import java.awt.\*;

import java.awt.event.\*;

public class AppInOut {

public static void main(String args[]) {

new FrameInOut();

}

}

class FrameInOut extends Frame implements ActionListener {

Label prompt;

FrameInOut() {

super("Hello World");

prompt = new Label("Welcome to Java World");

setLayout(new FlowLayout());// 界面上的图形对象的布局策略

add(prompt);

setSize(300, 200);

show();

}

public void actionPerformed(ActionEvent e) {

}

}

import java.awt.\*;

import java.applet.\*;

public class HelloJavaApp extends Applet

{

public void init(){

}

public void paint(Graphics g)

{

g.drawString("Welcome to Java applet World! ",25,25);

}

}

<applet code="helloworld\_app" width=150 height=100>

</applet>

package pkg;

import java.util.\*;

public class NumAdd {

public static void main(String[] args) {

int sum = 0;

int remainder;

System.out.print("Please input an integer: ");

Scanner in = new Scanner(System.in);

int num = in.nextInt();

in.close();

while (num > 0) {

remainder = num % 10;

sum += remainder;

num /= 10;

}

System.out.println("Sum of each digits is: " + sum);

}

}

package pkg;

import java.util.\*;

public class Fib {

public static int fib(int num) {

if (num == 1 || num == 2)

return 1;

else

return fib(num - 1) + fib(num - 2);

}

public static void main(String[] args) {

System.out.print("How many Fibonacci numbers do you want to print? ");

Scanner in = new Scanner(System.in);

int num = in.nextInt();

for (int i = 1; i <= num; i++) {

if ((i - 1) % 6 == 0)

System.out.println();

System.out.print(fib(i) + "\t");

}

in.close();

}

}

2:

package pkg;

import java.util.\*;

public class TestArray {

public static void main(String[] args) {

Scanner in = new Scanner(System.in);

System.out.print("How many elements in the array? ");

int n = in.nextInt();

System.out.print("The biggest element in the array? ");

int max = in.nextInt();

int[] array = new int[n];

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

int e = (int) (Math.random() \* max);

array[i] = e;

}

System.out.println("Before sorting: " + Arrays.toString(array));

Arrays.sort(array);

in.close();

System.out.println("After sorting: " + Arrays.toString(array));

}

}

package pkg;

import java.util.\*;

public class MyDate {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.println("请输入要设置的年月日");

System.out.print("年：");

int year = sc.nextInt();

System.out.print("月：");

int month = sc.nextInt();

System.out.print("日：");

int day = sc.nextInt();

Date dt = new Date();

dt.setDate(year, month, day);

dt.showDate();

dt.addDate();

}

}

class Date {

private int year;

private int month;

private int day;

public void setDate(int y, int m, int d) {

year = y;

month = m;

day = d;

}

void showDate() {

System.out.println("今天是" + year + "年" + month + "月" + day + "日");

}

void addDate() {

if ((month == 1 || month == 3 || month == 5 || month == 7 || month == 8 || month == 10) && day == 31) {

month += 1;

day = 1;

} else if ((month == 4 || month == 6 || month == 9 || month == 11) && day == 30) {

month += 1;

day = 1;

} else if ((month == 2) && (year % 4 == 0 && year % 100 != 0 || year % 100 == 0) && day == 29) {

month += 1;

day = 1;

} else if (month == 12 && day == 31) {

year += 1;

day = 1;

} else {

day++;

}

showDate();

}

}

package pkg;

import java.util.\*;

class ArraySort {

public int[] sim;

void setOrder() {

Arrays.sort(sim);

System.out.println(Arrays.toString(sim));

}

public ArraySort(int n) {

sim=new int[n];

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

int e = (int) (Math.random() \* 100);

sim[i] = e;

}

}

}

public class TestArray2 {

public static void main(String[] args) {

ArraySort array =new ArraySort(10);

array.setOrder();

}

}

package pkg;

public class Point {

private int x;

private int y;

public void setX(int x) {

this.x = x;

}

public void setY(int y) {

this.y = y;

}

public Point() { // constructor

this.x = 0;

this.y = 0;

}

public Point getPoint() {

return this;

}

public void movePoint(int x, int y) {

this.x = x;

this.y = y;

}

public String toString() {

return "[X=" + x + ", Y=" + y + "]";

}

public static void main(String[] args) {

Point point = new Point();

System.out.println(point);

point.movePoint(10, 20);

System.out.println(point);

}

}

package pkg;

public class Area {

public static void main(String[] args) {

Triangle1 tri = new Triangle1(5, 2);

Rectangle rec = new Rectangle(5, 2);

System.out.println("Triangle Area = " + tri.coutArea(tri.getBase(), tri.getHeight()));

System.out.println("Rectangle Area = " + rec.coutArea(rec.getBase(), rec.getHeight()));

}

}

class Triangle1 {

public int base;

public int height;

public float area;

public int getBase() {

return this.base;

}

public int getHeight() {

return this.height;

}

public Triangle1(int base, int height) {

this.base = base;

this.height = height;

}

public float coutArea(int base, int height) {

return (float) (base \* height \* 0.5);

}

}

class Rectangle {

public int base;

public int height;

public float area;

public int getBase() {

return this.base;

}

public int getHeight() {

return this.height;

}

public Rectangle(int base, int height) {

this.base = base;

this.height = height;

}

public float coutArea(int base, int height) {

return (float) (base \* height);

}

}

package pkg;

import java.util.\*;

public class IsPrime {

public static boolean isPrimeNorm(int num) {

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

if (num % i == 0) {

return false;

}

}

return true;

}

public static void main(String[] args) {

long startTime = System.currentTimeMillis();

Scanner in = new Scanner(System.in);

System.out.print("Please input n: ");

int n = in.nextInt();

in.close();

int[] array = new int[n + 1];

for (int i = 0; i < n + 1; i++)

array[i] = 1;

for (int i = 2; i < n + 1; i++) {

if (isPrimeNorm(i) == true)

array[i] = 0;

}

int count = 0;

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

if (array[i] == 0) {

System.out.print(i + "\t");

count++;

}

if (array[i] == 0 && count % 8 == 0)

System.out.println();

}

long endTime = System.currentTimeMillis();

System.out.println("\nTotal runtime: " + (endTime - startTime) + "ms");

}

}

3:

package pkg;

class Person {

public String name;

public char sex;

public int age;

public Person(String name, char sex, int age) {

this.name = name;

this.sex = sex;

this.age = age;

}

public void setData(String name, char sex, int age) {

this.name = name;

this.sex = sex;

this.age = age;

}

public Person getData() {

return this;

}

public String toString() {

return "Name: " + name + ", Sex: " + sex + ", Age: " + age;

}

}

class Student extends Person {

public Student() {

super("Name", 'M', 0); // !!!!

}

public int sID;

public int classNo;

public void setData(String name, char sex, int age, int sID, int classNo) {

setData(name, sex, age);

this.sID = sID;

this.classNo = classNo;

}

public Student getData() {

return this;

}

public String toString() {

return "Name: " + name + ", Sex: " + sex + ", Age: " + age + ", ID: " + sID + ", Class: " + classNo;

}

}

class Teacher extends Person {

public Teacher() {

super("Name", 'M', 0); // !!!

}

public int tID;

public String department = "NULL";

public void setData(String name, char sex, int age, int tID, String department) {

setData(name, sex, age);

this.tID = tID;

this.department = department;

}

}

public class Inheritance {

public static void main(String[] args) {

Person person = new Person("Tommy", 'M', 19);

System.out.println(person);

person.setData("New", 'F', 20);

System.out.println(person);

Student student = new Student();

System.out.println(student);

student.setData("Student", 'M', 15, 1001, 1618);

System.out.println(student);

}

}

package pkg;

public class NewPerson {

String nameString;

char sex;

int age;

abstract void setData(String name, char sex, int age);

abstract String getDetail();

}

class Student extends NewPerson {

int sID;

String speciality;

}

class Teacher extends NewPerson {

int tID;

String departmentString;

}

package pkg;

interface Print {

void print();

}

public class Interface implements Print {

public void print() {

System.out.println("Hello!");

}

public static void main(String[] args) {

Interface test = new Interface();

test.print();

}

}

package pkg;

interface Person1 {

public void setData(String name, char sex, String birthday);

public String getData();

}

public class Students implements Person1 {

private String name;

private char sex;

private String birthday;

private int sID;

private String speciality;

public void setData(String name, char sex, String birthday) {

this.name = name;

this.sex = sex;

this.birthday = birthday;

}

public String getData() {

return "Name: " + name + " Sex: " + sex + " Birthday: " + birthday + "\n" + "ID: " + sID + " Speciality: "

+ speciality;

}

public void setData(String name, char sex, String birthday, int sID, String speciality) {

this.name = name;

this.sex = sex;

this.birthday = birthday;

this.sID = sID;

this.speciality = speciality;

}

public static void main(String[] args) {

Students person = new Students();

person.setData("Tommy", 'M', "2000.06.27", 1001, "Music");

System.out.println(person.getData());

}

}

package pkg2;

public class PillarTest {

public static void main(String[] args) {

Pillar pillar1 = new Pillar(3, 3.5, 4.5, 3.3);

Pillar pillar2 = new Pillar(4, 3.5, 4.5, 3.3);

System.out.println("V-cuboid = " + pillar1.getV());

System.out.println("V-triangular prism = " + pillar2.getV());

}

}

class Pillar {

int n;

double a, b, h;

public Pillar(int n, double a, double b, double h) {

this.n = n;

this.a = a;

this.b = b;

this.h = h;

}

public double getV() {

if (n == 3) {

Triangle tri = new Triangle(n, a, b);

return h \* tri.getArea();

} else if (n == 4) {

Rectangle rec = new Rectangle(n, a, b);

return h \* rec.getArea();

}

return 0;

}

}

package pkg2;

public class GradeTest {

public static void main(String[] args) {

Grade grade = new Grade();

Student student = new Student(2018001, "张三", "男", "20000627", "软件学院", "软工");

Teacher teacher = new Teacher(1001, "李老师", "软件", "网安");

Course course = new Course(888001, "Java", 2, 48);

grade.stu = student;

grade.tea = teacher;

grade.cour = course;

grade.setNo(grade.stu.getNo());

grade.setStudentName(grade.stu.getName());

grade.setCourseName(grade.cour.getName());

grade.setTeacherName(grade.tea.getName());

grade.setPoint(100);

grade.setGNo(66660001);

System.out.print(grade.toString());

}

}

class Student {

private int sNo;

private String sName;

private String sSex;

private String sBirth; // YYYY.MM.DD

private String sSchool;

private String sMajor;

public Student() {

}

public Student(int sNo, String sName, String sSex, String sBirth, String sSchool, String sMajor) {

super();

this.sNo = sNo;

this.sName = sName;

this.sSex = sSex;

this.sBirth = sBirth;

this.sSchool = sSchool;

this.sMajor = sMajor;

}

public void setNo(int no) {

this.sNo = no;

}

public int getNo() {

return sNo;

}

public void setName(String name) {

this.sName = name;

}

public String getName() {

return sName;

}

public void setSex(String sex) {

this.sSex = sex;

}

public String getSex() {

return sSex;

}

public void setBirth(String birth) {

this.sBirth = birth;

}

public String getBirth() {

return sBirth;

}

public void setSchool(String school) {

this.sSchool = school;

}

public String getSchool() {

return sSchool;

}

public void setMajor(String major) {

this.sMajor = major;

}

public String getMajor() {

return sMajor;

}

public String toString() {

return sNo + "\t" + sName + "\t" + sSex + "\t" + sBirth + "\t" + sSchool + "\t" + sMajor;

}

}

class Teacher {

private int tNo;

private String tName;

private String tSchool;

private String tDepartment;

public Teacher() {

}

public Teacher(int tNo, String tName, String tSchool, String tDepartment) {

super();

this.tNo = tNo;

this.tName = tName;

this.tSchool = tSchool;

this.tDepartment = tDepartment;

}

public void setNo(int no) {

this.tNo = no;

}

public int getNo() {

return tNo;

}

public void setName(String name) {

this.tName = name;

}

public String getName() {

return tName;

}

public void setSchool(String school) {

this.tSchool = school;

}

public String getSchool() {

return tSchool;

}

public void setDepartment(String department) {

this.tDepartment = department;

}

public String getDepartment() {

return tDepartment;

}

public String toString() {

return tNo + "\t" + tName + "\t" + tSchool + "\t" + tDepartment;

}

}

class Course {

private int cNo;

private String cName;

private int cCredit;

private int cHour;

public Course() {

}

public Course(int cNo, String cName, int cCredit, int cHour) {

super();

this.cNo = cNo;

this.cName = cName;

this.cCredit = cCredit;

this.cHour = cHour;

}

public void setNo(int no) {

this.cNo = no;

}

public int getNo() {

return cNo;

}

public void setName(String name) {

this.cName = name;

}

public String getName() {

return cName;

}

public void setCredit(int credit) {

this.cCredit = credit;

}

public int getCredit() {

return cCredit;

}

public void setHour(int hour) {

this.cHour = hour;

}

public int getHour() {

return cHour;

}

public String toString() {

return cNo + "\t" + cName + "\t" + cCredit + "\t" + cHour;

}

}

class Grade {

public Student stu;

public Student getStu() {

return stu;

}

public void setStu(Student stu) {

this.stu = stu;

}

public Teacher tea;

public Teacher getTea() {

return tea;

}

public void setTea(Teacher tea) {

this.tea = tea;

}

public Course cour;

public Course getCour() {

return cour;

}

public void setCour(Course cour) {

this.cour = cour;

}

private int gNo;

private int sNo;

private String cName;

private String tName;

private String sName;

private int gPoint;

public Grade() {

}

public Grade(int gNo, int sNo, String sName, String cName, String tName, int gPoint) {

super();

this.gNo = gNo;

this.sNo = sNo;

this.cName = cName;

this.tName = tName;

this.sName = sName;

this.gPoint = gPoint;

}

public void setNo(int no) {

this.sNo = no;

}

public int getSNo() {

return sNo;

}

public void setGNo(int no) {

this.gNo = no;

}

public int getGNo() {

return gNo;

}

public void setCourseName(String name) {

this.cName = name;

}

public String getCourseName() {

return cName;

}

public void setTeacherName(String name) {

this.tName = name;

}

public String getTeacherName() {

return tName;

}

public void setStudentName(String name) {

this.sName = name;

}

public String getStudentName() {

return sName;

}

public void setPoint(int point) {

this.gPoint = point;

}

public int getPoint() {

return gPoint;

}

public String toString() {

return gNo + "\t" + sNo + "\t" + sName + "\t" + cName + "\t" + tName + "\t" + gPoint;

}

}

4:

package pkg;

public class yichang {

public static void main(String[] args) {

try {

// System.out.println(1 / 0);

// System.out.println(new int[] {}[0]);

// String str = null;

// System.out.println(str.toString());

}

catch (ArithmeticException e) {

System.out.println("算术异常");

}

catch (ArrayIndexOutOfBoundsException e) {

System.out.println("数组下标越界异常");

}

catch (NullPointerException e) {

System.out.println("空指针异常");

}

finally {

System.out.println("Finnaly");

}

}

}

package pkg2;

import java.util.Date;

import java.text.DateFormat;

import java.text.SimpleDateFormat;

import java.util.Locale;

public class Time {

public Time() {

FirstThread first = new FirstThread();

SecondThread second = new SecondThread();

ThirdThread third = new ThirdThread();

Thread thread1 = new Thread(first);

Thread thread2 = new Thread(second);

Thread thread3 = new Thread(third);

thread1.setPriority(Thread.MIN\_PRIORITY);

thread2.setPriority(Thread.MAX\_PRIORITY);

thread3.setPriority(Thread.MIN\_PRIORITY);

thread1.start();

thread2.start();

thread3.start();

}

public static void main(String[] args) {

new Time();

}

}

class FirstThread implements Runnable {

public void run() {

try {

for (int i = 0; i < 100; i++) {

SimpleDateFormat sdf = new SimpleDateFormat("北京时间yyyy年MM月dd日HH时mm分ss秒");// 格式化输出日期

Date now = new Date();

System.out.println(sdf.format(now));

Thread.sleep(1000);

Thread.yield();

}

} catch (InterruptedException e) {

}

}

}

class SecondThread implements Runnable {

public void run() {

try {

for (int i = 0; i < 100; i++) {

SimpleDateFormat sdf = new SimpleDateFormat("伦敦时间yyyy年MM月dd日HH时mm分ss秒");// 格式化输出日期

Date now = new Date();

long time = 60 \* 1000;// 60秒

Date beforeDate = new Date(now.getTime() - time \* 480);// 60秒前的时间

System.out.println(sdf.format(beforeDate));

Thread.sleep(1000);

Thread.yield();

}

} catch (InterruptedException e) {

}

}

}

class ThirdThread implements Runnable {

public void run() {

try {

for (int i = 0; i < 100; i++) {

SimpleDateFormat sdf = new SimpleDateFormat("纽约时间yyyy年MM月dd日HH时mm分ss秒");// 格式化输出日期

Date now = new Date();

long time = 60 \* 1000;// 60秒

Date beforeDate = new Date(now.getTime() - time \* 780);// 60秒前的时间

System.out.println(sdf.format(beforeDate));

Thread.sleep(1000);

Thread.yield();

}

} catch (InterruptedException e) {

}

}

}

package pkg2;

import java.io.BufferedWriter;

import java.io.File;

import java.io.FileWriter;

import java.io.IOException;

import java.util.Scanner;

public class IOTest {

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

File ofile=new File("src/Output.txt");

if (!ofile.isFile()) {

ofile.createNewFile();

}

BufferedWriter writer = new BufferedWriter(new FileWriter("src/Output.txt"));

Scanner scanner=new Scanner(System.in);

String input=scanner.nextLine();

writer.write(input);

writer.close();

scanner.close();

}

}

package pkg2;

import java.awt.Color;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import javax.swing.\*;

public class App extends JFrame implements ActionListener {

// 实现的点击按钮

JButton jb1, jb2, jb3;

public App() {

JFrame frm = new JFrame();

jb1 = new JButton("set red");

jb2 = new JButton("set green");

jb3 = new JButton("set blue");

JPanel jp = new JPanel();

jp.add(jb1);

jp.add(jb2);

jp.add(jb3);

this.add(jp);

this.setVisible(true);

this.setDefaultCloseOperation(3);

this.setSize(500, 500);

// 需要对按钮就行监听

jb1.addActionListener(this);// 这里需要指的是当前类

jb2.addActionListener(this);

jb3.addActionListener(this);

}

public void actionPerformed(ActionEvent e) {

if (e.getSource() == jb1) {

jb1.setForeground(Color.RED);

jb2.setForeground(Color.RED);

jb3.setForeground(Color.RED);

} else if (e.getSource() == jb2) {

jb1.setForeground(Color.GREEN);

jb2.setForeground(Color.GREEN);

jb3.setForeground(Color.GREEN);

} else if (e.getSource() == jb3) {

jb1.setForeground(Color.BLUE);

jb2.setForeground(Color.BLUE);

jb3.setForeground(Color.BLUE);

}

}

public static void main(String[] args) {

new App();

}

}

package pkg2;

import java.awt.Color;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.io.BufferedReader;

import java.io.BufferedWriter;

import java.io.FileReader;

import java.io.FileWriter;

import javax.swing.JFileChooser;

import javax.swing.JFrame;

import javax.swing.JMenu;

import javax.swing.JMenuBar;

import javax.swing.JMenuItem;

import javax.swing.JTextArea;

//让其继承窗口类

public class NotePal extends JFrame implements ActionListener {

private static final long serialVersionUID = 1L; //定义一个文本框

JTextArea jTextArea = null; //定义一个菜单栏

JMenuBar jMenuBar = null; //定义一个菜单

JMenu jMenu1 = null; //定义一个“打开”子选项

JMenuItem jMenuItem1 = null; //定义一个“保存”子选项

JMenuItem jMenuItem2 = null; //定义一个文件选择

JFileChooser jFileChooser = null; //定义一个FileReader文件输入流

FileReader fileReader = null; //定义一个FileWriter输出流

FileWriter fileWriter = null; //定义一个缓冲字符输入流

BufferedReader bufferedReader = null; //定义一个缓冲字符输出流

BufferedWriter bufferedWriter = null;

// @SuppressWarnings("unused")

public static void main(String[] args) {

NotePal notePal = new NotePal();

}

public NotePal()

{

jTextArea = new JTextArea();

jMenuBar = new JMenuBar();

jMenu1 = new JMenu("文件");

jMenuItem1 = new JMenuItem("打开");

jMenuItem1.addActionListener(this);

jMenuItem1.setActionCommand("打开");

jMenuItem2 = new JMenuItem("保存");

jMenuItem2.addActionListener(this);

jMenuItem2.setActionCommand("保存");

jTextArea.setBackground(Color.LIGHT\_GRAY);

//将组件添加上各自的位置

//将jMenuBar添加到JFrame中

this.setJMenuBar(jMenuBar);

//将jMenu添加到jMenuBar中

jMenuBar.add(jMenu1);

//将jMenuTItem1添加到Jmenu1中

jMenu1.add(jMenuItem1);

//将jMenuTItem2添加到Jmenu1中

jMenu1.add(jMenuItem2);

//将jTextArea添加到JFrame中

this.add(jTextArea);

//设置窗体的标题

this.setTitle("记事本");

//设置窗体的大小

this.setSize(600,400);

//当关闭窗口的时候，关闭进程

this.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

//显示窗口

this.setVisible(true);

}

@Override

public void actionPerformed(ActionEvent e) {

if(e.getActionCommand().equals("打开"))

{

//实例化一个JFileChoose

jFileChooser = new JFileChooser();

//设置文件选择窗口的名字

jFileChooser.setDialogTitle("选择文件...");

//设置文件窗口的默认路径

jFileChooser.showOpenDialog(null);

//显示文件窗口

jFileChooser.setVisible(true);

//用address保存用户编辑文件的绝对路径

String address = jFileChooser.getSelectedFile().getAbsolutePath();

try {

//为fileReader分配空间

fileReader = new FileReader(address);

//为bufferedReader分配空间

bufferedReader = new BufferedReader(fileReader);

//定义一个str判断输入的字符是否已经为空

String str = "";

//定义一个strAll接收文件的全部信息

String strAll = "";

//读取文件信息，并将其保存到strAll中

while((str = bufferedReader.readLine()) != null)

{

strAll += str + "\r\n";

}

//将strAll中的全部信息显示到JTextArea上

jTextArea.setText(strAll);

} catch (Exception e2) {

e2.printStackTrace();

}finally{

try {

//关闭文件

bufferedReader.close();

fileReader.close();

} catch (Exception e3) {

e3.printStackTrace();

}

}

}

//如果用户点的是保存按钮

if(e.getActionCommand().equals("保存"))

{

//创建一个保存窗口

JFileChooser jFileChooser1 = new JFileChooser();

//设置窗口名字

jFileChooser1.setDialogTitle("另存为...");

//设置默认设置

jFileChooser1.showSaveDialog(null);

//显示窗口

jFileChooser1.setVisible(true);

try {

//为fileWrite分配空间

fileWriter = new FileWriter(jFileChooser1.getSelectedFile().getAbsolutePath());

//为bufferedWrite分配空间

bufferedWriter = new BufferedWriter(fileWriter);

//保存进去

bufferedWriter.write(this.jTextArea.getText());

} catch (Exception e2) {

e2.printStackTrace();

}finally{

//关闭文件

try {

bufferedWriter.close();

fileWriter.close();

} catch (Exception e3) {

e3.printStackTrace();

}

}

}

}

}