TrafficLightProject

//Test.java

import java.util.Scanner;

public class TransposeTest {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

int[][] arr = Transpose.getMatrix(scanner);

System.out.println("Original Array: ");

Transpose.printMatrix(arr);

int[][] transpose = Transpose.findTranspose(arr);

System.out.println("Transpose Array: ");

Transpose.printMatrix(transpose);

}

}

//TrafficLightProject.java

import javax.swing.\*;

import java.awt.\*;

import java.util.Objects;

public class TrafficLightProject {

public TrafficLightProject() {

JFrame frame = new JFrame("TrafficLightProject");

JPanel panel = new JPanel();

panel.setBounds(125, 50, 50, 50);

frame.add(panel);

ButtonGroup buttonGroup = new ButtonGroup();

for (int i = 0; i < 3; i++) {

JRadioButton radioButton = new JRadioButton(getColorString(i), false);

radioButton.setBounds(20, 20 \* (i + 1), 100, 20);

frame.add(radioButton);

buttonGroup.add(radioButton);

radioButton.addActionListener(

actionEvent ->

panel.setBackground(

getColor(actionEvent.getActionCommand())

)

);

}

buttonGroup.clearSelection();

frame.setSize(200, 200);

frame.setLayout(null);

frame.setVisible(true);

}

private String getColorString(int i) {

if (i == 0)

return "Green";

if (i == 1)

return "Yellow";

return "Red";

}

private Color getColor(String s) {

if (Objects.equals(s, "Yellow"))

return Color.YELLOW;

if (Objects.equals(s, "Green"))

return Color.GREEN;

return Color.RED;

}

}

//TransposeTest

//Transpose.java

import java.util.Scanner;

public class Transpose {

static int[][] getMatrix(Scanner scanner) {

System.out.print("Number of rows of matrix: ");

int r = scanner.nextInt();

System.out.print("Number of columns of matrix: ");

int c = scanner.nextInt();

int A[][] = new int[c][r], B[][] = new int[c][r];

System.out.println("Enter the Matrix elements: ");

for(int i = 0; i < c; i++) {

for(int j = 0; j < r; j++) {

System.out.print("Enter A" + (i + 1) + (j + 1) + ": ");

A[i][j] = scanner.nextInt();

}

}

return A;

}

static int[][] findTranspose(int[][] A) {

int c = A.length, r = A[0].length;

int B[][] = new int[c][r];

for(int i = 0; i < c; i++) {

for(int j = 0; j < r; j++) {

B[i][j] = A[j][i];

}

}

return B;

}

static void printMatrix(int[][] A) {

int c = A.length, r = A[0].length;

for(int i = 0; i < c; i++) {

for(int j = 0; j < r; j++) {

System.out.print("\tA" + (i + 1) + (j + 1) + ": " + A[i][j]);

}

System.out.println();

}

}

}

//TransposeTest.java

import java.util.Scanner;

public class TransposeTest {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

int[][] arr = Transpose.getMatrix(scanner);

System.out.println("Original Array: ");

Transpose.printMatrix(arr);

int[][] transpose = Transpose.findTranspose(arr);

System.out.println("Transpose Array: ");

Transpose.printMatrix(transpose);

}

}

//VideoRentalInventorySystem

//Video.java

public class Video {

private String name;

private boolean checkout;

private int rating;

public Video(String name) {

this.name = name;

}

public void doCheckout(String name) {

checkout = true; //TODO: Why the name parameter?!

}

public void doReturn(String name) {

checkout = false; //TODO: Why the name parameter?!

}

public void receiveRating(int rating) {

this.rating = rating; //TODO: Why receive not get? Why not use the getter!!

}

public boolean getCheckout() {

return checkout;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public boolean isCheckout() {

return checkout;

}

public void setCheckout(boolean checkout) {

this.checkout = checkout;

}

public int getRating() {

return rating;

}

public void setRating(int rating) {

this.rating = rating;

}

}

//VideoLauncher.java

import java.util.InputMismatchException;

import java.util.Scanner;

public class VideoLauncher {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

while (true) {

System.out.println(

"MAIN MENU\n" +

"=========\n" +

"1. Add Videos:\n" +

"2. Check Out Video :\n" +

"3. Return Video :\n" +

"4. Receive Rating :\n" +

"5. List Inventory :\n" +

"6. Exit :\n"

);

int choice;

try {

System.out.println("Enter your option (1..6): ");

choice = scanner.nextInt();

} catch (InputMismatchException e) {

System.out.println("Please enter a valid input\n");

scanner.nextLine();

continue;

}

if (choice == 1) {

System.out.print("Enter the name of the video you want to add: ");

String name = scanner.next();

VideoStore.addVideo(name);

} else if (choice == 2) {

System.out.print("Enter the name of the video you want to check out: ");

String name = scanner.next();

VideoStore.doCheckout(name);

} else if (choice == 3) {

System.out.print("Enter the name of the video you want to Return: ");

String name = scanner.next();

VideoStore.doReturn(name);

} else if (choice == 4) {

System.out.print("Enter the name of the video you want to Rate: ");

String name = scanner.next();

System.out.print("Enter the rating for this video: ");

int rating = scanner.nextInt();

VideoStore.receiveRating(name, rating);

} else if (choice == 5) {

VideoStore.listInventory();

} else if (choice == 6) {

System.out.println("Exiting...!! Thanks for using the application.");

System.exit(0);

} else {

System.out.println("Please enter a valid choice!");

}

}

}

}

//VideoStore.java

public class VideoStore {

private static final Video[] store = new Video[100]; // TODO: What should be the size of this array? Better using an <code>ArrayList</code>?

private static int length = 0; // TODO: Can't find any other ways to do this other than of course to loop through the array to find the end!!

public static void addVideo(String name) {

store[length++] = new Video(name);

System.out.println("Video “" + name + "” added successfully.");

}

public static void doCheckout(String name) {

for (Video video : store) {

if (name.equals(video.getName())) { //TODO: Not doing .toLower() because it is the primary key

video.doCheckout(name); //TODO: This name is redundant, as far as I can decipher.

System.out.println("Video “" + name + "” checked out successfully.");

return;

}

}

}

public static void doReturn(String name) {

for (Video video : store) {

if (name.equals(video.getName())) { //TODO: Not doing .toLower() because it is the primary key

video.doReturn(name); //TODO: This name is redundant, as far as I can decipher.

System.out.println("Video “" + name + "” returned successfully.");

return;

}

}

}

public static void receiveRating(String name, int rating) {

for (Video video : store) {

if (name.equals(video.getName())) { //TODO: Not doing .toLower() because it is the primary key

video.receiveRating(rating);

System.out.println("Rating “9” has been mapped to the Video “" + name + "”.");

return;

}

}

}

public static void listInventory() {

System.out.println(

"--------------------------------------------------------\n" +

"Video Name | Checkout Status | Rating\n"

);

if (store != null && length > 0)

for (int i = 0; i < length; i++)

printDetails(store[i]);

System.out.println(

"--------------------------------------------------------"

);

}

private static void printDetails(Video video) {

System.out.println(

video.getName() + " | " + video.getCheckout() + " | " + video.getRating() + " | "

);

}

}