CS162FZ Introduction to Computer Science II

Lab 6

2021-5-23

Task1: Binary Search

In this guessing game, a random number n in the range x, y will be chosen. The computer must then find the number. At the end of the game, the number is revealed along with the number of tries it took the computer to get the correct number. The computer should use the binary search algorithm to solve this question.

Input

x, y, n, where all values are integers, x and y is the search range and n is the number you will search for.

Sample Input

```
1
10
7
```

Output

The number being searched for and the number of tries to find it

Sample Output

```
The number is: 7
It took 2 tries!
```

Task2: Circle

a) Write a class to represent a Circle called BasicCircle. This should include the following:

i. A single attribute named radius.

ii. A default constructor which will set the value of radius to 1.

iii. A general constructor which should set the radius to a value which is passed to the constructor.

iv. Getter and setter methods for the radius attribute.

v. A method called getCircumference to calculate and return the circumference of the Circle to 2 decimal places (see appendix).

You should use Math.PI in this method.

vi. A method called getArea to calculate and return the area of the Circle to 3 decimal places (see appendix).

You should use Math.PI in this method.

b) Write a main method in a new class called TestBasicCircle and in this method: i. Create **three** BasicCircle objects.

ii. The radius of each circle should be set by user input.

iii. Print the radius of each circle object (see sample output below)

iv. Use the getCircumference()method and print the circumference of each circle.

v. Use the getArea() method and print the area of each circle. (see sample output below)

Output

The TestBasicCircle class should print to the screen the correct message to match each part of the questions given above.

Sample Output - output should be exactly as below

```
Circle 1 has a radius of 5
Circle 2 has a radius of 7
Circle 3 has a radius of 3
Circle 1 has a circumference of 31.42
Circle 2 has a circumference of 43.98
Circle 3 has a circumference of 18.85
Circle 1 has an area of 78.54
Circle 2 has an area of 153.94
Circle 3 has an area of 28.27
```

Appendix

num1 will have a value of 43.36 after the piece of code is run.

```
double num1=Math.round(43.3565656 * 100.0) / 100.0;
```

To use Math.round & Math.pow include:

```
import java.lang.Math.*;
```

Task3: Employee

a) Write a class to represent an Employee. This should include the following:

i. Attributes for the Employee’s name (String), jobTitle (String), salary (float) and a unique employeeID (int). The salary should have a minimum value of 19000f.

ii. A default constructor which should set default values for all attributes.

iii. A second constructor which accepts and sets values for each of the attributes.

iv. Getter and setter methods for the name, jobTitle and salary.

v. A Getter method for the employeeID

vi. A method called showEmp() that takes no parameters, returns nothing and prints the information for each employee to the screen. (See sample output below).

vii. A method called toString()that returns a string representation of the attributes of an Employee (see sample output below for format)

b) Write a main method in a new class called TestEmployee and in this method:

i. Declare 3 Employee objects. Use the General Constructor for the first 2 Employees and the default constructor for the last employee. Details for Employee 1 and 2 are as follows:

ii. Call the getName(), getJobTitle(), getSalary() and getEmployeeID() methods to print the information of the 2nd Employee to the screen (it should be displayed exactly as the sample output below).

iii. Call the setName(<…>), setJobTitle(<…>) and setSalary(<…>) method to change the information on the 3rd Employee to the following details:

iv. Use the showEmp() method to print all the information for all People to the screen (it should be displayed exactly as the sample output below).

v. Call the toString() method on the 1st Employee object.

Num	Name	jobTitle	salary
1	“Tom Daly”	“Teacher”	45000
2	“Mary Murphy”	“Developer”	30000

Output

The TestEmployee class should print to the screen the correct message to match each question given above.

Sample output

```
Employee 2 is called Mary Murphy They are a Developer
Their salary is 45000.0
Their Employee ID number is 1002
Entered salary of 1000.0 is too low. Salary set at minimum wage of €19000.0 EMPLOYEE RECORD
NAME: Tom Daly JOB TITLE: Teacher
CURRENT SALARY: 30000.0
EMPLOYEE ID: 1001
EMPLOYEE RECORD
NAME: Mary Murphy JOB TITLE: Developer
CURRENT SALARY: 45000.0
EMPLOYEE ID: 1002
EMPLOYEE RECORD
NAME: Michael Higgins JOB TITLE: President
CURRENT SALARY: 19000.0
EMPLOYEE ID: 1003
NAME: Tom Daly JOB TITLE: Teacher
CURRENT SALARY: 30000.0
EMPLOYEE ID: 1001
```

Task 4: Square

1. Write a class to represent a Square. This should include the following: 1.1. A single attribute named length

1.2. A default constructor which should set the length attribute to a default value of 1.

1.3. A second constructor which should set the length attribute to a value which is passed to the constructor.

1.4. Getter and setter methods for the length attribute.

1.5. A method, called getArea(), to calculate and return the area of the Square.

1.6. A method called toString() that returns the value of the attribute length (see sample output below for format)

2. Write a main method in a new class called TestSquare and in this method. 2.1. Create an array containing 4 square objects.

2.2. Use a ‘for’ loop to set each object to default values when created.

2.3. Use the setter method on the 1st and 3rd objects in the array of Square objects to set the length to 10.

2.4. Print out the length of each of Square object in the array.

2.5. Print out the area of the 1st and 2nd objects in the array.

2.6. Call the toString() method on the 3rd Square object in the array.

Output
The TestSquare class should print to the screen the correct message to match each question given above.
Sample output

```
Square 1 has a side length of 10
Square 2 has a side length of 1
Square 3 has a side length of 10
Square 4 has a side length of 1
Square 1 has an area of 100
Square 2 has an area of 1
LENGTH: 10
```

Task 5: Movie

You are given a class Movie.java. The class represents a Movie. Every Movie has a name (movieName), a year the movie was released (releaseYear), a movie Rating (rating) and a unique movieID. You are required to add a toString() method to the movie class that will print out the Movie Details.

The SortMovie.java file will read in a list of movies from a Movies.txt. This will store the movies in a String array. Each String will look like the following:

9.2,The Shawshank Redemption,1994

You need to make a Movie array and split the strings from this array to create a new Movie object for each index of the Movie array.

You can use <String>.split(“,”) to help you achieve this. (See https://beginnersbook.com/2013/12/java-string-split-method-example/ for more information) You are also provided a Bubble sort method to sort the Movie array from Highest to Lowest according to releaseYear. You are required to add five other Methods to this program that will sort the Movie Array From Highest to Lowest and Lowest to Highest based on movieName, releaseYear, and rating.

You should take input from the User on How they want the Movie array to be sorted and then print the array after it is sorted.