**日期：2020/5/22**

**常用类**

1. **实验目的**

1、熟悉Java中String、StringBuffer、Math、Scanner、Data等类的使用方法。

2、使用常用类解决一般性的应用问题。

1. **实验内容**

1、编程要求：

请仔细阅读右侧代码，根据方法内的提示，在Begin - End区域内进行代码补充，具体任务如下：

Demo类和Person类已经写好，在测试类中创建Demo类对象d1，传入输入值num1， d1调用toString方法并打印输出该值；

创建Demo类对象d2，同样传入输入值num1，打印判断d1和d2是否相等（实际是比较地址）；

创建Person类对象p，传入输入值num2，打印判断d1和p是否相等（实际是比较地址）；

测试说明：

补充完代码后，点击测评，平台会对你编写的代码进行测试，当你的结果与预期输出一致时，即为通过。

测试输入：

100

100

预期输出：

Demo:100

true

false

2、编程要求：

请仔细阅读右侧代码，根据方法内的提示，在Begin - End区域内进行代码补充，具体任务如下：

编写测试类，输入int型值aa传递给Integer类对象a，String型值bb传递给Integer类对象b，输入int型值c，aa、bb、c对应的数值相等；

输入String型值str11传递给String类对象str1，输入String型值str22传递给String类对象str2，str1和str2对应的内容相同；

打印输出a == b、a == c、b == c、a.equals(b)；

打印输出str1 == str2、str1.equals(str2)。

测试说明：

补充完代码后，点击测评，平台会对你编写的代码进行测试，当你的结果与预期输出一致时，即为通过。

测试输入：

100

100

100

educoder

educoder

预期输出：

false

true

true

true

false

true

3、编程要求：

请仔细阅读右侧代码，根据方法内的提示，在Begin - End区域内进行代码补充，具体任务如下：

编写测试类，输入一个标准的网址，输出其域名的主体；

将该主体转换为大写，打印输出；

具体输出要求请看测试说明。

测试说明：

补充完代码后，点击测评，平台会对你编写的代码进行测试，当你的结果与预期输出一致时，即为通过。

测试输入：

https://www.educoder.net/

预期输出：

educoder

EDUCODER

4、编程要求：

请仔细阅读右侧代码，根据方法内的提示，在Begin - End区域内进行代码补充，具体任务如下：

设计一个密码的自动生成器：密码由大写字母/小写字母/数字组成，生成六位随机密码。

分别以1、2、3作为种子数创建Random对象，生成六位随机密码进行测试。

具体输出要求请看测试说明。

测试说明：

补充完代码后，点击测评，平台会对你编写的代码进行测试，当你的结果与预期输出一致时，即为通过。

测试输入：

1

预期输出：

NAvZuG

5、编程要求：

请仔细阅读右侧代码，根据方法内的提示，在Begin - End区域内进行代码补充，具体任务如下：

编写测试类，算一下你的出生日期到2020年10月1日总共有多少天；

具体输出要求请看测试说明。

测试说明：

补充完代码后，点击测评，平台会对你编写的代码进行测试，当你的结果与预期输出一致时，即为通过。

测试输入：

1999-09-01

预期输出：

你的出生日期距离2020年10月1日:7701天

6、编程要求：

请仔细阅读右侧代码，根据方法内的提示，在Begin - End区域内进行代码补充，具体任务如下：

编写测试类MathTest，输入4个int类型数值，7个double类型数值；

分别输出第一个数的平方根，第二个数的立方根，第三个数的第四个数次方，第五个数和第六个数的最大值、最小值，第七个数的绝对值，第八个数的ceil值，第九个数的floor值，第十个数的rint值，第十一个数的round值。

测试说明：

补充完代码后，点击测评，平台会对你编写的代码进行测试，当你的结果与预期输出一致时，即为通过。

测试输入：

25

27

4

3

3.5

5.5

-8.8

10.8

9.6

10.62

11.51

预期输出：

5.0

3.0

64.0

5.5

3.5

8.8

11.0

9.0

11.0

12

1. **实验结果**

1、

import java.util.Scanner;

public class ObjectTest {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        int num1 = sc.nextInt();

        int num2 = sc.nextInt();

        // 在测试类中创建Demo类对象d1，传入输入值num1， d1调用toString方法并打印输出该值

        // 创建Demo类对象d2，同样传入输入值num1，打印判断d1和d2是否相等（实际是比较地址）

        /\*\*\*\*\*\*\*\*\* Begin \*\*\*\*\*\*\*\*\*/

Demo d1=new Demo(num1);

        System.out.println(d1.toString());

        Demo d2=new Demo(num1);

        if (d1.equals(d2)) {

            System.out.println("true");

        } else {

            System.out.println("false");

        }

        /\*\*\*\*\*\*\*\*\* End \*\*\*\*\*\*\*\*\*/

        // 创建Person类对象p，传入输入值num2，打印判断d1和p是否相等（实际是比较地址）

        /\*\*\*\*\*\*\*\*\* Begin \*\*\*\*\*\*\*\*\*/

Person p=new Person(num2);

        if (d1.equals(p)) {

            System.out.println("true");

        } else {

            System.out.println("false");

        }

        /\*\*\*\*\*\*\*\*\* End \*\*\*\*\*\*\*\*\*/

    }

}

class Demo {

    private int num;

    public Demo(int num) {

        this.num = num;

    }

    public boolean equals(Object obj) // Object obj = new Demo()

    {

        if (!(obj instanceof Demo)) // 判断obj是否和Demo是同类

            return false;

        Demo d = (Demo) obj; // 将父类的引用(Object)向下转换为子类(Demo)

        return this.num == d.num;

    }

    public String toString() {

        return "Demo:" + num; // 返回对象的值（每一个对象都有自己的特定的字符串）

    }

}

class Person {

    private int num;

    public Person(int num) {

        this.num = num;

    }

}

2、

import java.util.Scanner;

public class WrapperTest {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        int aa = sc.nextInt();

        String bb = sc.next();

        int c = sc.nextInt();

        String str11 = sc.next();

        String str22 = sc.next();

        // 包装类中“==”与equals的用法比较

        // 值得注意的是，包装类中的equals方法和String类一样，都是重写了Object类中的equals方法，因此比较的是内容而不是地址，

        // 而“==”比较的依然是引用变量的地址，只是当包装类型和与之相对应的基本类型进行“==”比较时会先做自动拆箱处理。

        /\*\*\*\*\*\*\*\*\* Begin \*\*\*\*\*\*\*\*\*/

        Integer a=new Integer(aa);

        Integer b=Integer.parseInt(bb);

        String str1=new String(str11);

        String str2=new String(str22);

        System.out.println(a==b);

        System.out.println(a==c);

        System.out.println(b==c);

        System.out.println(a.equals(b));

        System.out.println(str1==str2);

        System.out.println(str1.equals(str2));

        /\*\*\*\*\*\*\*\*\* End \*\*\*\*\*\*\*\*\*/

    }

}

3、

import java.util.Scanner;

public class StringTest {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        String str = sc.next();

        // String substring(int start,int end)

        // 截取字符串，传入的两个参数分别为要截取边界的下标

        // 在java api 中,通常使用两个数字表示范围时,都是含头不含尾,即包含起始下标对应的内容,但不包含结束下标的处对应的内容

        // String toUpperCase() 将当前字符串中的英文部分转换为全大写

        /\*\*\*\*\*\*\*\*\* Begin \*\*\*\*\*\*\*\*\*/

        String str1=str.substring(12,str.lastIndexOf('.'));

        if(str1.indexOf('.')>0){

            str1=str1.substring(0,str1.indexOf('.'));

        }

        System.out.println(str1);

        str1=str1.toUpperCase();

        System.out.println(str1);

        /\*\*\*\*\*\*\*\*\* End \*\*\*\*\*\*\*\*\*/

    }

}

4、

//密码的自动生成器：密码由大写字母/小写字母/数字组成，生成六位随机密码

import java.util.Random;

import java.util.Scanner;

public class RandomTest {

    public static void main(String[] args) {

        // 定义一个字符型数组

        char[] pardStore = new char[62];

        // 把所有的大写字母放进去 把所有的小写字母放进去 把0到9放进去

        /\*\*\*\*\*\*\*\*\* Begin \*\*\*\*\*\*\*\*\*/

        for(int i=0;i<26;i++)

        {

            pardStore[i]=(char)('A'+i);

            pardStore[26+i]=(char)('a'+i);

        }

        for(int i=0;i<10;i++)

        {

            pardStore[52+i]= (char)('0' + i);

        }

        /\*\*\*\*\*\*\*\*\* End \*\*\*\*\*\*\*\*\*/

        // 分别以1、2、3作为种子数 生成6位随机密码

        Scanner sc = new Scanner(System.in);

        int seed = sc.nextInt();

        /\*\*\*\*\*\*\*\*\* Begin \*\*\*\*\*\*\*\*\*/

        Random r=new Random(seed);

        String str="";

        int[] arr=r.ints(6,0,62).toArray();

        for(int i=0;i<6;i++)

        {

            str+=pardStore[arr[i]];

        }

        System.out.print(str);

        /\*\*\*\*\*\*\*\*\* End \*\*\*\*\*\*\*\*\*/

    }

}

5、

import java.text.ParseException;

import java.text.SimpleDateFormat;

import java.util.Date;

import java.util.Scanner;

public class DateTest {

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

        // 键盘录入你的出生年月日 格式为yyyy-MM-dd

        // 把该字符串转换为一个日期

        // 通过该日期得到一个毫秒值

        // 获取2020年10月1日的毫秒值

        // 两者想减得到一个毫秒值

        // 把该毫秒值转换为天 打印输出

        Scanner sc = new Scanner(System.in);

        String line = sc.nextLine();

        /\*\*\*\*\*\*\*\*\* Begin \*\*\*\*\*\*\*\*\*/

      SimpleDateFormat sdf=new SimpleDateFormat("yy-MM-dd");

      Date d1=sdf.parse(line);

      Date d2=sdf.parse("2020-10-01");

      long diff=d2.getTime()-d1.getTime();

      diff=diff/86400000;

      System.out.println("你的出生日期距离2020年10月1日："+diff+"天");

        /\*\*\*\*\*\*\*\*\* End \*\*\*\*\*\*\*\*\*/

    }

}

6、

import java.util.Scanner;

import java.lang.Math;

public class MathTest {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        int a1 = sc.nextInt();

        int a2 = sc.nextInt();

        int a3 = sc.nextInt();

        int a4 = sc.nextInt();

        double a5 = sc.nextDouble();

        double a6 = sc.nextDouble();

        double a7 = sc.nextDouble();

        double a8 = sc.nextDouble();

        double a9 = sc.nextDouble();

        double a10 = sc.nextDouble();

        double a11 = sc.nextDouble();

        /\*\*\*\*\*\*\*\*\* Begin \*\*\*\*\*\*\*\*\*/

        System.out.println(Math.sqrt(a1));

        System.out.println(Math.cbrt(a2));

        System.out.println(Math.pow(a3,a4));

        System.out.println(Math.max(a5,a6));

        System.out.println(Math.min(a5,a6));

        System.out.println(Math.abs(a7));

        System.out.println(Math.ceil(a8));

        System.out.println(Math.floor(a9));

        System.out.println(Math.rint(a10));

        System.out.println(Math.round(a11));

        /\*\*\*\*\*\*\*\*\* End \*\*\*\*\*\*\*\*\*/

    }

}

1. **实验总结**

通过实践使我更加了解了Java常用类。以后需要通过经常练习java语言掌握Java类。