Date 26-09-2024

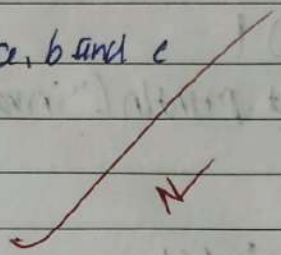
Develop a Java program that prints all real solutions to the quadratic equation $ax^2+bx+c=0$. Read in a, b, c and use the quadratic formula. If the discriminant $b^2-4ac$ is negative display a message.

```
impurt java.util.Scanner;
class quadratic{
float d;
Scanner sc = new scanner(system.in);

void check()
{
system.out.println("enter the values of a, b and c");
  int a = sc.nextInt();
  int b = sc.nextInt();
  int c = sc.nextInt();

 if(a==0){
 System.out.println("invalid equation");
}
else{
   d = b*b-4*a*c;

system.out.println(" the solutions are");
 if(d>0){
   System.out.println(" the solutions are");
 system.ow.println("roots are unique");
  double r1 =(-b+Math.sqrt(d))/(2*a);
  double r2=(-b-Math.sqrt(d))/(2*a);
  system.out.println(r1+" "+r2);
}
if(d==0){
  system.out.println("roots are equal");
   double r = -b/(2*a);
  system.out.println(r2+"+i"+r1+" "+r2+"-i"+
                                    r1);
```

```
}
}
}
}

public class main{
  public static void main(String[] args) {
     quadratic q1 = new quadratic();
     q1.check();
}
}
```

output:-

Enter the values of a, b and c
10
20
30



the solutions are
both are imaginary
-1.0+11.414213562373095i

Date: 3-10-2024

Bafna Gold

Develop a Java program to create a class student with members usn, name, an array credits and an array marks include methods to accept and display details and a method to calculate SGPA of a student.

```
import java.util.Scanner;
public class cgpa{
public static void main(String[] args){
    Scanner sc= new Scanner(System.in);

System.out.print("Enter the number of subjects:");
 int num subjects = sc.nextInt();

double [] gradepoints = new double [numsubjects];
int [] credits = new int [numsubjects];
 int total credits =0;
 double total=0;

for(int i=0; i<numsubjects; i++){
System.out.print("Enter the grade points for subject"+
      (i+1)+": ");
   gradepoints[i] = sc.nextDouble();

System.out.print("Enter the credits for subject"+
      (i+1)+": ");
   credits[i]= sc.nextInt();

total += gradepoints[i] * credits[i];
 totalcredits += credits[i];
}
double cgpa = total/ total credits;
system.out.print("Your cgpa is:" + cgpa);
```

}
}

Output:

Enter USN: 1bm23cs319
Enter Name: shreyas
Enter the number of subjects: 3
Enter credits for subject 1: 4
Enter the marks for subject 1: 3
Enter the credits for subject 2: 2
Enter the marks for subject 2: 89
Enter credits for subject 3: 4
Enter marks for subject 3: 98

Student details:
USN: 1bm23cs319
Name: shreyas
Subject and marks:
Subject 1: Marks = 3, credits = 4
subject 2: marks = 89, credits = 2
Subject 3: Marks = 98, credits = 4

SGPA: 5.8

Date: 19-10-2024

Bafna Gold

Create a class Book which contains four members: name, author, price, num_pages. Include a constructor to set the values for the members. Include the methods to set and get the details of the object. Create a Java Pg to n Book objs.

```
import java.util.Scanner;

class Book {
    int price;
    string author;
    string name;
    int pages;

public Book (int price, string author, string name, int pages)
{   this.price = price;
    this.author = author;
    this.name = name;
    this.pages = pages;
}

public void setter() {
System.out.println("enter the price, author, name and
                    pages of the book");
    Scanner sc = new Scanner (System.in);
     int price = sc.nextInt();
     string author = sc.nextInt();
     string name = sc.nextInt();
     int pages = sc.nextInt();
  }


public void getter() {
    system.out.println(price);
    system.out.println(author);
```

```
system.out.println(name);
system.out.println(pages);
}

public string toString(){
    return "this is object";
}
}

public class pro{
  public static void main(String[] args){
    Scanner s1 = new Scanner(System.in);
System.out.println("enter the number of objects");
    int n=s1.nextInt();

Book[] b1 = new Book[n];
for(int i=0; i<n; i++){
 b1[i] = new book(200, "virat", "the century", 111);

 b1[i].getter();
 b1[i].setter();
 b1[i].getter();

system.out.println(b1[i]);

  }

}

}
```

output

Enter the number of objects
2
Enter the price, author, name and page of the book
999
shreyas
goat
219
999
Shreyas
goat
969
this is object
enter the price, author, name and page of the book
899
sanath
old
49
899
sanath
old
49
this is object

Date 24-10-2024

4) Develop a Java program to create an abstract class named shape that contains two integers and an empty method named printArea(). Provide three classes named rectangle, Triangle and circle such that each one of the class extends the class shape. Each one of the class contains only the method printArea() that prints the area of the given shape.

```
abstract class shape {
 int dim1;
 int dim2;

 abstract void printArea();
}
class Rectangle extends shape {
 public Rectangle(int length, int width) {
      this.dim1 = length;
      this.dim2 = width;
}

 void printArea() {
 int area = dim1 * dim2;
 System.out.println("Area of Rectangle." + area);
 }
}

class Triangle extends shape {
  public Triangle(int base, int height) {
     this.dim1 = base;
     this.dim2 = height;
```

```
}

void printArea() {
  double area = 0.5 * dim1 * dim2;
System.out.println("Area of Triangle:" + area);
 }
}

class circle extends Shape{
  public circle(int radius) {
   this.dim1 = radius;
  this.dim2 = 0;
}

 void printArea() {
double area = Math.PI * dim1 * dim1;
System.out.println("Area of circle:" + area);
 }
}

public class Main {
 public static void main(String[] args) {
  Shape rectangle = new Rectangle(10, 5);
 Shape triangle = new Triangle(8,2);
 Shape circle = new circle(7);

  rectangle.printArea();
  triangle.printArea();
  circle.printArea();
 }
}
```

Output:

Area of Rectangle: 50
Area of Triangle: 24.0
Area of circle: 153.93804002589985

Observed
07/11/24

Date 7-11-2024.

Bafna Gold

Develop a java program to create a class Bank that maintains two kinds of account for its customers one called saving account and the other current account That Savings account provides compound interest and withdrawal facilities but no. cheque book facilities The current account provide cheque book facilities but no interest, current account holds should also maintain a minimum balance and if the balance falls below. this level, a service charge is imposed create a class Account that stores customer name, acc num and type of acc. from this derive the classes current and sav-acct to make them more specific to their requirements include the necessary methods in order to achieve the following.

a) Accept deposit from customer and update the balance
b) display the balance.
c) compute and deposit interest
d) permit withdrawal and update the balance:
check min. imposed penalt if necessary and update the bal.

```
import java.util.Scanner;
class Account{
String customerName;
int accountNumber;
String accountType;
double balance;

public Account(String customerName, int accountNumber,
String accountType){
    this.customerName = customerName;
```

```
this.acc
  this.accountNumber = accountNumber;
  this.accountType = accountType;
  this.balance = 0.0;
}

public void deposit(double amount) {
   if (amount > 0) {
   balance += amount;
System.out.println("Amount deposited: " +
amount);
 System.out.println("Updated balance: " + balance
);
} else {
   System.out.println("Invalid deposit amount!");
}
}

public void displayBalance() {
  System.out.println("Balance: " + balance);
}
}

class SavAcct extends Account {
private double interestRate;

public SavAcct(String customerName, int accountNumber,
  double interestRate) {
  super(customerName, accountNumber, "Savings");
  this.interestRate = interestRate;
}
```

```
public void compoundAndDeposit(){
 double interest = balance * (interestRate/100);
 balance += interest;
 System.out.println("Interest added: "+interest);
 System.out.println("Updated balance: "+balance);
 }

public void withdraw(double amount){
   if (amount <= balance){
    balance -= amount;
 System.out.println("Amount withdrawn: "+amount);
 System.out.println("Updated balance: "+balance);
 } else {
  System.out.println("Insufficient balance");
  }
 }
}

class currAcct extends Account {
 double minimumBalance;
 double serviceCharge;

public currAcct(String customerName, int accountNumber,
 double minimumBalance, double serviceCharge) {
 super(customerName, accountNumber, "current");
 this.minimumBalance = minimumBalance;
 this.serviceCharge = serviceCharge;
 }

public void withdraw(double amount) {
   if (amount <= balance){
    balance -= amount;
```

```
System.out.println("Amount withdrawn: " + amount);
if (balance < minimumBalance) {
    imposePenalty();
}
System.out.println("Updated balance: " + balance);
} else {
System.out.println("Insufficient balance.");
}
}

void imposePenalty() {
balance -= serviceCharge;
System.out.println("Balance fell below minimum. Service charge imposed: " + serviceCharge);
}
}

public class Bank {
public static void main(String[] args) {
Scanner scanner = new Scanner(System.in);
System.out.println("Choose account type: 1. Savings Account 2. Current Account");
int choice = scanner.nextInt();
scanner.nextLine();

System.out.println("Enter name: ");
String name = scanner.nextLine();
System.out.println("Enter account number: ");
int accNum = scanner.nextInt();

if (choice == 1) {
System.out.println("Enter interest rate for
```

```
saving account :");
double interestRate = scanner.nextDouble();
SavAcct savAccount = new SavAcct(name, accNum, interestRate);

system.out.println("Enter amount to deposit :");
double deposit = scanner.nextDouble();
savAccount.deposit(deposit);

savAccount.computeAndDepositInterest();
System.out.println("Enter amount to withdraw :");
double withdrawAmount = scanner.nextDouble();
  savAccount.withdraw(withdrawAmount);
}

else if (choice == 2) {
system.out.println("Enter minimum balance for current ac :");
  double minBalance = scanner.nextDouble();
system.out.println("Enter service charge for falling below
minimum balance:");
double serviceCharge = scanner.nextDouble();
CurAcct curAccount = new CurAcct(name, accNum, minBalance,
serviceCharge);

system.out.println("Enter amount to deposit :");
  double deposit = scanner.nextDouble();
curAccount.deposit(deposit);

System.out.println("Enter amount to withdraw:");
  double withdrawAmount = scanner.nextDouble();
curAccount.withdraw(withdrawAmount);
else {
  system.out.println("Invalid account type selected.");
```

```
}
cancurrentModel();
}
}
```

output:

```
choose account type:
1. Savings Account
2. Current Account
1
Enter customer name:
Shreyas
Enter account number:
BDS90
Enter interest rate for savings account:
10
Enter amount to deposit :
20000
```

Amount deposited: 20000.0
updated balance: 20000.0
Interest added: 2000.0
Updated balance: 22000.0
Enter amount to withdraw:
2000
Amount withdrawn: 2000.0
Updated balance: 20000.0

Create a package CIE which has two classes Student and Internals. The class personal has members like usn, name, sem. The class internals has an array that stores the internal marks scored in 5 courses of the current semester of the student. Create another package SEE which has the class External which is a derived class of student. This class has an array that stores the SEE marks scored in five courses of the current semester of the student. Import the two packages in a file that declares the final marks of n students in all five courses

Package CIE :

```
public class Student
    String name;
    int usn;
    int sem;

public Student (String name, int usn, int sem) {
    this.name = name;
    this.usn = usn;
    this.sem = sem;
}
}
```

```
package CIE;

public class internals extends Student {

    public int marks[] = new int[5];

    public internals(String name, int usn, int sem, int[]
            marks) { super(name, usn, sem);
        this.marks = marks;
    }
}
```

```
package SEE;
import CIE.Student;

public class externals extends Student {
    public int smarks[] = new int[5];

    public externals(String name, int usn, int sem,
            int[] marks) { super(name, usn, sem);
        this.smarks = marks;
    }
}
```

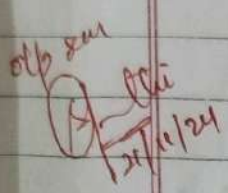
```
import CIE, internals;
// import SEE, externals;
import java.util.Scanner;

public class cut {
    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        int cmarks[] = new int[5];
        int smarks[] = new int[5];
System.out.println("enter the num of student");
int n = sc.nextInt();
for(int k = 0; k<n; k++) {

System.out.println("enter the name of the student");
        String name = sc.next();
        System.out.println("enter the usn of the student");
        String usn = sc.next();
        System.out.println("enter the sem of the student");
        int sem = sc.nextInt();

System.out.println("enter the cie marks of 5 subject");
    for(int i=0; i<5; i++) {
        cmarks[i] = sc.nextInt();
    }
System.out.println("enter the see marks of 5 subject");
for(int i=0; i<5; i++) {
    smarks[i] = sc.nextInt();
    }
internals oi = new internals(name, usn, sem, cmarks);
externals o2 = new externals(name, usn, sem, smarks);
```

```
for (int i=1; i<=5; i++){
System.out.println("the total marks of students
    in sub" + i);
System.out.println(o1.marks[i-1] + o2.marks(i-1));
}
}
}
```

output

enter the num of students
2
enter the name of the student
sanath
enter the rno of the student
123456
enter the sno of the student
1
enter the six marks of 5 subjects
49
48
49
50
50
enter the sec marks of 5 subjects
50
50
50
50
50
the total marks of student in sub 1
99

the total marks of student in sub 2
98
the total marks of student in sub 3
99
the total marks of student in sub 4
100
the total marks of student in sub 5.
100



7) write a program that demonstrates handling of exceptions in inheritance tree. Create a base class called "father" and derived class called "son" which extends the base class. In father class, implement a constructor which take the age and throws the exception WrongAge() when the input age < 0. In son class, implement a constructor that takes both father and son's age and throws an exception if son's age >= father age.

```
class WrongAgeException extends Exception {
public WrongAgeException (String message) {
        super (message);
  }
}

class father {
  int fatherAge;

public father(int fatherAge) throws WrongAgeException
  if (fatherAge < 0) {
   throw new WrongAgeException ("Fathers age
  cannot be negative.");
  }
  this.fatherAge = fatherAge;
  }
}

class Son extends father {
 int sonAge;
```

```
public Son(int fatherAge, int sonAge) throws WrongAge
Exception {super(fatherAge);
 if (sonAge >= fatherAge) {
  throw new WrongAgeException("Son's age cannot
be greater than or equal to fathers age.");
}
 if (sonAge < 0) {
  throw new WrongAgeException("Son's age cannot
be negative.");
 }
 this.sonAge = sonAge;
 }
}

public class ExceptionHandlingDemo {
  public static void main(String[] args) {
   try {
 System.out.println("Creating a Father object...");
  Father father = new Father(40);
 System.out.println("Father created with age:" + father.fatherAge);

 System.out.println("Creating a Son object...");
  Son son = new Son(40, 20);
 System.out.println("Son created with age:" + son.sonAge);
} catch (WrongAgeException e) {
   System.out.println("\n Testing invalid scenarios...");
 try {
  Father invalidFather = new Father(-10);
 } catch (WrongAgeException e) {
  System.out.println("Exception caught:" + e.getMessage());
 }
```

```
try {
    Son invalidSon = new Son(30,40);
} catch (WrongAgeException e) {
    System.out.println("Exception caught = " +
    e.getMessage());
    }
}
```

output

Enter fathers age 50
Enter son's age 60
Son's age cannot be greaterthan or equal to fathers age
SonAgeException

write other two cases

21/11/24

28-11-2021

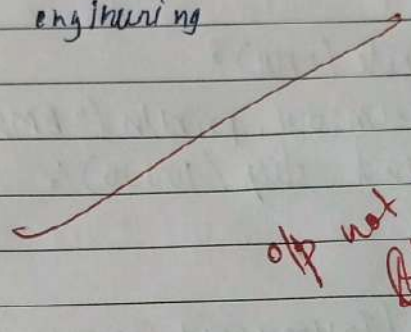
Write a program which creates two threads, displaying "BMS college of engineering" once every ten seconds and another displaying "CSE" every two seconds.

```
class Thread {
public static void main (String args[]) {
Threads threadBMS = new Thread (new DisplayBMS());
Thread threadCSE = new Thread (new DisplayCSE());
threadBMS.start();
threadCSE.start();
}
}

class DisplayBMS implements Runnable {
  public void run() {
  try {
  while (true) {
  System.out.println("BMS College of Engineering");
  Thread.sleep(10000);
  }
}
catch (InterruptedException e) {
  System.out.println("Interrupted" + e.getMessage());
}
}
}

class DisplayCSE implements Runnable {
public void run() {
  try {
  while (true) {
      system.out.println("CSE");
      Thread.sleep(2000);
```

```
    }
}
catch (InterruptedException e) {
System.out.println("Interrupted" + e.getMessage());
    }
  }
}
```

output:

```
BMS college of engineering
CSE
CSE
CSE
CSE
BMS college of engineering
CSE
CSE
CSE
CSE
CSE
```



o/p not seen

14/12/24