montant){
        soldeCompte.set(soldeCompte.get() - montant);
    }
}
```