import java.util.ArrayList;

import java.util.Scanner;

/\*\*

\* @author Avi Byrne

\* @category Assignment 2

\*

\*/

public class Main {

static Scanner input = new Scanner(System.in);

static ArrayList<Student> allStudents = new ArrayList<Student>();

/\*

\* this arraylist holds all the student objects that are created in the Student class

\*/

public static void main(String[] args) { //main method that starts the program by calling the main menu

mainMenu();

}

private static void mainMenu() {

System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

System.out.println(" Main Menu");

System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

System.out.println("1\t Student Menu");

System.out.println("2\t Class Menu"); //displays options to the user

System.out.println("3\t Admin Menu");

System.out.println("4\t Exit");

System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

String choice = input.nextLine(); //captures the user's input

switch (choice){

case "1":{

studentMenu();

}

case "2":{

classMenu();

}

case "3":{

adminMenu();

}

case "4":{

System.exit(0);

}

default:{

System.out.println("Error. Please enter a numeric value between 1 and 4"); //error message displayed when invalid data is entered

mainMenu();

}

}

}

private static void adminMenu() {

System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

System.out.println(" Admin Menu ");

System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

System.out.println("1\t Print Paid Students"); //displays options to the user

System.out.println("2\t Print Unpaid Students");

System.out.println("3\t Back to Main Menu");

System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

String choice = input.nextLine(); //captures the user's input

switch (choice){

case "1":{

for(Student i: allStudents){

if(i.havePaid()==true){ //prints all the students with paid variable set to true

i.print();

}

}

adminMenu();

}

case "2":{

for(Student i: allStudents){

if(i.havePaid()==false){//prints all the students with paid variable set to false

i.print();

}

}

adminMenu();

}

case "3":{

mainMenu();

}

default:{

System.out.println("Error. Please enter a numeric value between 1 and 3"); //error message displayed when invalid data is entered

adminMenu();

}

}

}

private static void classMenu() { //menu with options directly relating to whole classes

System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

System.out.println(" Class Menu ");

System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

System.out.println("1\t Print Computer Science Class");

System.out.println("2\t Print Games Design Class"); //displays options to the user

System.out.println("3\t Print All Students");

System.out.println("4\t Clear Class List");

System.out.println("5\t Back to Main Menu");

System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

String choice = input.nextLine(); //captures the user's input

switch (choice){

case "1":{

printComputerScience();

break;

}

case "2":{

printGamesDesign();

break;

}

case "3":{

printAll();

break;

}

case "4":{

allStudents.clear();

mainMenu();

}

case "5":{

mainMenu();

}

default:{

System.out.println("Error. Please enter a numeric value between 1 and 4"); //error message displayed when invalid data is entered

classMenu();

}

}

}

private static void studentMenu() { //menu with options directly relating to students

System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

System.out.println(" Student Menu ");

System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

System.out.println("1\t Add New Student");

System.out.println("2\t Edit Student Details"); //displays options to the user

System.out.println("3\t Delete a Student");

System.out.println("4\t Pay Fees");

System.out.println("5\t Back to Main Menu");

System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

String choice = input.nextLine(); //captures the user's input

switch (choice){

case "1":{

addStudent();

}

case "2":{

editStudent();

}

case "3":{

deleteStudent();

}

case "4":

{

makePayment();

}

case "5":{

mainMenu();

}

default:{

System.out.println("Error. Please enter a numeric value between 1 and 5"); //error message displayed when invalid data is entered

studentMenu();

}

}

}

private static void deleteStudent(){ //allows user to delete students

System.out.println("Please enter name of student to delete");

String search = input.nextLine(); //captures the user's input

for(Student i : allStudents){

if(i.getName() != null && i.getName().contains(search)){ //searches for student objects using the name variable

allStudents.remove(i);

studentMenu();

}

}

System.out.println("No Student with this name found");

studentMenu();

}

private static void editStudent(){ //allows user to edit student details

System.out.println("Please enter name of student to edit");

String search = input.nextLine(); //captures the user's input

for(Student i : allStudents){

if(i.getName() != null && i.getName().contains(search)){ //searches for student objects using the name variable

i.print();

System.out.println("1\t Edit Name");

System.out.println("2\t Edit Age");

System.out.println("3\t Edit Course"); //displays options to the user

System.out.println("4\t Edit Paid");

System.out.println("5\t Back to Main Menu");

String choice = input.nextLine(); //captures the user's input

switch (choice){

case "1":{

System.out.println("Enter First Name");

String localFirstName = input.next();

System.out.println("Enter Surname");

String localSurame = input.next(); //captures the data for the name variable for editing

String localName = localFirstName + " " + localSurame;

localName = localName.toLowerCase();

i.editName(localName);

mainMenu();

break;

}

case "2":{

System.out.println("Enter Age");

String age = input.next(); //captures the user's input

i.editAge(age); //allows changing of a students age variable

mainMenu();

break;

}

case "3":{

System.out.println("Press 1 to assign Student to Computer Science or press 2 to assign Student to Games Design");

String courseChoice = input.next(); //captures the user's input

switch(courseChoice){

case "1":{

i.setCourse("Computer Science");

mainMenu();

break;

}

case "2":{ //allows changing of a students course variable

i.setCourse("Games Design");

mainMenu();

break;

}

default:{

System.out.println("Invalid Entry"); //error message displayed when invalid data is entered

editStudent();

}

}

}

case "4":{

System.out.println("Press 1 to set Student to Paid");

System.out.println("Press 2 to set Student to Unpaid"); //displays options to the user

String paidChoice = input.next(); //captures the user's input

switch(paidChoice){

case "1":{

i.editPaid(true);

mainMenu();

break;

}

case "2":{ //allows changing of the paid status of a student

i.editPaid(false);

mainMenu();

break;

}

default:{

System.out.println("Invalid Entry"); //error message displayed when invalid data is entered

editStudent();

}

}

}

case "5":{

mainMenu();

break;

}

default:{

System.out.println("Error. Please enter a numeric value between 1 and 5");

editStudent();

}

}

}

}

System.out.println("No Student with that name found"); //error message displayed when invalid data is entered

studentMenu();

}

private static void addStudent() { //adds a new student object with required information

System.out.println("Enter First Name");

String localFirstName = input.next();

System.out.println("Enter Surname");

String localSurame = input.next();

String localName = localFirstName + " " + localSurame; //captures the data for the name variable

localName = localName.toLowerCase();

System.out.println("Enter age"); //captures the data for the age variable

String localAge = input.next();

Student s = new Student(localName, localAge); //constructor to pass values to student object

System.out.println("1\t Add to Computer Science"); //displays options to the user

System.out.println("2\t Add to Games Design");

String courseChoice = input.next(); //captures the user's input

switch (courseChoice){

case"1":{

s.setCourse("Computer Science");

break;

}

case"2":{ //sets the course of the student

s.setCourse("Games Design");

break;

}

default:{

System.out.println("Invalid Entry"); //error message displayed when invalid data is entered

addStudent();

}

}

System.out.println("1\t Pay Now"); //option for user to pay now or later

System.out.println("2\t Pay Later"); //displays options to the user

String choice = input.next(); //captures the user's input

switch (choice)

{

case "1":

{

try { //when numeric values must be entered by user, a try catch block helps to prevent program from crashing if invalid characters are entered

System.out.println("Current balance is " + s.getBalance() + ". Enter payment");

double moneyEntered = input.nextDouble();

while (moneyEntered<s.getBalance()) //while loop to remain active until the total amount to be paid has been reached

{

System.out.println("Remaining balance is " + (s.getBalance()-moneyEntered)); //displays amount still to be paid

moneyEntered = moneyEntered + input.nextDouble();

}

if(moneyEntered>=s.getBalance())

{

s.setBalance();

}

allStudents.add(s); //adds the student data to it's student class variables

mainMenu();

break;

} catch (Exception e) {

System.out.println("Invalid Entry"); //error message displayed when invalid data is entered

addStudent();

}

}

case "2":

{

allStudents.add(s);

mainMenu();

break;

}

default:{

System.out.println("Invalid Entry"); //error message displayed when invalid data is entered

addStudent();

}

}

}

private static void printAll(){ //prints all students info

for(Student i: allStudents){

i.print();

}

mainMenu();

}

private static void printComputerScience(){ //prints only computer science student's info

for(Student i: allStudents){

if(i.getCourse()=="Computer Science"){ //checks that only computer science students' info is printed

i.print();

}

}

mainMenu();

}

private static void printGamesDesign(){ //prints only game design student's info

for(Student i: allStudents){

if(i.getCourse()=="Games Design"){ //checks that only game design students' info is printed

i.print();

}

}

mainMenu();

}

private static void makePayment(){ //allows user to pay fees if they did not do so on registration

int index =1;

for(Student i: allStudents){

if(i.getBalance()>0){

System.out.print("Index: " + index + "\t"); //displays all students who have yet to pay with an index number to allow user to choose which student to pay for

i.print();

index++;

}

}

try { //when numeric values must be entered by user, a try catch block helps to prevent program from crashing if invalid characters are entered

System.out.println("Please input index number of student for payment");

index=1;

int toPay = input.nextInt(); //captures the user's input

for(Student i: allStudents){

if(i.getBalance()>0){

if (index==toPay){

System.out.println("Current balance is " + i.getBalance() + ". Enter payment");

double moneyEntered = input.nextDouble();

while (moneyEntered<i.getBalance()){ //keeps the payment active until all balance has been paid

System.out.println("Remaining balance is " + (i.getBalance()-moneyEntered));

moneyEntered = moneyEntered + input.nextDouble();

}

if(moneyEntered>=i.getBalance()){

i.setBalance();

}

}

index++;

}

}

mainMenu();

} catch (Exception e) {

System.out.println("Invalid Entry"); //error message displayed when invalid data is entered

makePayment();

}

}

}