1.Swapping of two numbers using Abstract class

import java.util.\*;

abstract class swap {

int x,y;

swap(int x,int y)

{

this.x=x;

this.y=y;

}

public abstract void swapno();

}

class swapnum extends swap

{

swapnum(int x,int y)

{

super(x,y);

}

public void swapno()

{ int c=x; x=y; y=c;

} }

class funzone

{

public static void main(String ar[])

{ int a,b; Scanner sc=new Scanner(System.in);

System.out.print(“Enter a:”);

a=sc.nextInt();

System.out.print(“Enter b:”);

b=sc.nextInt();

System.out.println(“Before Swap a:” + a + “,b:” +b);

swapnum ob = new swapnum(a,b);

ob.swapno();

System.out.println(“After Swapping a:”+ ob.x + “,b:” + ob.y); } }

Output:

Enter a:6

Enter b:8

Before Swap a:6,b:8

After Swapping a:8,b:6

[Program finished]

2.Write a program in Fibonacci series, Odd or Even, Simple Interest Using Dynamic dispatch

import java.util.\*;

class A{

Scanner sc=new Scanner(System.in);

void exampleMethod(){

int a=0,b=1,c=0;

System.out.print(“\nEnter the number: “);

Int n=sc.nextInt();

if(n==1)

System.out.println(“Fibonacci series of 1 is 0”);

else if(n==2)

System.out.println(“Fibonacci series of 2 is 1”);

else{

for(int i=2;i<n;i++){

C=a+b;

A=b;

B=c;

}

System.out.println(“Fibonacci series of “+n+” is “+c);

}

}

}

class B extends A{

void exampleMethod(){

System.out.print(“\nEnter the number: “);

int n=sc.nextInt();

if(n%2==0)

System.out.println(n+” is an Even number”);

else

System.out.println(n+” is an Odd number”);

}

}

class C extends B{

void exampleMethod(){

System.out.print(“\nEnter the Principle amount (P): “);

int p=sc.nextInt();

System.out.print(“\nEnter number of times (N): “);

int n=sc.nextInt();

System.out.print(“\nEnter the rate of interest : “);

int r=sc.nextInt();

System.out.println(“\n\t SIMPLE INTEREST = “+((p\*n\*r)/100));

}

}

class Main{

public static void main(String[] args) {

A a=new A();

B b=new B();

C c=new C();

A r;

R=a;

r.exampleMethod();

r=b;

r.exampleMethod();

r=c;

r.exampleMethod();

}

}

Output:

Enter the number: 5

Fibonacci series of 5 is 3

Enter the number: 4

4 is an Even number

Enter the Principle amount (P): 5000

Enter number of times (N): 5

Enter the rate of interest : 8

SIMPLE INTEREST = 2000

[Program finished]

3.Binary search using Constructor

import java.util.\*;

class bi

{

public static int bsearch(int a[],int x,int low,int high)

{

int mid;

while(low<=high)

{

mid=(low+high)/2;

if(a[mid] == x)

return mid;

If(x<a[mid])

high=mid-1;

else

Low=mid+1;

}

return -1;

}

bi()

{

int a[]=new int[20];

int n,se,I,res;

System.out.println(“Binary search”);

Scanner sc=new Scanner(System.in);

System.out.println(“Enter limit : “);

n=sc.nextInt();

System.out.println(“Enter values : “);

for(i=0;i<n;i++)

a[i]=sc.nextInt();

System.out.println(“Enter search value : “);

se=sc.nextInt();

res=bsearch(a,se,0,n-1);

if(res==-1)

System.out.println(“Not found”);

else

System.out.println(“Found”);

}

public static void main(String args[])

{

bi o=new bi();

}

}

Binary search

Enter limit :

5

Enter values :

8 9 10 11 12

Enter search value :

12

Found

[Program finished]

Binary search

Enter limit :

5

Enter values :

8 9 10 11 12

Enter search value :

7

Not found

[Program finished]

5.Stack using array

import java.io.\*;

import java.util.\*;

class Operation{

Scanner sc = new Scanner (System.in);

int n = 3;

int stack[] = new int [n];

int top = -1;

void push(){

if(top == n-1){

System.out.print(“\nStack is overflow”);

}

else{

System.out.print(“\nEnter the number : “) ;

int x = sc.nextInt();

stack[++top]=x;

System.out.print(“The value “+x+” is pushed”);

}

}

void pop(){

if(top==-1){

System.out.print(“\nStack is underflow”);

}

else {

System.out.print(“The value “+stack[top]+” is popped”);

top--;

}

}

void display(){

if(top==-1){

System.out.print(“\nStack is underflow”);

}

else {

for(int I = top;i>=0;i--){

System.out.print(“\n “+stack[i]);}

}

}

}

class Opselection{

public static void main(String[] ar){

Scanner sc = new Scanner (System.in);

Operation op = new Operation();

while(true){

System.out.print(“\n\t\t\tStack Operation “);

System.out.print(“\n1.Push\n2.Pop\n3.Display\n4.Exit”);

System.out.print(“\nEnter the choice : “);

int ch = sc.nextInt();

switch(ch){

case 1:

op.push();

break;

case 2:

op.pop();

break;

case 3:

op.display();

break;

case 4:

System.exit(0);

default :

System.out.print(“\nInvalid Choice”);

break;

}

}

}

}

Output:

Stack Operation

1.Push

2.Pop

3.Display

4.Exit

Enter the choice : 1

Enter the number : 5

The value 5 is pushed

Stack Operation

1.Push

2.Pop

3.Display

4.ExitEnter the choice : 1

Enter the number : 7

The value 7 is pushed

Stack Operation

1.Push

2.Pop

3.Display

4.ExitEnter the choice : 1

Enter the number : 4

The value 4 is pushed

Stack Operation

1.Push

2.Pop

3.Display

4.ExitEnter the choice : 3

4

7

5Stack Operation

1.Push

2.Pop

3.Display

4.Exit

Enter the choice : 2

The value 4 is poppedStack Operation

1.Push

2.Pop

3.Display

4.Exit

Enter the choice : 2

The value 7 is popped

Stack Operation

1.Push

2.Pop

3.Display

4.Exit

Enter the choice : 2

The value 5 is popped

Stack Operation

1.Push

2.Pop

3.Display

4.Exit

Enter the choice : 3

Stack is underflow

Stack Operation

1.Push

2.Pop

3.Display

4.Exit

Enter the choice : 4

6.Recursion using Sum of Digits, Duplicate number removal method overloading.

import java.util.\*;

import java.io.\*;

public class Load

{

int meth(int n)

{

if(n==0)

return 0;

return(n%10+meth(n/10));

}

void meth()

{

int [] a=new int[] {1,2,3,1,4,7,2,8,4};

System.***out***.println("duplicate elements in given array:");

for(int i=0;i<a.length;i++)

for(int j=i+1;j<a.length;j++)

{

if(a[i]==a[j])

System.***out***.println(a[j]);

}

}

public static void main(String[] args)

{

Scanner sc=new Scanner(System.***in***);

Load obj=new Load();

System.***out***.println("enter the value for to find sum of digits:");

int x=sc.nextInt();

int a=obj.meth(x);

System.***out***.println(a);

obj.meth();

}

}

Output:

enter the value for to find sum of digits:

123

6

duplicate elements in given array:

1

2

4

[Program finished]

7.Command line argument

class A{

public static void main(String args[]){

for(int i=0;i<args.length;i++)

System.out.println(args[i]);

}

}

1. compile by > javac A.java
2. run by > java A This is a test

Output:

This

is

a

test

8.Write a program in prime number using innerclass.

import java.util.\*;

class outer {

void test() {

int n, f = 0, i;

Scanner sc = new Scanner(System.in);

System.out.println("Enter a number:");

n = sc.nextInt();

for (i = 1; i <= n; i++) {

if (n % i == 0)

f++;

}

class InnerClass {

void display(int f) {

if (f == 2)

System.out.println("The number is prime");

else

System.out.println("The number is not prime");

}

}

InnerClass in = new InnerClass();

in.display(f);

}

public static void main(String args[]) {

outer o = new outer();

o.test();

}

}

Output:

Enter a number:

5

The number is prime

[Program finished]

9.MATRIX MULTIPLICATION PASSING OBJECT AS A PARAMETER

import java.util.\*;

class Matrix

{

int I,j,k;

int a[][]=new int[3][3];

void a()

{

System.out.println(“Matrix multiplication”);

Scanner sc=new Scanner(System.in);

System.out.println(“a value:”);

for(i=0;i<3;i++){

for(j=0;j<3;j++){

a[i][j]=sc.nextInt();

}

}

System.out.println(“Resultant matrix of a:”);

for(i=0;i<3;i++){

for(j=0;j<3;j++){

System.out.print(a[i][j]+” “);

}

System.out.println();

}

}

}

class Matrix1

{

int I,j,k;

int b[][]=new int[3][3];

void b()

{

Scanner sc=new Scanner(System.in);

System.out.println(“b value:”);

for(i=0;i<3;i++){

for(j=0;j<3;j++){

b[i][j]=sc.nextInt();

}

}

System.out.println(“Resultant matrix of b:”);

for(i=0;i<3;i++){

for(j=0;j<3;j++){

System.out.print(b[i][j]+ “ “);

}

System.out.println();

}

}

}

class Multiply{

void mul(Matrix ob,Matrix1 ob1)

{

int I,j,k;

int a[][]=ob.a;

int b[][]=ob1.b;

int c[][]=new int[3][3];

for(i=0;i<3;i++){

for(j=0;j<3;j++){

c[i][j]=0;

for(k=0;k<3;k++){

c[i][j]=c[i][j]+(a[i][k]\*b[k][j]);

}

}

}

System.out.println(“Resultant matrix:”);

System.out.println(“value of c:”);

for(i=0;i<3;i++){

for(j=0;j<3;j++){

System.out.print(c[i][j]+” “);

}

System.out.println();

}

}

}

class result{

public static void main(String args[])

{

Matrix ob=new Matrix();

ob.a();

Matrix1 ob1=new Matrix1();

ob1.b();

Multiply o=new Multiply();

o.mul(ob,ob1);

}

}

Output:

Matrix multiplication

A value:

1

1

1

1

1

1

1

1

1

Resultant matrix:

1 1 1

1 1 1

1 1 1

B value:

2

2

2

2

2

2

2

2

2

Resultant matrix:

2 2 2

2 2 2

2 2 2

Resultant matrix:

Value of c:

6 6 6

6 6 6

6 6 6

10.Palindrome using Static

import java.util.\*;

class PalindromeExample{

static int sum=0;

static int r;

static void Pali(int n)

{

int temp=n;

while(n>0){

r=n%10;

sum=(sum\*10)+r;

n=n/10;

}

if(temp==sum)

System.out.println(“palindrome number “);

else

System.out.println(“not palindrome”);

}

public static void main(String args[]){

scanner sc=new Scanner(System.in);

System.out.println(“Enter number”);

int n=sc.nextInt();

Pali(n);

}

}

Output:

Enter number

121

Palindrome number

[Program finished]

Enter number

123

not palindrome

[Program finished]

11.Check whether the given number is neon or not using Super keyword

Import java.util.\*;

class base

{

int n;

base()

{

Scanner sc=new Scanner(System.in);

System.out.print(“Enter a number :”);

n=sc.nextInt();

}

}

class derived extends base

{

int s=0,a;

derived(){

super();

int sq=super.n\*super.n;

int n=sq;

while(n>0)

{

a=n%10;

s=s+a;

n=n/10;

}

if(s==super.n)

System.out.print(“neon”);

else

System.out.print(“not neon”);

}

}

class Demo

{

public static void main(String args[])

{

derived d=new derived();

}

}

Output:

Enter a number :9

Neon

[Program finished]

12.Print welcome 10 times using inner class inside loop

class Outer

{

String x=”welcome”;

void test()

{

for(int i=0;i<10;i++)

{

class Innerclass

{

void display ()

{

System.out.println(x);

}

}

Innerclass in= new Innerclass();

In.display();

}

}

}

class demo

{

public static void main(String args[])

{

Outer o=new Outer ();

o.test();

}

}

Output:

Welcome

Welcome

Welcome

Welcome

Welcome

Welcome

Welcome

Welcome

Welcome

Welcome

[Program finished]