GIRI's Tech Hub, Pune.

Programming Machine Test

Batch: Nov-23 to May-24 & June-24/July-24

Date : 27/12/2024
Time : 02 to 05 Pm.

Instructions:

1. **Solve any 9 questions.**
2. **Input should be from user.**
3. **Indentation and comments mandatory.**

Q1. Write a java program to print following pattern.

```
A
A B A
A B C B A
A B C D C B A
A B C D E D C B A
```

Q2. WAP to Create class name as StringOperation with a parameterized constructor.

StringOperation(String): this constructor can accept the string data.
displayAlphabet()
{
//this function use implement the logic.
Input : - Indian
Search Value : - d
Output : - n i
}

Q3. Create class name as ArrayOperation with method name as setArray() and create its Two child. Classes name as BiggerValue, SmallestMissing. We need to inherit the ArrayOperation class in BiggerValue, SmallestMissing and create function and write the logic.

1. BiggerValue Class : -

To print next greater elements in a given unsorted array. Elements for which no greater element exist, consider next greater element as -1.
Expected Output :
The given array is : 5 3 10 9 6 13
Next Bigger Elements are:
Next bigger element of 5 in the array is: 10
Next bigger element of 3 in the array is: 10
Next bigger element of 10 in the array is: 13
Next bigger element of 9 in the array is: 13
Next bigger element of 6 in the array is: 13
Next bigger element of 13 in the array is: -1
Next Bigger Elements Array:
10 10 13 13 13 -1

2. SmallestMissing class :-

Input :
The given array is : 0 1 3 4 5 6 7 9

Output :
The missing smallest element is: 2

Q4. Problem Statement:
Design an Employee Management System using both abstract classes and interfaces that performs the following operations:

Requirements:
1. Abstract Class: Employee

- **Fields:**
  - **int empId**
  - **String name**
  - **double basicSalary**
- **Constructor to initialize the fields.**
- **Abstract method:**
  - **void calculateNetSalary()**
- **Non-abstract method:**
  - **void displayEmployeeDetails()**

2. Interface: Taxable

- **Method:**
  - **double calculateTax(double netSalary);**

3. Subclasses:

- **Manager class (extends Employee, implements Taxable)**
  - **HRA = 20% of basic**
  - **DA = 15% of basic**
  - **Tax = 10% of net salary**
- **Clerk class (extends Employee, implements Taxable)**
  - **HRA = 10% of basic**
  - **DA = 5% of basic**
  - **Tax = 5% of net salary**

Operations to Perform in main()

1. **Accept details for multiple employees (both Manager and Clerk).**
2. **Calculate and display net salary.**
3. **Calculate and display tax using the calculateTax() method.**
4. **Display all employee details.**

Q5. Create a menu-driven program where the user can select an option to simulate different exceptions and handle them using try-catch blocks.

1. ArithmeticException:
2. NullPointerException:
3. NumberFormatException:
4. ArrayIndexOutOfBoundsException:
5. ClassNotFoundException:

Q6. Write a Java program to read a text file named input.txt, count the number of words, lines, and characters in it, and display the result.

Output :-

Lines: 5 , Words: 35 , Characters: 180

Q7. Write a Java program that performs the following operations using ArrayList:

- **Add 5 student names.**
- **Display the names.**
- **Remove the third name from the list.**
- **Display the updated list in reverse order.**

Q8. Create a Product class with fields: int id, String name, double price, and int quantity.

- **Implement two Comparators: one for sorting by price, and another for sorting by quantity.**
- **In the main method:**
  - **Add at least 5 products to an ArrayList.**
  - **Sort and display the list by price (ascending).**
  - **Then sort and display by quantity (descending).**
  - **Find and display the product with the maximum price.**

Q9. Create a Vehicle class with fields: String regNo, String model, String type, and int mileage.

- **Implement Comparable to sort vehicles by mileage ascending.**
- **Create a Comparator to sort by model.**
- **In main:**
  - **Add 6 vehicles.**
  - **Display vehicles sorted by mileage.**
  - **Display all "SUV" type vehicles.**
  - **Sort and display vehicles by model.**
  - **Display vehicle with highest mileage.**

Q10. Write a Java program that reads a sentence from the user and counts the frequency of each word using a HashMap. The program should:

- **Accept a sentence as input.**
- **Split the sentence into words.**
- **Use a HashMap to count how many times each word appears.**
- **Display each word and its frequency.**

Input : Java is easy and Java is powerful
Output : Word Frequencies: Java: 2
is: 2
easy: 1
and: 1
powerful: 1

-----ALL THE BEST-----