import java.util.HashMap;

import java.util.Map;

public class ContractManager {

private Map<String, String> contracts;

public ContractManager() {

contracts = new HashMap<>(); // Stores contracts using a HashMap

}

// Method to add a new contract

public void addContract(String contractId, String contractDetails) {

contracts.put(contractId, contractDetails);

System.out.println("Contract " + contractId + " added.");

}

// Method to remove a contract by ID

public void removeContract(String contractId) {

if (contracts.containsKey(contractId)) {

contracts.remove(contractId);

System.out.println("Contract " + contractId + " removed.");

} else {

System.out.println("Contract " + contractId + " not found.");

}

}

// Method to retrieve contract details by ID

public String getContract(String contractId) {

return contracts.getOrDefault(contractId, "Contract not found.");

}

public static void main(String[] args) {

ContractManager manager = new ContractManager();

manager.addContract("001", "Contract A, Amount: $1000");

System.out.println(manager.getContract("001"));

manager.removeContract("001");

}

}

Explanation

1. ContractManager class: This class manages contract data.

contracts: A HashMap is used to store contracts, where each contract is identified by a unique contractId. The key is contractId, and the value is contractDetails.

2. addContract method: Adds a contract using a unique contractId and its details (contractDetails). It stores the data in the contracts HashMap.

3. removeContract method: Removes a contract from the HashMap using its contractId. If the contract ID is not found, it prints a message indicating the contract was not found.

4. getContract method: Retrieves contract details based on the contractId. If the contract is not found, it returns "Contract not found."

5. main method: Demonstrates usage by adding, retrieving, and removing a contract.