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

Important Points:-

1. Run the server on port 9000. The APIs should only serve on port 9000.

2. Exceptions should be handled.

3. Casting if any, should be type safe.

4. You need to explain your answer.

5. Keep your results ready in a HTTP client application like postman.

Question:-

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.

Step 1:-

Importing Packages In this step, the necessary packages are imported to set up an HTTP server for handling requests and responses. The com.sun.net.httpserver package provides classes for working with HTTP server functionality.

Step 2:-

Main Class and Databases The main class of the program is defined. This class sets up two static Map objects, one for storing department information (referred to as the "department Database") and another for storing employee information (referred to as the "employee Database").

Step 3:-

Main Method and Server Setup The main method acts as the entry point of the program. It populates the department and employee databases with sample data. Then, it creates an instance of the HttpServer class, binds it to a specified port (in this case, port 9000), and associates a handler class (let's call it DepartmentNameHandlers) with the /api context. Finally, the server is started, and a message is printed to indicate that the server has started.

Step 4:-

Employee Class An Employee class is defined. This class serves as a blueprint for the structure of an employee. It has properties like employee ID, name, department number (referred to as "dno"), and salary. The class also provides getter methods to access these properties.

Step 5:-

Department Class A Department class is defined. This class defines the structure of a department, including properties like Department Number (dno) and Department Name (dname). Like the Employee class, it also includes getter methods to access these properties.

Step 6:-

DepartmentNameHandlers Class An inner static class named DepartmentNameHandlers is created. This class implements the HttpHandler interface, which allows it to handle incoming HTTP requests. The handle method is the core of this class. Inside this method, the program iterates through the employee Database, checking if each employee belongs to the "Admin" department. If an employee is found in the "Admin" department, the details of that employee (using the previously created Employee and Department objects) are prepared. The response containing the employee details is then sent back to the client using the sendResponse method.

This sequence of steps essentially sets up an HTTP server that listens on port 9000. When a request is made to the /api context, the DepartmentNameHandlers class processes the request, filters employees belonging to the "Admin" department, and sends back their details as a response.