Benewende Pierre BONKOUNGOU

Programming Unit 3

In [ ]: import java.util.ArrayList;

import java.util.Scanner;

public class StudentRecordManagementSystem {

private static ArrayList**<**Student**>** students **=** new ArrayList**<>**(); private static int totalStudents **=** 0;

public static void main(String[] args) { displayMenu();

}

private static void displayMenu() { Scanner scanner **=** new Scanner(System.in); int option **=** 0; **//** Initialize option before using it in the loop c

do {

System.out.println("Student Record Management System Menu:");

System.out.println("1. Add a new student");

System.out.println("2. Update student information");

System.out.println("3. View student details");

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

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

try { option **=** scanner.nextInt();

scanner.nextLine(); **//** Consume the newline character left } catch (Exception e) {

System.out.println("Invalid input. Please enter a valid op scanner.nextLine(); **//** Consume the invalid input continue;

}

switch (option) { case 1: addStudent(scanner); break; case 2: updateStudent(scanner); break; case 3: viewStudentDetails(scanner); break; case 4:

System.out.println("Exiting the Student Record Managem break; default:

System.out.println("Invalid option. Please enter a num

}

} while (option **!=** 4);

scanner.close();

}

private static void addStudent(Scanner scanner) { System.out.println("Add a new student:");

System.out.print("Enter student name: ");

String name **=** scanner.nextLine(); System.out.print("Enter student ID: "); int id **=** scanner.nextInt();

System.out.print("Enter student age: "); int age **=** scanner.nextInt();

System.out.print("Enter student grade: "); double grade **=** scanner.nextDouble();

Student newStudent **=** new Student(id, name, age, grade); students.add(newStudent); totalStudents**++**;

System.out.println("Student added successfully.");

}

private static void updateStudent(Scanner scanner) {

System.out.println("Update student information:"); System.out.print("Enter student ID to update: "); int idToUpdate **=** scanner.nextInt();

boolean studentFound **=** false; for (Student student : students) { if (student.getId() **==** idToUpdate) {

System.out.print("Enter new student name: "); student.setName(scanner.next());

System.out.print("Enter new student age: "); student.setAge(scanner.nextInt());

System.out.print("Enter new student grade: "); student.setGrade(scanner.nextDouble());

System.out.println("Student information updated successful studentFound **=** true; break;

}

}

if (**!**studentFound) {

System.out.println("Student with ID " **+** idToUpdate **+** " not fou }

}

private static void viewStudentDetails(Scanner scanner) { System.out.println("View student details:"); System.out.print("Enter student ID to view details: "); int idToView **=** scanner.nextInt();

boolean studentFound **=** false; for (Student student : students) { if (student.getId() **==** idToView) {

System.out.println("Student Details:");

System.out.println("Name: " **+** student.getName());

System.out.println("Age: " **+** student.getAge());

System.out.println("Grade: " **+** student.getGrade());

studentFound **=** true; break;

}

}

if (**!**studentFound) {

System.out.println("Student with ID " **+** idToView **+** " not found

}

}

private static class Student { private int id; private String name; private int age; private double grade;

public Student(int id, String name, int age, double grade) { this.id **=** id; this.name **=** name; this.age **=** age; this.grade **=** grade;

}

public int getId() { return id;

}

public String getName() { return name;

}

public void setName(String name) {

this

.

name

**=**

name

;

}

public

int

getAge

()

{

return

age

;

}

public

void

setAge

(

int

age

)

{

this

.

age

**=**

age

;

}

public

double

getGrade

()

{

return

grade

;

}

public

void

setGrade

(

double

grade

)

{

this

.

grade

**=**

grade

;

}

}

}

A screenshot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generated

## Explanation:

1. **StudentRecordManagementSystem Class:**
   * It serves as the main class containing the **main** method and managing the student records.
2. **ArrayList<Student> and totalStudents:**
   * **students**: An ArrayList to store instances of the nested **Student** class.
   * **totalStudents**: A static variable to keep track of the total number of students.
3. **displayMenu Method:**
   * Displays the main menu for the Student Record Management System and handles user input.
4. **addStudent, updateStudent, viewStudentDetails Methods:**
   * Methods to add a new student, update student information, and view student details, respectively.
5. **Student Class (Nested):**
   * Encapsulates the details of a student (id, name, age, grade) with getter and setter methods.
6. **Main Method:**
   * Calls the **displayMenu** method to start the Student Record Management System.

This program allows administrators to interactively add, update, and view student records in a console-based system. It uses a nested **Student** class for encapsulation, an ArrayList to store students, and a menu-driven approach for user interaction. Error handling is implemented to handle invalid user inputs.