**HASEEB TARIQ**

**232506**

**OOP SEMESTER PROJECT**

**CODE**

import java.util.Scanner;

interface UserOperations {

int handleIntegerInput(Scanner scanner);

}

interface AdminOperations {

void adminMenu();

void addUser();

void deleteUser();

void updateUser();

void viewUsers();

void facultyAttendanceMenu();

void viewFacultyAttendance();

void markFacultyAttendance();

}

interface FacultyOperations {

void facultyMenu();

void markStudentAttendance();

}

interface StudentOperations {

void studentMenu();

void viewStudentAttendance();

}

class User implements UserOperations {

Scanner s = new Scanner(System.in);

public int[] AdminId = new int[100];

public String[] AdminName = new String[100];

public String[] AdminEmail = new String[100];

public String[] StudentName = new String[100];

public int[] StudentId = new int[100];

public String[] FacultyName = new String[100];

public int[] FacultyId = new int[100];

public String[] VAttendance = new String[100];

public String[] StudentAttendance = new String[100];

public String[] MFAttendance = new String[100];

int adminSize;

int studentSize;

int facultySize;

public int handleIntegerInput(Scanner scanner) {

System.out.println("Enter your choice:");

while (!scanner.hasNextInt()) {

System.out.println("Invalid input. Please enter a number:");

scanner.next();

}

return scanner.nextInt();

}

}

// Admin class implementing AdminOperations

class Admin extends User implements AdminOperations {

private User user;

public Admin(User user) {

this.user = user;

}

public void adminMenu() {

System.out.println("");

System.out.println("--------------------------WELCOME TO THE ADMIN MENU--------------------------");

boolean exit = false;

while (!exit) {

System.out.println("1. Add User");

System.out.println("2. Delete User");

System.out.println("3. Update User");

System.out.println("4. View Users");

System.out.println("5. Faculty Attendance");

System.out.println("6. Exit");

int choice = user.handleIntegerInput(s);

switch (choice) {

case 1:

addUser();

break;

case 2:

deleteUser();

break;

case 3:

updateUser();

break;

case 4:

viewUsers();

break;

case 5:

facultyAttendanceMenu();

break;

case 6:

System.out.println("Logging out. Goodbye!");

exit = true;

break;

default:

System.out.println("Invalid choice. Please try again.");

}

}

}

public void addUser() {

System.out.println("");

boolean exit = false;

while (!exit) {

System.out.println("Press 1 to Add Admin");

System.out.println("Press 2 to Add Student");

System.out.println("Press 3 to Add Faculty");

System.out.println("Press 4 to Exit");

int choice = user.handleIntegerInput(s);

switch (choice) {

case 1:

System.out.println("You have selected to Add Admin");

System.out.println("How many Admin you want to Enter");

int adminCount = user.s.nextInt();

user.s.nextLine();

for (int i = user.adminSize; i < user.adminSize + adminCount; i++) {

System.out.println("Enter the name of the admin " + (i + 1));

user.AdminName[i] = user.s.nextLine();

System.out.println("Enter the email of the admin " + (i + 1));

user.AdminEmail[i] = user.s.nextLine();

System.out.println("Enter the id of the admin " + (i + 1));

user.AdminId[i] = user.s.nextInt();

user.s.nextLine();

}

user.adminSize += adminCount;

break;

case 2:

System.out.println("You have selected to Add Student");

System.out.println("How many Students you want to Enter");

int studentCount = user.s.nextInt();

user.s.nextLine();

for (int i = user.studentSize; i < user.studentSize + studentCount; i++) {

System.out.println("Enter the name of the student " + (i + 1));

user.StudentName[i] = user.s.nextLine();

System.out.println("Enter the id of the student " + (i + 1));

user.StudentId[i] = user.s.nextInt();

user.s.nextLine();

}

user.studentSize += studentCount;

break;

case 3:

System.out.println("You have selected to Add Faculty");

System.out.println("How many Faculty you want to Enter");

int facultyCount = user.s.nextInt();

user.s.nextLine();

for (int i = user.facultySize; i < user.facultySize + facultyCount; i++) {

System.out.println("Enter the name of the faculty " + (i + 1));

user.FacultyName[i] = user.s.nextLine();

System.out.println("Enter the id of the faculty " + (i + 1));

user.FacultyId[i] = user.s.nextInt();

user.s.nextLine();

}

user.facultySize += facultyCount;

break;

case 4:

System.out.println("Exiting!");

exit = true;

break;

default:

System.out.println("Invalid choice. Please try again.");

break;

}

}

}

public void deleteUser() {

System.out.println("");

System.out.println("Press 1 to Delete Admin");

System.out.println("Press 2 to Delete Student");

System.out.println("Press 3 to Delete Faculty");

int choice = user.handleIntegerInput(s);

switch (choice) {

case 1:

System.out.println("Enter the id of the admin you want to delete");

int adminId = user.s.nextInt();

user.s.nextLine();

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

if (user.AdminId[i] == adminId) {

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

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

user.AdminName[i] = null;

user.AdminEmail[i] = null;

user.AdminId[i] = 0;

break;

}

}

break;

case 2:

System.out.println("Enter the id of the student you want to delete");

int studentId = user.s.nextInt();

user.s.nextLine();

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

if (user.StudentId[i] == studentId) {

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

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

user.StudentName[i] = null;

user.StudentId[i] = 0;

break;

}

}

break;

case 3:

System.out.println("Enter the id of the faculty you want to delete");

int facultyId =user.handleIntegerInput(s);

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

if (user.FacultyId[i] == facultyId) {

System.out.println("Faculty Found");

System.out.println("Deleting Faculty");

user.FacultyName[i] = null;

user.FacultyId[i] = 0;

break;

}

}

break;

default:

System.out.println("Invalid choice. Please try again.");

break;

}

}

public void updateUser() {

System.out.println("Press 1 to Update Admin");

System.out.println("Press 2 to Update Student");

System.out.println("Press 3 to Update Faculty");

int choice = user.handleIntegerInput(s);

switch (choice) {

case 1:

System.out.println("Enter the id of the admin you want to update");

int adminId = user.s.nextInt();

user.s.nextLine();

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

if (user.AdminId[i] == adminId) {

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

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

System.out.println("Enter the new name of the admin");

user.AdminName[i] = user.s.nextLine();

System.out.println("Enter the new email of the admin");

user.AdminEmail[i] = user.s.nextLine();

System.out.println("Enter the new id of the admin");

user.AdminId[i] = user.s.nextInt();

user.s.nextLine();

break;

}

}

break;

case 2:

System.out.println("Enter the id of the student you want to update");

int studentId = user.s.nextInt();

user.s.nextLine();

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

if (user.StudentId[i] == studentId) {

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

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

System.out.println("Enter the new name of the student");

user.StudentName[i] = user.s.nextLine();

System.out.println("Enter the new id of the student");

user.StudentId[i] = user.s.nextInt();

user.s.nextLine();

break;

}

}

break;

case 3:

System.out.println("Enter the id of the faculty you want to update");

int facultyId =user.handleIntegerInput(s);

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

if (user.FacultyId[i] == facultyId) {

System.out.println("Faculty Found");

System.out.println("Updating Faculty");

System.out.println("Enter the new name of the faculty");

user.FacultyName[i] = user.s.nextLine();

System.out.println("Enter the new id of the faculty");

user.FacultyId[i] = user.s.nextInt();

user.s.nextLine();

break;

}

}

break;

default:

System.out.println("Invalid choice. Please try again.");

break;

}

}

public void viewUsers() {

System.out.println("Press 1 to View Admins");

System.out.println("Press 2 to View Students");

System.out.println("Press 3 to View Faculty");

int choice = user.handleIntegerInput(s);

switch (choice) {

case 1:

System.out.println("Admins:");

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

if (user.AdminName[i] != null) {

System.out.println("Admin Name: " + user.AdminName[i] + " | Admin Email: " + user.AdminEmail[i] + " | Admin ID: " + user.AdminId[i]);

}

}

break;

case 2:

System.out.println("Students:");

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

if (user.StudentName[i] != null) {

System.out.println("Student Name: " + user.StudentName[i] + " | Student ID: " + user.StudentId[i]);

}

}

break;

case 3:

System.out.println("Faculty:");

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

if (user.FacultyName[i] != null) {

System.out.println("Faculty Name: " + user.FacultyName[i] + " | Faculty ID: " + user.FacultyId[i]);

}

}

break;

default:

System.out.println("Invalid choice. Please try again.");

break;

}

}

public void facultyAttendanceMenu() {

boolean exit = false;

while (!exit) {

System.out.println("Press 1 to Mark Faculty Attendance");

System.out.println("Press 2 to View Faculty Attendance");

System.out.println("Press 3 to Exit");

int choice = user.handleIntegerInput(s);

user.s.nextLine();

switch (choice) {

case 1:

markFacultyAttendance();

break;

case 2:

viewFacultyAttendance();

break;

case 3:

exit = true;

break;

default:

System.out.println("Invalid choice. Please try again.");

break;

}

}

}

public void markFacultyAttendance() {

System.out.println("Enter the ID of the faculty to mark attendance:");

int facultyId = user.s.nextInt();

user.s.nextLine();

System.out.println("Enter attendance status (Present/Absent):");

String attendance = user.s.nextLine();

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

if (user.FacultyId[i] != 0 && user.FacultyId[i] == facultyId) {

user.MFAttendance[i] = attendance;

System.out.println("Attendance for Faculty ID " + facultyId + " marked as " + attendance);

break;

}

}

}

public void viewFacultyAttendance() {

System.out.println("Faculty Attendance:");

for (int i = 0; i < user.facultySize; i++) {

if (user.FacultyName[i] != null) {

System.out.println("Faculty ID: " + user.FacultyId[i] + ", Name: " + user.FacultyName[i] + ", Attendance: " + user.MFAttendance[i]);

}

}

}

}

class Faculty extends User implements FacultyOperations {

private User user;

public Faculty(User user) {

this.user = user;

}

public void facultyMenu() {

System.out.println("");

System.out.println("--------------------------WELCOME TO THE FACULTY MENU--------------------------");

boolean exit = false;

while (!exit) {

System.out.println("1. Mark Student Attendance");

System.out.println("2. Exit");

int choice = user.handleIntegerInput(s);

switch (choice) {

case 1:

markStudentAttendance();

break;

case 2:

System.out.println("Logging out. Goodbye!");

exit = true;

break;

default:

System.out.println("Invalid choice. Please try again.");

}

}

}

public void markStudentAttendance() {

System.out.println("Enter the ID of the student to mark attendance:");

int studentId = user.s.nextInt();

user.s.nextLine();

boolean found = false;

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

if (user.StudentId[i] == studentId) {

found = true;

System.out.println("Student Found: " + user.StudentName[i]);

System.out.println("Enter attendance status (Present/Absent):");

String attendance = user.s.nextLine();

user.StudentAttendance[i] = attendance;

System.out.println("Attendance for student ID " + studentId + " marked as " + attendance);

break;

}

}

if (!found) {

System.out.println("Student with ID " + studentId + " not found.");

}

}

}

class Student extends User implements StudentOperations {

private User user;

public Student(User user) {

this.user = user;

}

public void studentMenu() {

System.out.println("");

System.out.println("--------------------------WELCOME TO THE STUDENT MENU--------------------------");

boolean exit = false;

while (!exit) {

System.out.println("1. View Attendance");

System.out.println("2. Exit");

int choice = user.handleIntegerInput(s);

switch (choice) {

case 1:

viewStudentAttendance();

break;

case 2:

System.out.println("Logging out. Goodbye!");

exit = true;

break;

default:

System.out.println("Invalid choice. Please try again.");

}

}

}

public void viewStudentAttendance() {

System.out.println("Enter your student ID to view attendance:");

int studentId = user.s.nextInt();

boolean found = false;

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

if (user.StudentId[i] == studentId) {

found = true;

String attendance = user.StudentAttendance[i] != null ? user.StudentAttendance[i] : "Not Marked";

System.out.println("ID: " + user.StudentId[i] + ", Name: " + user.StudentName[i] + ", Attendance: " + attendance);

break;

}

}

if (!found) {

System.out.println("Student with ID " + studentId + " not found.");

}

}

}

public class HelloWorld {

public static void main(String[] args) {

Scanner s = new Scanner(System.in);

User user = new User();

Admin admin = new Admin(user);

Faculty faculty = new Faculty(user);

Student student = new Student(user);

boolean exit = false;

while (!exit) {

System.out.println("Choose the type of user:");

System.out.println("1. Admin");

System.out.println("2. Faculty");

System.out.println("3. Student");

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

int userType = user.handleIntegerInput(s);

switch (userType) {

case 1:

admin.adminMenu();

break;

case 2:

faculty.facultyMenu();

break;

case 3:

student.studentMenu();

break;

case 4:

System.out.println("Exiting the system. Goodbye!");

exit = true;

break;

default:

System.out.println("Invalid choice. Please try again.");

}

}

}

}

**CODE SCREENSHOTS**











