GenericList.java

import java.util.ArrayList;

public class GenericList<T> {

private ArrayList<T> list;

// Constructor

public GenericList() {

list = new ArrayList<>();

}

// Method to add elements to the list

public void add(T element) {

list.add(element);

}

// Method to retrieve elements by index

public T get(int index) {

if (index >= 0 && index < list.size()) {

return list.get(index);

} else {

throw new IndexOutOfBoundsException("Index is out of bounds");

}

}

// Method to print the list

public void print() {

for (T element : list) {

System.***out***.println(element);

}

}

// Main method for testing

public static void main(String[] args) {

// Creating a GenericList of integers

GenericList<Integer> intList = new GenericList<>();

intList.add(10);

intList.add(20);

intList.add(30);

// Printing the integer list

System.***out***.println("Integer List:");

intList.print();

// Retrieving and printing element by index

System.***out***.println("Element at index 1: " + intList.get(1));

// Creating a GenericList of strings

GenericList<String> stringList = new GenericList<>();

stringList.add("Hello");

stringList.add("World");

// Printing the string list

System.***out***.println("\nString List:");

stringList.print();

}

}

Output

Integer List:

10

20

30

Element at index 1: 20

String List:

Hello

World

MaxFinder.java

public class MaxFinder {

// Generic method to find the maximum element in an array

public static <T extends Comparable<T>> T findMax(T[] arr) {

if (arr == null || arr.length == 0) {

throw new IllegalArgumentException("Array is empty or null");

}

T max = arr[0];

for (int i = 1; i < arr.length; i++) {

if (arr[i].compareTo(max) > 0) {

max = arr[i];

}

}

return max;

}

// Main method for testing

public static void main(String[] args) {

// Test with integers

Integer[] intArray = {10, 20, 5, 30, 15};

System.***out***.println("Maximum integer: " + *findMax*(intArray));

// Test with doubles

Double[] doubleArray = {3.14, 2.71, 1.618, 2.718};

System.***out***.println("Maximum double: " + *findMax*(doubleArray));

// Test with strings

String[] stringArray = {"apple", "banana", "orange", "grape"};

System.***out***.println("Maximum string: " + *findMax*(stringArray));

}

}

Output

Maximum integer: 30

Maximum double: 3.14

Maximum string: orange

EmployeeDemo.java

import java.util.HashSet;

class Employee {

private int employeeID;

private String employeeName;

// Constructor

public Employee(int employeeID, String employeeName) {

this.employeeID = employeeID;

this.employeeName = employeeName;

}

// Overriding equals method

*@Override*

public boolean equals(Object obj) {

if (this == obj) {

return true;

}

if (obj == null || getClass() != obj.getClass()) {

return false;

}

Employee other = (Employee) obj;

return employeeID == other.employeeID;

}

// Getter methods

public int getEmployeeID() {

return employeeID;

}

public String getEmployeeName() {

return employeeName;

}

// toString method

*@Override*

public String toString() {

return "EmployeeID: " + employeeID + ", EmployeeName: " + employeeName;

}

}

public class EmployeeDemo {

public static void main(String[] args) {

// Create three employee instances with different employeeID

Employee employee1 = new Employee(101, "John");

Employee employee2 = new Employee(102, "Alice");

Employee employee3 = new Employee(103, "Bob");

// Create a HashSet to store employees

HashSet<Employee> employeeSet = new HashSet<>();

// Add all three employees to the HashSet

employeeSet.add(employee1);

employeeSet.add(employee2);

employeeSet.add(employee3);

// Create another employee with the same employeeID as the first employee

Employee employee4 = new Employee(101, "Jane");

// Add employee4 to the HashSet

boolean added = employeeSet.add(employee4);

// Print if the employee is added or not

if (added) {

System.***out***.println("Employee with ID " + employee4.getEmployeeID() + " is added to the HashSet.");

} else {

System.***out***.println("Employee with ID " + employee4.getEmployeeID() + " is not added to the HashSet.");

}

// Print all the employees in the HashSet

System.***out***.println("\nEmployees in the HashSet:");

for (Employee employee : employeeSet) {

System.***out***.println(employee);

}

}

}

Output

Employees in the HashSet:

EmployeeID: 101, EmployeeName: Jane

EmployeeID: 102, EmployeeName: Alice

EmployeeID: 103, EmployeeName: Bob

EmployeeID: 101, EmployeeName: John

StudentDemo.java

import java.util.HashSet;

import java.util.Objects;

class Student {

private int studentID;

private String studentName;

// Constructor

public Student(int studentID, String studentName) {

this.studentID = studentID;

this.studentName = studentName;

}

// Overriding equals method

*@Override*

public boolean equals(Object obj) {

if (this == obj) {

return true;

}

if (obj == null || getClass() != obj.getClass()) {

return false;

}

Student other = (Student) obj;

return studentID == other.studentID;

}

// Overriding hashCode method

*@Override*

public int hashCode() {

return Objects.*hash*(studentID);

}

// Getter methods

public int getStudentID() {

return studentID;

}

public String getStudentName() {

return studentName;

}

// toString method

*@Override*

public String toString() {

return "StudentID: " + studentID + ", StudentName: " + studentName;

}

}

public class StudentDemo {

public static void main(String[] args) {

// Create a HashSet to store students

HashSet<Student> studentSet = new HashSet<>();

// Create a few students and add them to the HashSet

Student student1 = new Student(101, "John");

Student student2 = new Student(102, "Alice");

Student student3 = new Student(103, "Bob");

studentSet.add(student1);

studentSet.add(student2);

studentSet.add(student3);

// Create another student with the same studentID

Student student4 = new Student(101, "Jane");

// Add student4 to the HashSet

boolean added = studentSet.add(student4);

// Print if the student is added or not

if (added) {

System.***out***.println("Student with ID " + student4.getStudentID() + " is added to the HashSet.");

} else {

System.***out***.println("Student with ID " + student4.getStudentID() + " is not added to the HashSet.");

}

// Print all the students in the HashSet

System.***out***.println("\nStudents in the HashSet:");

for (Student student : studentSet) {

System.***out***.println(student);

}

}

}

Output

Student with ID 101 is not added to the HashSet.

Students in the HashSet:

StudentID: 101, StudentName: John

StudentID: 102, StudentName: Alice

StudentID: 103, StudentName: Bob