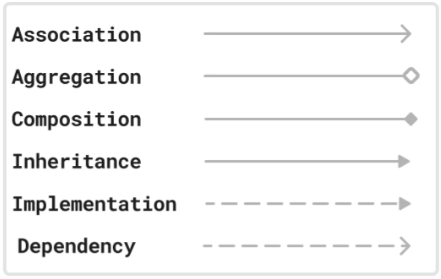
**UML Diagram(Class Diagram)**

Visibility markers indicate the accessibility of attributes and methods within a class.

* +**(Public):** The attribute or method is accessible from any class.
* -**(Private):** The attribute or method is only accessible within the same class.
* #**(Protected):** The attribute or method is accessible within the same class and its subclasses.
* ~**(Package):** The attribute or method is accessible within the same package.

**Relationships in UML class diagrams:**

****

**Enhanced Forloop using Arrays:**

public class EnhancedForLoopExample {

public static void main(String[] args) {

int[] numbers = {10, 20, 30, 40, 50};

System.out.println("Using Enhanced For Loop:");

for (int num : numbers) {

System.out.println(num);

}

}

}

**Enhanced for loop using an Array of Employee Objects:**

class Employee {

private String name;

private int id;

// Constructor

public Employee(int id, String name) {

this.id = id;

this.name = name;

}

// Getter methods

public int getId() {

return id;

}

public String getName() {

return name;

}

}

public class EmployeeArrayExample {

public static void main(String[] args) {

// Creating an array of Employee objects

Employee[] employees = {

new Employee(101, "Alice"),

new Employee(102, "Bob"),

new Employee(103, "Charlie")

};

// Using Enhanced for loop to iterate over the array

for (Employee emp : employees) {

System.out.println("ID: " + emp.getId() + ", Name: " + emp.getName());

}

}

}

**Enhanced for loop using ArrayList :**

import java.util.ArrayList;

import java.util.List;

public class ForEachListExample {

public static void main(String[] args) {

List<String> names = new ArrayList<>();

names.add("Alice");

names.add("Bob");

names.add("Charlie");

System.out.println("Using Enhanced For Loop:");

for (String name : names) {

System.out.println(name);

}

}

}

**Enhanced for loop using an ArrayList of Employee Objects:**

import java.util.ArrayList;

import java.util.List;

class Employee {

private String name;

private int id;

// Constructor

public Employee(int id, String name) {

this.id = id;

this.name = name;

}

// Getter methods

public int getId() {

return id;

}

public String getName() {

return name;

}

}

public class EmployeeArrayListExample {

public static void main(String[] args) {

// Creating an ArrayList of Employee objects

List<Employee> employeeList = new ArrayList<>();

employeeList.add(new Employee(101, "Alice"));

employeeList.add(new Employee(102, "Bob"));

employeeList.add(new Employee(103, "Charlie"));

// Using Enhanced for loop to iterate over the ArrayList

for (Employee emp : employeeList) {

System.out.println("ID: " + emp.getId() + ", Name: " + emp.getName());

}

}

}

**With Nornal forloop:**

class Employee {

private String name;

private int id;

// Constructor

public Employee(int id, String name) {

this.id = id;

this.name = name;

}

// Getter methods

public int getId() {

return id;

}

public String getName() {

return name;

}

}

public class EmployeeArrayExample {

public static void main(String[] args) {

// Creating an array of Employee objects

Employee[] employees = {

new Employee(101, "Alice"),

new Employee(102, "Bob"),

new Employee(103, "Charlie")

};

// Using a normal for loop to iterate over the array

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

System.out.println("ID: " + employees[i].getId() + ", Name: " + employees[i].getName());

}

}

}