

Story 19/2024
18/09/2024

1) Aim: Java program to create a class student with constructor than takes in a name.

Pseudo code:-

class student;

declare variables:

name (string)

id (int)

grade [] (char array)

ang (int), initially 0

constructor: student (name, id, grade)

set sum to 0

set count to 0

Assign name to the provided name

Assign id to the provided id

Assign grade to provided grade

for each char c in grade:

increment count by 1

After it is ..

Procedure?
Initialize (variables)
key: NAME, ID, grade
Constructor
Set sum

if
 Add .9 to sum
else if c is 'B'
 Add .8 to sum
else if c is 'C'
 Add .7 to sum
calculate the avg by counting the
result of sum divided by count
and display()

Aim

class called car with constructor

pseudo code:-

class car?

variables:

string make

string model

int year

Constructor

input: make, model, year.

set class variables to input

variables

Method print car details

Output: display make, model, and
year of the car

Main;

Create an object of car with make,
model and year.

call print car details method to
display car details.

3) Bank

Pseudo code:-

class BankAccount:

 int accountNumber

 double balance

 constructors

 input : accountNumber ? initial

 balance

 set account Number and balance to

 input values

 Method deposit :

 input : amount

 if amount is less than or equal

 to balance subtract amount

 from balance

 else

 display "Insufficient Balance"

 Main :

 create an object of Bank Account with

 account number and initial Balance

 call deposit method to money

 call withdraw method to take out
 money

4) Rectangle

pseudo code:

class rectangle;

variables:

double length

double width

constructor:

input: length width

set length and width to input

values

method calculateArea:

returns length * width

Main:

create an object of rectangle

with length and width

call calculateArea method to

get and display the area.

5) Person

Pseudo code:

class Person:

variables:

string name

int age

constructor

input: name, age.

set . class-variable. to input value

method . print . person details.

output: display . name and age

Main:

create. an . object of person with

name . and age.

call . print person . details . methods

to . display . the . person's name and

age

6) method overriding - with a simple

calculator . to .

Pseudocode: `Normal . add`

method . add :

input: two numbers

return: sum of the two number

method .. subtract:

input: two numbers - obiect

return: difference . between . the two
numbers

method multiply:

input: two numbers

return: difference . b/w the two num

method.
input : two numbers
if denominator is not zero,
return : quotient
else
display : can't divide by zero
calculate (inherits calculator)
override . method . multiply ;
input : two numbers
return : more complex
calculation

main :
create an object of calculate

class

call . add ; sub ; multiply . and divide

7) . method overloading

class calculator :

method . sum (int ... sum)

initialize total as 0

for each integer in num :

Add integer to total

return total

overloaded method . sum (double

... num) ;

initialize total as 0

for each double in numbers;
· add the double to total
· return total

main:

Create an object of calculator

class call sum method with a variable
number of integers
call overloaded sum method with
a variable number of doubles

8) Polymorphism

Abstract class Animal;

No implementation (to be implemented
by child class)

Abstract method sleep();

No implementation

class Dog (inherits Animal);

Implement method eat();

display: "Dog is eating"

Implement method sleep();

display: "Dog is sleeping"
class eat (inherits Animal);

Implement method eat();

interface .Drawble;
method draw();
no implementation.

class .Circle;

implement .method .draw;

display : "Drawing .a circle"

class .Square;

implement .method .draw

display : "Drawing .a square"

main;

create .an object .of .circle class

call .draw .method .(circle)

create .an .object .of .square

class

10) Pseudo code;

class .Shape;

variables?

double area();

double perimeter.

method calculate .Area();

NO implementation

method calculatePerimeter();

NO implementation

Method getArea();

return area;

Method getPerimeter();

return perimeter();

class Circle;

variables:

double radius

constructors

input: radius

set radius and calculate area and perimeter

✓
Solve

1) Count no of occurrences
prime number

sum of first n prime numbers