**Contents**

1. MySQL
2. Java Collection Framework
3. Java Multithreading
4. Java Spring
5. Spring Boot & Microservices
6. React.js

Pre-requisites

* OOPS concepts
  + Inheritance
  + Encapsulation
  + Polymorphism – Overriding
  + Abstraction – abstract class & interface
* Exception Handling
  + try, catch, finally, throw & throws
* Object – root class
* String, LocalDate, LocalDateTime, LocalTime

Exceptions – Types

1. Checked Exceptions – Need to handle it at the compilation time
2. Unchecked Exceptions – These are ignored by the compiler

ArithmeticException, NullPointerException, ArrayIndexOutpBoundsException – Unchecked Exceptions

SQLException, IOException, Exception – Checked Exceptions

Naming conventions:

Keywords: lowercase, example: private, public, int, long, void, double, …

Classnames & Constructors: Starts with uppercase & follows the camel case

Method names & Variable names: Starts with lowercase & follows the camel case

Final variables: Every letter will be in uppercase & uses \_ for multiple words

Example: HELLO\_WORLD

Predefined-classes

1. LocalDate: It is to represent the date like doj, dob, current date
2. LocalDateTime: It is to represent the date & time both like login times
3. LocalTime: It is to represent the time

LocalDate follows ISO standard format to represent the date, i.e, yyyy-MM-dd

ISO format is followed by all the applications & devices

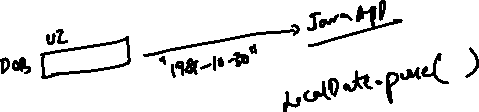
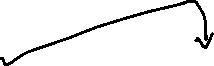
How to create LocalDate

LocalDate dob = LocalDate.of(1980, 11, 25); // 1980-11-25

LocalDate dob2 = LocalDate.parse(“1981-10-30”); // 1981-10-30

LocalDate dob3 = LocalDate.parse(“20-11-1989”, DateTimeFormatter.ofPattern(“dd-MM-yyyy”))

// dob3 = 1989-11-20



LocalDate is present in java.time package which you must import.

Note: Object, String, System and etc are part of java.lang which you don’t have to import

toString(): It is called when you print the object, by default it returns memory address & it is present in Object class, but you can override to return the object property so that when you print object it can print the object property by calling toString().

Signature of toString(): public String toString()

Creating multiple layers to maintain the user

Since we don’t have the database we are maintaining the user data in an array.

class UserService {   
 User[] users = new User[5];   
 store(User user) { // store user in the array }  
 findAll() { // returns the array }   
}

How to handle exception

You need to handle exceptions using try catch, however the code can propagate a checked exception to the caller so that caller will be notified at the compilation time to handle the exception

Scenario 1:   
User findByName(String name) {   
 try {   
 if(..) { throw check-exception }  
 } catch (…) { }  
}

Scenario 2:

User findByName(String name) throws checked-exception {   
 if(…) { throw checked-exception }  
}

You can’t throw predefined exceptions because they don’t look valid for all the business requirements, hence you must create your own checked exceptions by extending Exception class

class UserNotFoundException extends Exception {   
 UserNotFoundException(String message) { … }   
}

We can use this constructor with String to pass error messages so that caller can get those error messages.

If(…) {   
 throw new UserNotFoundException(“User with “+name+” is not found”);  
}

Note: You must create all the exception classes related to your requirement in a separate package like com.npci.exceptions

Database

* Types of SQL
* Joins
* Views
* Stored Procedures
* Indexes

SQL: Structured Query Language which is used to communicate the RDBMS

SQL types

1. DDL (Data Definition Language): CREATE, ALTER, DROP, TRUNCATE
2. DML (Data Manipulation Language): INSERT, DELETE, UPDATE
3. DQL/DRL (Data Query/Retrieval Language): SELECT
4. TCL (Transaction Control Language): COMMIT, ROLLBACK, SAVEPOINT
5. DCL (Data Control Language): GRANT, REVOKE

Constraints: These are rules applied on the table & their data

1. PRIMARY KEY: No null values & must be unique
2. UNIQUE: Supports null values
3. NOT NULL: Needs value mandatorily
4. CHECK: Conditions on the value like gender must be male or female, age must be between 18 to 60
5. FOREIGN KEY: To reference the primary key of a table

SQL Queries

mysql> create database npci\_db;

Query OK, 1 row affected (0.00 sec)

mysql> use npci\_db;

Database changed

mysql> create table employees(employee\_id int primary key, employee\_name varchar(15), phone\_number int unique);

Query OK, 0 rows affected (0.04 sec)

mysql> insert into employees values(500, 'Alex', 99008800);

Query OK, 1 row affected (0.01 sec)

mysql> insert into employees values(501, 'Bruce', null);

Query OK, 1 row affected (0.00 sec)

mysql> insert into employees values(502, 'Charles', null);

Query OK, 1 row affected (0.00 sec)

mysql> select \* from employees;

+-------------+---------------+--------------+

| employee\_id | employee\_name | phone\_number |

+-------------+---------------+--------------+

| 500 | Alex | 99008800 |

| 501 | Bruce | NULL |

| 502 | Charles | NULL |

+-------------+---------------+--------------+

3 rows in set (0.00 sec)

mysql> update employees set phone\_number=8899888 where employee\_id=501;

Query OK, 1 row affected (0.01 sec)

Rows matched: 1 Changed: 1 Warnings: 0

mysql> update employees set phone\_number=8899888 where employee\_id=502;

ERROR 1062 (23000): Duplicate entry '8899888' for key 'employees.phone\_number'

mysql> update employees set phone\_number=8899898 where employee\_id=502;

Query OK, 1 row affected (0.00 sec)

Rows matched: 1 Changed: 1 Warnings: 0

mysql> alter table employees add column salary double;

Query OK, 0 rows affected (0.02 sec)

Records: 0 Duplicates: 0 Warnings: 0

mysql> select \* from employees;

+-------------+---------------+--------------+--------+

| employee\_id | employee\_name | phone\_number | salary |

+-------------+---------------+--------------+--------+

| 500 | Alex | 99008800 | NULL |

| 501 | Bruce | 8899888 | NULL |

| 502 | Charles | 8899898 | NULL |

+-------------+---------------+--------------+--------+

3 rows in set (0.00 sec)

mysql> alter table employees add column email\_id varchar(20), add column pan varchar(10);

Query OK, 0 rows affected (0.01 sec)

Records: 0 Duplicates: 0 Warnings: 0

mysql> desc employees;

+---------------+-------------+------+-----+---------+-------+

| Field | Type | Null | Key | Default | Extra |

+---------------+-------------+------+-----+---------+-------+

| employee\_id | int | NO | PRI | NULL | |

| employee\_name | varchar(15) | YES | | NULL | |

| phone\_number | int | YES | UNI | NULL | |

| salary | double | YES | | NULL | |

| email\_id | varchar(20) | YES | | NULL | |

| pan | varchar(10) | YES | | NULL | |

+---------------+-------------+------+-----+---------+-------+

6 rows in set (0.01 sec)

mysql> alter table employees drop column pan, drop column email\_id, drop column salary;

Query OK, 0 rows affected (0.01 sec)

Records: 0 Duplicates: 0 Warnings: 0

mysql> desc employees;

+---------------+-------------+------+-----+---------+-------+

| Field | Type | Null | Key | Default | Extra |

+---------------+-------------+------+-----+---------+-------+

| employee\_id | int | NO | PRI | NULL | |

| employee\_name | varchar(15) | YES | | NULL | |

| phone\_number | int | YES | UNI | NULL | |

+---------------+-------------+------+-----+---------+-------+

3 rows in set (0.00 sec)

mysql> select \* from employees;

+-------------+---------------+--------------+

| employee\_id | employee\_name | phone\_number |

+-------------+---------------+--------------+

| 500 | Alex | 99008800 |

| 501 | Bruce | 8899888 |

| 502 | Charles | 8899898 |

+-------------+---------------+--------------+

3 rows in set (0.00 sec)

mysql> select databases();

ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near 'databases()' at line 1

mysql> select database();

+------------+

| database() |

+------------+

| npci\_db |

+------------+

1 row in set (0.00 sec)

mysql> select PI();

+----------+

| PI() |

+----------+

| 3.141593 |

+----------+

1 row in set (0.00 sec)

mysql> select now();

+---------------------+

| now() |

+---------------------+

| 2023-10-13 17:44:02 |

+---------------------+

1 row in set (0.00 sec)

mysql> select current\_date;

+--------------+

| current\_date |

+--------------+

| 2023-10-13 |

+--------------+

1 row in set (0.00 sec)