一、

class Foo{

static void printFun(int test)

{

    if (test >0)  {

        System.out.printf("%d ",test);

        printFun(test-1);

          }

}

  public static void main(String[] args) {

       printFun(N);

}}

二、

class Foo{

static void printFun(int test)

{

    if (test >0)  {

        System.out.format("%d ",test);

        printFun(test-1);

  System.out.format("%d ",test);

          }

}

public static void main(String[] args) {

       printFun(N);

} }

三、

public static boolean isPrime(int n){

assert(n>=0);

int count=0;

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

if(n%i==0)

count++;

**return count==2?true:false;**

}

四、

public static boolean isParlome(String word){

if(word.length()==0||word.length()==1)

return true;

else if(word.charAt(0)!=word.charAt(word.length()-1))

return false;

else

return isParlome(word.substring(1,word.length()-1));

}

五、

public static void setValues(int[] arr)

{

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

//initialize the array range from 1 to 6

**arr[i]= (int)( Math.random()\*6+1);**

}

}

六

import java.io.\*;

public class ReadFileDemo {

// 异常可以捕获亦可以抛出

public static void main(String[] args) **throws IOException** {

try{

FileReader fr=new FileReader(new File(args[0]));

BufferedReader br=new BufferReader(fr);

String line;

// 注意算符的优先级

**while((line=br.readLine())!=null)**

System.out.println(line);

fr..close();

br.close();

}catch(IOException e){

e.printStackTrace();

}

}

}

七、

//类方法，关键字static

// 数，需要使用泛型， 另外数必须可以比较，实施Comparable接口

public static <T extends Comparable <T> > T myMax(T a ,T b){

return a.compareTo(b)>0? a:b;

}

八

class MyThread  **extends** Thread{

public void run(){

for(int i=0;i<60;i++){

try{

**Thread.sleep((int)(Math.random()\*1000+5000);**

}catch(InterruptedException e){

}

System.out.println(“Coding ….”);}

public static void main(String[] args){

**new MyThread().start();**

}}

九、

class MyThread **implements Runnable{**

public void run(){

**for(char c=’A’;i<=’Z’;c++){**

try{

**Thread.sleep(100);**

}catch(InterruptedException e){

}

System.out.format(“%c“，c）;

}

public static void main(String[] args){

Runnable r=newMyThread();

**new Thread(r).start();**

}}

十、

import java.util.Random;

public class MyDemo {

public static void main(String[] args){

int [] a={3,-4,0,2};

mySolution(a);

for(int x:a)

System.out.format("%4d",x);

}

**static void mySolution(int [] arr){**

**while(!isSorted(arr)){**

**myShuffle(arr);**

**}**

**}**

static void myShuffle(int[] arr) {

Random r = new Random();

for (int index = arr.length - 1; index > 0; index--)

mySwap(arr,index,r.nextInt(index));

}

static void mySwap(int[] arr,int i, int j){

int temp=arr[i];

arr[i]=arr[j];

arr[j]=temp;

}

// default ascending order;

static boolean isSorted(int[] arr){

for(int i=0;i<arr.length-1;i++) {

if (arr[i] > arr[i + 1])

return false;

}

return true;

} }

十一、

package edu.aku.band2022; //Good style

//abstract class MyShape { also works

public interface MyShape{

public static double PI=3.1415926;

public abstract double getArea();

public abstract double getPerimeter();

}

class MyPoint {

private int x,y;

public MyPoint(int x, int y){

this.x=x;

this.y=y;

}

public int getX(){

return x;

}

public int getY(){

return y;

}

}

//class MyCircle extends MyShape {

class MyCircle implements MyShape{

private MyPoint p;

private double radius;

public MyCircle(MyPoint p){

this.p=p;

}

public MyCircle(MyPoint p,double radius){

// this.p=p;

this(p);

this.radius=radius;

}

public void setRadius(double a){

radius=a;

}

public MyPoint getPoint(){

return p;

}

public double getArea(){

return PI\*radius\*radius;

}

@Override

public double getPerimeter() {

return PI\*2.0\*radius;

}

}

public class Test {   
 // project only hava one main method

public static void main(String[] args) {

MyPoint p=new MyPoint(2,2);

MyShape circle = new MyCircle(p,3);

//circle.setRadius(3);

System.out.format("%4.2f\n",circle.getArea());

System.out.format("%4.2f",circle.getPerimeter());

System.out.println(circle.toString());

}

}