# 

**院系：信息科学与技术学院**

**系别：计算机**

**班号：1**

**实验者姓名：柳煜颖**

**学号：22920132203826**

**实验日期：** 04.20

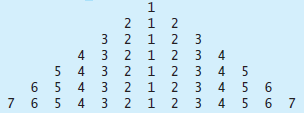
**实验报告完成日期：04.27**

**一 .三角形类**

题目

从课本的**GeometricObject**类继承一个**Triangle** 类，并满足下列条件：

* 1. 有3个**私有double**成员表示三条边，默认长度为1.0；
  2. 有1个无参构造方法，用于创建默认三角形；
  3. 有1个3参数构造方法，用于创建指定三角形，不需要考虑3边非法的情况；
  4. 有1个getArea()，1个 getPerimeter() 分别返回三角形的面积和周长；
  5. 有1个toString()方法返回三条边的信息，形如 "Triangle: side1 = " + side1 + " side2 = " + side2 + " side3 = " + side3;
  6. 写一个测试类，测试Triangle类，并且需要设置该类继承来的**color**和**filled** 属性。



3.实验心得

本实验的关键类的继承，程序写完后怎么调试也不对，后来发现是{的地方不对，以及**class** Triangle **extends** GeometricObject前多了**public.**

4.实验代码

**public** **class** test {

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

Triangle t = **new** Triangle(3,3,3);

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

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

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

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

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

}

}

**class** Triangle **extends** GeometricObject {

**private** **double** s1=1.0;

**private** **double** s2=1.0;

**private** **double** s3=1.0;

**public** Triangle(){

}

**public** Triangle(**double** s1,**double** s2,**double** s3){

**this**.s1 = s1; **this**.s2 = s2; **this**.s3 = s3;

}

**public** **double** getArea(){

**double** p = **this**.getPerimeter() / 2;

**double** a = p \* (p - **this**.s1) \* (p - **this**.s2) \* (p - **this**.s3);

**double** s = Math.*sqrt*(a);

**return** s;

}

**public** **double** getPerimeter(){

**return** **this**.s1+**this**.s2+**this**.s3;

}

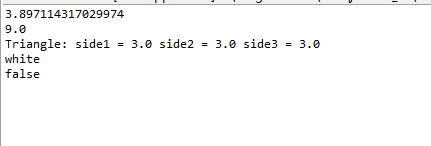
**public** String toString(){

**return** ("Triangle: side1 = " + s1 + " side2 = " + s2 + " side3 = " + s3);

}

}

5.实验结果



1. **删除重复数字**
2. 实验题目

* 写一个方法，能够删除ArrayList中的重复数字，其中数字的类型是Integer。要求**在原ArrayList中直接删除**重复数字。方法原型如下：

**public static void** removeDuplicate(ArrayList<Integer> list)

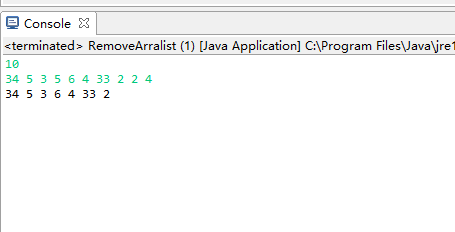
* 在main方法中调用上述方法，输出删除后的结果。例如某次运行结果如下：
  + 输入：34 5 3 5 6 4 33 2 2 4
  + 输出：34 5 3 6 4 33 2

1. 实验心得

课件里面关于ArrayList的用法不详细，要自己通过其它的学习掌握ArrayList的用法，例如Add、AddRange、Remove、RemoveAt、RemoveRange、Insert、InsertRange

1. 实验代码
2. **import** java.util.ArrayList;
3. **import** java.util.Scanner;
4. **public** **class** RemoveArralist{
5. **public** **static** **void** main(String[] args) {
6. ArrayList<Integer>a=**new** ArrayList<Integer>();
7. Scanner input=**new** Scanner(System.***in***);
8. **int** n=input.nextInt();
9. **for**(**int** i=0;i<n;i++){
10. a.add(input.nextInt());
11. }
12. *removeDuplicate*(a);
13. **for**(**int** i=0;i<a.size();i++)
14. System.***out***.print(a.get(i)+" ");
15. }
16. **public** **static** **void** removeDuplicate(ArrayList<Integer>a){
17. **for**(**int** i=1;i<a.size();i++){
18. **for**(**int** j=0;j<i;j++){
19. **if**(a.get(i)==a.get(j)){
20. a.remove(i);
21. i--;
22. }
23. }
24. }
25. }
26. }

4.实验结果



1. 异常捕捉

1.实验题目

* 1. 首先创建一个100个元素的数组，并用随机数填充该数组，类型随意。
  2. 提示用户输入一个下标，输出该下标对应的元素值。如果输入不是整数，输出“**Input Mismatch.**”并提示用户重新输入；如果下标越界，输出“**Out of Bounds.**”并提示用户重新输入；如果不是以上情况，程序输出元素值后正常结束。
  3. 代码不允许使用if语句，只能用异常捕捉实现上述功能。

2.实验心得

参考课件上的例子，开始一直不能run，后来发现是开头少了**import** java.util.InputMismatchException;

3.实验代码

**import** java.util.InputMismatchException;

**import** java.util.Scanner;

**public** **class** CatchError {

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

**int**[] myList = **new** **int**[100];

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

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

myList[i] =(**int**)( Math.*random*() \* 100);

}

**boolean** continueInput=**true**;

**do**{

**try**{

System.***out***.print("Enter a number:");

**int** number=input.nextInt();

System.***out***.println(myList[number]);

continueInput = **false**;

}

**catch**(InputMismatchException ex){

System.***out***.println("Input Mismatch.\n");

input.nextLine();

}

**catch**(IndexOutOfBoundsException ex){

System.***out***.println("Out of Bounds.\n");

input.nextLine();

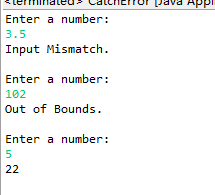
}

}**while**(continueInput);

}

}

4.实验结果



1. **网页分析**
2. 实验题目

* 抓取厦大主页中的“快速链接”下的所有内容。
* 提示，“快速链接”部分的HTML形如：

<li><a href="http://e.weibo.com/xmunews" target="\_blank"><i class="icon icon-list-alt"></i>官方微博</a></li>

* 程序输出为：[序号] 标题 URL ，每行一个链接，如：  
  [1]官方微博 http://e.weibo.com/xmunews  
  [2]易班厦大 http://yiban.xmu.edu.cn/  
  [3]厦大网址导航 http://123.xmu.edu.cn/  
  ……

1. 实验心得

这个题是通过爬虫抓取网页链接，参考课件上的例题，还是好难TwT, 并没有很懂，就是根据例题改改

1. 实验代码

**import** java.util.Scanner;

**import** java.util.ArrayList;

**import** java.net.URL;

**public** **class** ReadFileFromURL {

**static** ArrayList<String> *listtitle*=**new** ArrayList<>();

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

java.util.Scanner input=**new** Scanner(System.***in***);

String url="http://www.xmu.edu.cn/";

*crawler*(url);

ArrayList<String> listurl=**new** ArrayList<>();

listurl=*getSubURL*(url);

**for**(**int** i=0;i<listurl.size();i++){

**int** j=i+1;

System.***out***.println("["+j+"]"+*listtitle*.get(i)+" "+listurl.get(i));

}

}

**public** **static** **void** crawler(String staringURL){

ArrayList<String> listOfPENDINGurlS=**new** ArrayList<>();

ArrayList<String> listOfTraversedURLsl=**new** ArrayList<>();

listOfPENDINGurlS.add(staringURL);

**int** count=1;

**while**(listOfPENDINGurlS.isEmpty()&&listOfTraversedURLsl.size()<=100){

String urlstring=listOfPENDINGurlS.remove(0);

**if**(!listOfTraversedURLsl.contains(urlstring)){

listOfTraversedURLsl.add(urlstring);

System.***out***.println("craw"+urlstring);

**for**(String s:*getSubURL*(urlstring)){

**if**(!listOfTraversedURLsl.contains(s)){

listOfPENDINGurlS.add(s);

}

}

}

}

}

**private** **static** ArrayList<String> getSubURL(String urlstring) {

ArrayList<String> list=**new** ArrayList<>();

**try**{

URL url=**new** URL(urlstring);

Scanner input = **new** Scanner(url.openStream(),"UTF-8");

**int** current=0,current1=0;

**while**(input.hasNext()){

String line=input.nextLine();

current=line.indexOf("http:",current);

**while**(current>0){

**int** endindex=line.indexOf("target=\"\_blank\"><i class=\"icon icon-list-alt\">",current)-2;

**if**(endindex>0){

current1=line.indexOf("</i>")+4;

**int** endindex1=line.indexOf("</a></li>",current1);

list.add(line.substring(current,endindex));

*listtitle*.add(line.substring(current1,endindex1));

current=line.indexOf("blank\">",endindex);

}

**else**

current=-1;

}

}

}

**catch**(Exception ex){

System.***out***.println("Error:"+ex.getMessage());

}

**return** list;

}

}

}

1. 实验结果

