/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package vermsudh;

import javafx.application.Application;

import javafx.fxml.FXMLLoader;

import javafx.scene.Parent;

import javafx.scene.Scene;

import javafx.stage.Stage;

/\*\*

\*

\* @author verms

\*/

public class Assign5 extends Application {

public static void main (String []args){

Application.launch(args);

}

@Override

public void start(Stage stage) throws Exception {

//throw new UnsupportedOperationException("Not supported yet."); //To change body of generated methods, choose Tools | Templates.

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

stage.setTitle("Cookie Inventory");

Scene s = new Scene(root);

stage.setScene(s);

stage.show();

}

}

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

<?import java.lang.\*?>

<?import java.util.\*?>

<?import javafx.scene.\*?>

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

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

<BorderPane id="AnchorPane" prefHeight="400.0" prefWidth="600.0" xmlns:fx="http://javafx.com/fxml/1" fx:controller="vermsudh.CookieFXMLController">

<top>

<GridPane>

<Label text="Select Cookie:" />

<ComboBox fx:id="comboBox" GridPane.rowIndex="0" GridPane.columnIndex="2" GridPane.columnSpan="2" />

</GridPane>

</top>

<center>

<GridPane>

<Label text="Enter Quantity Sold:" GridPane.rowIndex="0" GridPane.columnIndex="1"/>

<Label text="Enter Quantity Baked:" GridPane.rowIndex="0" GridPane.columnIndex="2"/>

<TextField fx:id="qtySold" GridPane.rowIndex="1" GridPane.columnIndex="1" />

<TextField fx:id="qtyBaked" GridPane.rowIndex="1" GridPane.columnIndex="2" />

<Button fx:id="sellbtn" onAction="#sellButtonAction" mnemonicParsing="true" text="\_Sell" maxWidth="Infinity" maxHeight="Infinity" GridPane.rowIndex="2" GridPane.columnIndex="1" />

<Button fx:id="addbtn" onAction="#addButtonAction" mnemonicParsing="true" text="\_Add" maxWidth="Infinity" maxHeight="Infinity" GridPane.rowIndex="2" GridPane.columnIndex="2" />

</GridPane>

</center>

<bottom>

<BorderPane>

<right>

<Button fx:id="exitbtn" mnemonicParsing="true" onAction="#CloseButtonAction" text="E\_xit" GridPane.rowIndex="2" GridPane.columnIndex="2" />

</right>

</BorderPane>

</bottom>

</BorderPane>

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package vermsudh;

import java.io.File;

import java.net.URL;

import java.util.ArrayList;

import java.util.List;

import java.util.Optional;

import java.util.ResourceBundle;

import java.util.Scanner;

import static javafx.collections.FXCollections.observableArrayList;

import javafx.event.ActionEvent;

import javafx.fxml.FXML;

import javafx.fxml.Initializable;

import javafx.scene.control.Alert;

import javafx.scene.control.Button;

import javafx.scene.control.ButtonType;

import javafx.scene.control.ComboBox;

import javafx.scene.control.TextField;

import javafx.stage.Stage;

import prog24178.labs.objects.Cookies;

import prog24178.labs.objects.CookieInventoryItem;

/\*\*

\* FXML Controller class

\*

\* @author verms

\*/

public class CookieFXMLController implements Initializable {

@FXML

private Button addbtn, sellbtn, exitButton;

@FXML

private ComboBox<Cookies> comboBox;

@FXML

private TextField qtySold, qtyBaked;

CookieInventoryItem cInvItem = new CookieInventoryItem();

CookieInventoryFile cInvFile = new CookieInventoryFile();

List<CookieInventoryItem> itemList = new ArrayList<CookieInventoryItem>();

@FXML

void CloseButtonAction(ActionEvent event) {

Alert a = new Alert(Alert.AlertType.NONE);

((Stage) (((Button) event.getSource()).getScene().getWindow())).close();

Alert alert = new Alert(Alert.AlertType.CONFIRMATION);

alert.setHeaderText("Exit Program?");

alert.setContentText("Are you sure you wish to exit?");

Optional<ButtonType> showAndWait = alert.showAndWait();

if (alert.getResult() == ButtonType.OK) {

System.exit(0);

}

}

@FXML

void sellButtonAction(ActionEvent event){

if(qtySold.getText().equals("")){

Alert a = new Alert(Alert.AlertType.ERROR);

((Stage) (((Button) event.getSource()).getScene().getWindow())).close();

Alert alert = new Alert(Alert.AlertType.ERROR);

alert.setHeaderText("Please enter the number of cookies sold.");

alert.setTitle("Data Entry Error");

Optional<ButtonType> showAndWait = alert.showAndWait();

if (alert.getResult() == ButtonType.OK) {

System.exit(0);

}

}else if(String.valueOf(cInvItem.getQuantity()).equals(qtySold.getText())) {

System.out.println(cInvItem.getQuantity());

System.out.println("Hello this is sell button");

Alert alert = new Alert(Alert.AlertType.ERROR);

alert.setHeaderText("Sorry, there are no "+comboBox.getValue());

alert.setTitle("Insuffient Inventory");

}

}

@FXML

void addButtonAction(ActionEvent event){

if (qtyBaked.getText().equals("")) {

Alert a = new Alert(Alert.AlertType.ERROR);

((Stage) (((Button) event.getSource()).getScene().getWindow())).close();

Alert alert = new Alert(Alert.AlertType.ERROR);

alert.setHeaderText("Please enter the number of cookies added.");

alert.setTitle("Data Entry Error");

Optional<ButtonType> showAndWait = alert.showAndWait();

if (alert.getResult() == ButtonType.OK) {

System.exit(0);

}

}else if( Integer.parseInt(qtyBaked.getText())>0){

for(Cookies cook : Cookies.values()) {

if(cook.equals(comboBox.getValue())) {

cInvItem = new CookieInventoryItem(cook.getId(), Integer.parseInt(qtyBaked.getText()));

itemList.add(cInvItem);

System.out.println("item list" + itemList);

}

}

File f = new File("cookies.dat");

CookieInventoryFile f1 = new CookieInventoryFile();

f1.addAll(itemList);

f1.writeToFile(f);

}

}

@Override

public void initialize(URL url, ResourceBundle rb) {

// TODO

comboBox.setItems(observableArrayList(Cookies.values()));

}

}

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package vermsudh;

import java.io.BufferedWriter;

import java.io.File;

import java.io.FileNotFoundException;

import java.io.FileWriter;

import java.io.IOException;

import java.io.PrintWriter;

import java.util.ArrayList;

import java.util.Scanner;

import java.util.logging.Level;

import java.util.logging.Logger;

import prog24178.labs.objects.CookieInventoryItem;

/\*\*

\*

\* @author verms

\*/

public class CookieInventoryFile extends ArrayList<CookieInventoryItem>{

public CookieInventoryFile(){}

public CookieInventoryFile(File file){

writeToFile(file);

loadFromFile(file);

}

public CookieInventoryItem find(int id ){

//"this" is used to reference the arraylist

for(int i =0;i< this.size();i++){

CookieInventoryItem temp = this.get(i);

// if(id == temp.getCookie()){}

}

return null;

//why we dont have a getter method for the cookie so that we can match the id

}

public void loadFromFile(File file){

try {

Scanner in = new Scanner(file);

while(in.hasNext()){

//Get input line by line

//Use input(Input as in "1|124", the string input from the file) to make CookieInventoryItem object

//Add the object to "this" arraylist ex. this.add(...)

String line = in.nextLine();

String item = "";

int id = 0, qty = 0;

for (int i = 0; i < line.length(); i++) {

if (line.substring(i, i + 1).equals("|")) {

id = Integer.parseInt(item);

item = "";

}else {

item += line.substring(i, i + 1);

}

}

qty = Integer.parseInt(item);

CookieInventoryItem cit = new CookieInventoryItem(id, qty);

this.add(cit);

}

}catch (FileNotFoundException ex) {

Logger.getLogger(CookieInventoryFile.class.getName()).log(Level.SEVERE, null, ex);

}

}

public void writeToFile(File file){

System.out.println("Hello file" +file.getName());

CookieInventoryItem c1 = null;

for(int i =0; i<this.size();i++){

c1 = this.get(i);

}

try{

FileWriter fw = new FileWriter (file, true);

BufferedWriter bf = new BufferedWriter(fw);

try (PrintWriter output = new PrintWriter(bf)) {

output.println(c1.toFileString());

}

}

catch(IOException e){

System.out.println(e.getMessage());

}

}

}