Pentonix Internship Assignment 2:

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:-

- Run the server on port 9000. The APIs should only serve on port 9000.
- Exceptions should be handled.
- Casting if any, should be type safe.
- You need to explain your answer.
- Keep your results ready in a HTTP client application like postman.

Write an Api to return response depending on ENO. E.g.: http://localhost:9000/api?ENO=1

This Api should return the details of the employee whose ENO is 1.

Approach:

Step 1: -

Package and Import Statements In this step, the necessary packages and import statements are included to ensure access to the required classes and functionality. These could include imports for HTTP server-related classes, data structures, and utilities.

Step 2:-

Main Class and Static Maps The main class of the program is responsible for managing the department and employee information. It employs static maps to store this data efficiently. These maps are used to link employee IDs with their corresponding department details, creating a relational structure.

Step 3:-

Main Method and Server Setup Inside the main method, the program performs several key actions. It populates the employee and department databases, ensuring that relevant data is available for processing. Subsequently, an HTTP server is created, listening on port 9000. To handle incoming requests related to departments, a DepartmentHandler instance is associated with the /api context. Finally, the server is started, making it ready to accept and respond to requests.

Step 4:-

Employee Class This class defines the blueprint for an Employee object. The object's properties include attributes such as employee ID, name, department number (DNO), and salary. The class provides getter methods to access these attributes, encapsulating the data and enabling controlled access.

Step 5:-

Department Class In this step, a class named Department is established to model the structure of a department. The class incorporates attributes like the department number (DNO) and department name (DNAME). It offers getter methods to retrieve these attributes, ensuring proper data encapsulation.

Step 6:-

DepartmentHandler Inner Class An inner static class named DepartmentHandler is implemented within the main class. This inner class implements the HttpHandler interface, which mandates the implementation of the handle method. This method processes incoming HTTP requests, particularly focusing on requests containing an employee

number (ENO) query parameter. The ENO is extracted from the query, and the corresponding employee's details are retrieved. If the employee exists, a Department instance is created, and the employee's details are sent as a response along with the associated department name.

Main Execution Process:-

When the program is executed, it initializes an HTTP server on port 9000. This server is primed to respond to requests directed at the /api endpoint. Upon receiving a request with an employee number (ENO) query parameter, the server engages its lookup mechanism. It cross-references the given ENO with the stored employee data. If a match is found, the server generates a response containing the employee's particulars and their department name. Conversely, if no matching employee is located or the request is malformed, the server dispatches relevant error responses, ensuring proper feedback to the client.

In essence, this program establishes a communication framework via an HTTP server, facilitating the retrieval of employee and department information in a controlled and organized manner.