Department of computer Science and Engineering

National Institute of Technology Calicut

Monsoon Semester-2022-23

Evaluation:6 CS 4097D Object-Oriented Systems Lab Date:27/10/2022

1. Write a Java program to calculate the area and perimeter of a circle, rectangle and square. Your program should contain classes for representing the circle (Attributes: radius) , rectangle (Attributes: length, breadth) and square (Attributes: side). Each class should have respective functions for reading the attributes and for calculating the area and perimeter of the shape. Finally, the program should list all the shapes, which have an area greater than a specified value.

Sample input and output:

Enter the number of shapes: 3

Enter 1 for Circle, 2 for Rectangle, 3 for Square: 1

Enter Radius : 1

The area is 3.14 and the perimeter is 6.28

Enter 1 for Circle, 2 for Rectangle, 3 for Square: 2

Enter Length and Breadth: 5 4

The area is 20 and the perimeter is 18

Enter 1 for Circle, 2 for Rectangle, 3 for Square: 3

Enter Side: 2

The area is 4 and the perimeter is 8

Enter the threshold value: 15

The shapes are Rectangle(5,4)

2. Create a class Point which has attributes to store x coordinate and y coordinate values. Create another class Line which contains two objects of Point class. The first Point object holds the value of the line starting coordinates and the other object holds the line ending coordinates. In the implementation ask for the details about N Line objects and find out the line having smallest and largest length values. Make use of findLength() function inside the Point class to find the length of the line.

Sample input:

Enter the number of Line objects: 3

Enter Starting and Ending Coordinates for Line 1: 0 0 10 0

Enter Starting and Ending Coordinates for Line 2: 0 0 0 5

Enter Starting and Ending Coordinates for Line 3: 1 1 10 10

Output:

The largest line is Line 1 with length 10

The smallest line is Line 2 with length 5

3. The phone book contains the contact details of different persons. Each contact detail in the phone book keeps track of the name of the person, and associates the name with one phone number(A ten digit number). Identify the relationship between phone book and contact details, and properly organize the attributes.

The **PhoneBook** has the following functionalities:

- *PhoneBook*() : creates a new empty phone book.
- *insertEntry*(): Adds a new entry at the end of the phone book, if there is no contact already with the new contact's name and phone number.
- *lookUp*(substring): Searches the phone book for the contacts with the name containing the given substring.

Input Format

- First line contains the number of phone book entries, N.
- Next N lines of the input are space separated values String Integer
- Next line inputs P substring, where P indicates lookup.

Output Format

- If the input line contains P substring, then
    - print all the contacts whose name starts with substring. Each contact should be displayed in each line in the format name phone-number
    - print NoSuchEntry - If no contact is found in the phone book whose name starts with the given substring.
- Exceptions:-
    - Print IncorrectPhonenumber and terminate program - If the phone number is invalid.

Sample Input 0

6

Mukesh Kumar 9717635807

Anil Sharma 9810747515

Brijesh Kumar 9811296965
Sunita Sharma 8800767646
Kumar Karthik 8956774210
Dhiraj Kr. Sharma 9868585688
P Kumar

Sample Output 0

Kumar Karthik 8956774210

4. Create a java program with class account with properties account_holder_name, Balance and functions credit(), debit() and two subclass SBI and HDFC which inherit class account and with an extra function calculate_interest(). For SBI interest is 8% of balance and for HDFC 7% of balance). Input the details of two customers, one HDFC and one SBI, in the order account holder's name, initial balance, credited amount, debited amount. Output is balance and interest of both customers.

Sample input:

Enter SBI customer
details
Raju 1000 1500 500
Enter HDFC customer
details Aditya 2000
1500 500

Output:

Raju 2000 160
Aditya 3000 210

5. From the below given description, identify the different classes and the inheritance relationship among them. Implement the java program according to that inheritance hierarchy. The program should print the salary of Security and HOD objects using object oriented concepts.

Employee is a person (name, age) with additional properties emp_id, salary_per_day and no_of_days_present.

Security is an employee with additional property shift (can have only two values “Day” or “Night”. Use enum to save shift details).

Teacher is an employee with branch and specialization.

HOD is a teacher with additional attributes number_of_teachers_working_under and number_of_guideship.

Implement the displaySalary() function for both Security and HOD which displays the final salary. The basic salary is calculated as salary_per_day*no_of_days_present.

For Security, if his shift is on Day, then additional 200 will be added with basic salary.

For HOD, based on the number_of_teachers_working_under, the additional salary is calculated as 50*number_of_teachers_working_under.

Sample input/Output:

1- Insert HOD details
2- Insert security details
3- Display HOD details
4- Display Security details
5 – Exit

Sample Input 0

1

Suresh 30 1023 2000 30 cse networking 20 5

(name , age, emp_id, salary_per_day, No_of_days_present, branch,

specialization , number_of_Teachers_working_under, number_of_guideship)

2

Raju 35 7823 500 25 Day(name , age, emp_id, salary_per_day, No_of_days_present, shift)

3

4

5

Sample Output 0

Suresh 61000

Raju 12700