**2. Create the following two database in in-memory database:**

| DNO | DNAME |
|---|---|
| 10 | Admin |
| 20 | Accounts |
| 30 | Sales |
| 40 | Marketing |
| 50 | Purchasing |

| ENO | ENAME | DNO | SALARY |
|---|---|---|---|
| 1 | Amal | 10 | 30000 |
| 2 | Shyamal | 30 | 50000 |
| 3 | Kamal | 40 | 10000 |
| 4 | Nirmal | 50 | 60000 |
| 5 | Bimal | 20 | 40000 |
| 6 | Parimal | 10 | 20000 |

Design the two APIs

Important Points:
I) Run the server on port 9000. The APIs should only serve on port 9000.
II) Exceptions should be handled.
III) Casting if any, should be type safe.
IV) You need to explain your answer.
V) Keep your results ready in a HTTP client application like postman.

**b. Write an Api to return list of all employees by their DNAME.**

E.g.: http://localhost:9000/api?DNAME=Admin
This Api should return the list of details of all the employees whose DNAME is Admin. Do not use DNO directly for creating this API.

**Explanation**

It is an API that return the details of the employee whose Eno is 1

Step 1:

First the packages are imported. The code uses classes from the com.sun.net.httpserver package to work with HTTP server functionality.

Step 2:

This class is the main class of the program. It defines two static Map objects to store department information (department Database) and employee information (employee Database).

Step 3:

The main method is the entry point of the program. It populates the department Database and employee Database maps with sample data. Then, it creates an instance of the HttpServer class, binds it to port 9000, and associates a DepartmentNameHandlers instance with the /api context. Finally, the server is started and a message is printed indicating that the server has started.

Step 4:

A Employee class, this defines the structure of an Employee. It has properties for employee ID, name, department number (dno), and salary. It also has getter methods to access these properties.

Step 5:

A Department class, this defines the structure of an Employee. It has properties for Department No(dno) and Department Name(dname). It also has getter methods to access these properties.

Step 6:

This inner static class implements the HttpHandler interface, which is responsible for handling incoming HTTP requests. The handle method generates a response by iterating through the employee Database and checking if each employee belongs to the "Admin" department. If yes then shows the details of that employees using the previously created object of department & employee class. The response is then sent back to the client using the sendResponse method.

The Code starts a port and there returns the details of employee whose eno is 1