1. 编写一个Date类：

1）有nextDay方法，能够实现增加一天的功能（考虑如果是某月最后一天，加一天后应该到下个月的第一天；如果是某年的最后一天，加一天后到下一年的第一天）；

2）要对日期进行验证；

**public** **class** Date

{

**private** **int** month; //1-12

**private** **int** day; //1-31 按月变化

**private** **int** year; //不限制

//在构造器中直接限制 month、year的值的范围（也可以改为使用函数来判断，类似checkDay那样；课后可以自行修改）

//调用checkDay方法来判断day是否正确

**public** Date(**int** theYear, **int** theMonth, **int** theDay) {

**if** (theMonth > 0 && theMonth <= 12) {

month = theMonth;

}

**else** {

month = 1;

System.***out***.println("Month " + theMonth + " invalid. Set to month 1.");

}

**if** (theYear > 0) {

year = theYear;

}

**else** {

year = 1;

System.***out***.println("Year " + theYear + " invalid. Set to year 1.");

}

day = checkDay(theDay);

System.***out***.println("Date object constructor for date " + toDateString());

}

//验证day是否正确，要根据year和month变化

**public** **int** checkDay(**int** testDay) {

**int** daysPerMonth[] = {0, 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31 };

//检查润年

**if** (month == 2 && testDay == 29 && (year % 400 == 0 || (year % 4 == 0 && year % 100 != 0))) {

**return** testDay;

}

//检查日期是否符合月份

**if** (testDay > 0 && testDay <= daysPerMonth[month]) {

**return** testDay;

}

System.***out***.println("Day " + testDay + " invalid. Set to day 1.");

**return** 1;

}

//日期增加一天

**public** **void** nextDay() {

**int** testDay = day + 1;

**if** (checkDay(testDay) == testDay) {

day = testDay;

}

**else** {

day = 1;

nextMonth();

}

}

//月份增加一月

**public** **void** nextMonth() {

**if** (12 == month) {

year++;

}

month = month % 12 + 1;

}

//返回月份

**public** **int** getMonth() {

**return** month;

}

//返回日期字符串，以 year-month-day 格式

**public** String toDateString() {

**return** year + "-" + month + "-" + day;

}

}

编写一个DateTest类：

1）利用循环测试Date的nextDay方法；设置一个初始日期（如2009年1月1日，或者2009年12月1日），在循环中（至少40次，这样可以看出日期增加后的月份、年份等的变化），打印每次循环迭代的结果，以证明nextDay方法正确。

**public** **class** DateTest

{

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

System.***out***.println("测试日期增加：");

Date testDate = **new** Date(2016, 11, 2);

//测试日期增加，看日期、月份、年份变化是否正确；

**for**(**int** counter = 0; counter < 40; counter ++){

testDate.nextDay();

System.***out***.println("日期增加一天："+testDate.toDateString());

}

}

}

2. 输入出租车公里数，输出应付的车费数：

收费标准如下：3000米以下为8元，每超过500米增加1.2元，不足500米按500米计算。

**public** **class** TaxiCharge

{

**public** **static** **double** getFee(**long** meter){

**double** fee = 0;

**if**(meter <= 3000)

{

fee = 8;

}

**else**

{

//先计算出整数部分

**int** n = (**int**)(meter - 3000)/500;

//再判断是否有余数

**int** m = (**int**)(meter - 3000)%500;

**if**(m > 0)

{

fee = 8 + 1.2\*(n+1);

}

**else**

{

fee = 8 + 1.2\*n;

}

}

**return** fee;

}

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

System.***out***.println(*getFee*(1000));//不足3000；

System.***out***.println(*getFee*(3500));//超过3000，无余数；

System.***out***.println(*getFee*(4300));//超过3000，余数不足500；

}

}

3. 古典问题：有一对兔子，从出生后第3个月起每个月都生一对兔子，小兔子长到第三个月后每个月又生一对兔子，假如兔子都不死，问每个月的兔子对数为多少？ 程序分析： 兔子的规律为数列1,1,2,3,5,8,13,21....

**public** **class** Rabbit

{

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

System.***out***.println("1-10个月的兔子对数分别为：");

**for**(**int** n = 1;n < 11;n++)

{

System.***out***.println("第"+n+"个月兔子对数为"+*fun*(n));

}

}

**private** **static** **int** fun(**int** n)

{

**if**(n==1 || n==2)

**return** 1; //当if分支中只有一个语句的时候，可以省略{}，但是不推荐。

**else**

**return** *fun*(n-1)+*fun*(n-2);

}

}

4. 判断1-1000之间有多少个素数，并输出所有素数。

程序分析：判断素数的方法：用一个数分别去除2到sqrt(这个数)，如果能被整除，则表明此数不是素数，反之是素数。

**public** **class** PrimeNumber

{

**public** **static** **void** main(String[] args)

{

**int** m = 1;

**int** n = 1000;

**int** count = 0; //统计素数个数

**for**(**int** i=m;i<n;i++)

{

**if**(*isPrime*(i))

{

count++;

System.***out***.print(i + " ");

**if** (count % 10 == 0)

{

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

}

}

}

System.***out***.println();System.***out***.println("在"+m+"和"+n+"之间共有"+count+"个素数");

}

//判断素数

**private** **static** **boolean** isPrime(**int** n)

{

**boolean** flag = **true**;

**if**(n==1)

flag = **false**;

**else**

{

**for**(**int** i=2;i<=Math.*sqrt*(n);i++)

{

**if**((n%i)==0 || n==1)

{

flag = **false**;

**break**;

}

**else**

{

flag = **true**;

}

}

}

**return** flag;

}

}

5. 定义一个20\*5的二维数组，用来存储某班级20位学员的5门课的成绩；这五门课按存储顺序依次为：core C++，coreJava，Servlet，JSP和EJB。

（1）循环给二维数组的每一个元素赋0~100之间的随机整数。

（2）按照列表的方式输出这些学员的每门课程的成绩。

（3）要求编写程序求每个学员的总分，将其保留在另外一个一维数组中。

（4）要求编写程序求所有学员的某门课程的平均分。

附：获取0~100之间的随机整数的代码：(int)Math.round(Math.random() \* 100)

**public** **class** Scorecalculate

{

**int**[][] score = **new** **int**[20][5];

**int**[] scoreTotal = **new** **int**[20];

**public** Scorecalculate() {

**int** scoreTmp = 0;

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

**for** (**int** j = 0; j < score[i].length; j++) {

scoreTmp = (**int**) Math.*round*(Math.*random*() \* 100);

score[i][j] = scoreTmp;

}

}

}

**public** **void** printScore() {

System.***out***.println("下面是所有学员的成绩：");

System.***out***.println(" \t core C++ \t coreJava \t Servlet \t JSP \t\t EJB");

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

System.***out***.print("第" + (i + 1) + "位：\t");

**for** (**int** j = 0; j < score[i].length; j++) {

System.***out***.print(score[i][j] + "\t\t");

}

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

}

System.***out***.println("------------------------");

}

**public** **void** getScoreTotal() {

**int** scoreTmp = 0;

System.***out***.println("下面是所有学员的总成绩：");

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

scoreTmp = 0;

**for** (**int** j = 0; j < score[i].length; j++) {

scoreTmp += score[i][j];

}

scoreTotal[i] = scoreTmp;

System.***out***.println("第" + (i + 1) + "位：\t" + scoreTotal[i]);

}

System.***out***.println("------------------------");

}

/\*\*

\* **@param** course：在 0 到4 之间，包括0和4；

\* **@return**

\*/

**public** **double** getAverage(**int** course) {

**double** scoreAverage = 0;

**int** i = 1;

**if** (course >= 0 && course < 5) {

**for** (i--; i < score.length; i++) {

scoreAverage += score[i][course];

}

}

// System.out.println(i);

**return** scoreAverage / i;

}

/\*\*

\* **@param** args

\*/

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

// **TODO** Auto-generated method stub

Scorecalculate classScore = **new** Scorecalculate();

//打印所有人的成绩：

classScore.printScore();

//获取所有人的总成绩，并打印：

classScore.getScoreTotal();

//获取所有学员的某门课程的平均分：

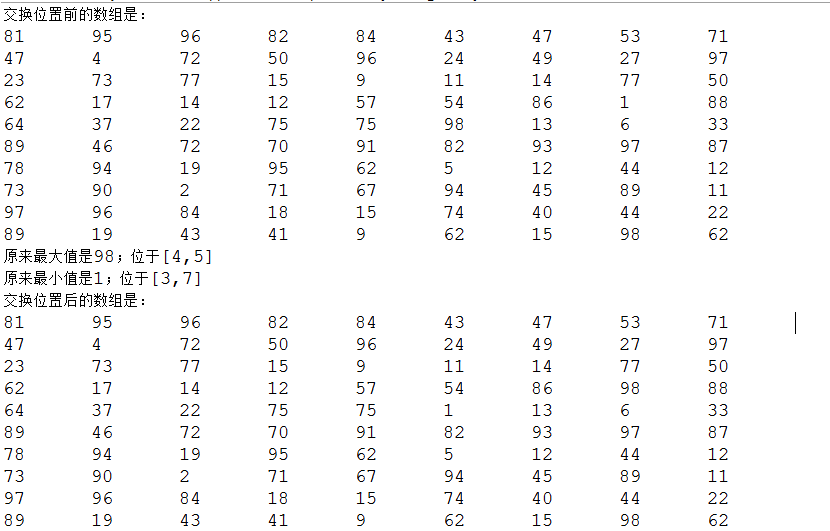
System.***out***.println(classScore.getAverage(0));

}

}

6. 用随机数对象产生0---100范围内的不同整数，用以初始化一个M\*N的二维数组(M\*N<100)，然后编程找出二维数组中的最大数和最小数，并将其交换位置。要求输出原始二维数组及交换后的二维数组。

参考输出：



**public** **class** ArrayExercise

{

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

**int**[][] array = **new** **int**[10][9];

System.***out***.println("交换位置前的数组是：");

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

**for**(**int** j=0; j<array[i].length; j++){

array[i][j] = (**int**)Math.*round*(Math.*random*()\*100);

System.***out***.print(array[i][j] + "\t");

}

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

}

**int** max = array[0][0];

**int** max\_x=0,max\_y=0; // 记录最大值所在位置

**int** min = array[0][0];

**int** min\_x=0,min\_y=0; //记录最小值所在位置

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

**for**(**int** j=0; j<array[i].length; j++){

**if**(max < array[i][j]){

max = array[i][j];

max\_x = i;

max\_y = j;

}

**if**(min > array[i][j]){

min = array[i][j];

min\_x = i;

min\_y = j;

}

}

}

System.***out***.println("原来最大值是"+max+"；位于["+(max\_x)+","+(max\_y)+"]");

System.***out***.println("原来最小值是"+min+"；位于["+(min\_x)+","+(min\_y)+"]");

//交换位置

array[min\_x][min\_y] = max;

array[max\_x][max\_y] = min;

System.***out***.println("交换位置后的数组是：");

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

**for**(**int** j=0; j<array[i].length; j++){

System.***out***.print(array[i][j] + "\t");

}

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

}

}

}