**实验四 类与对象**

学生姓名： 胡权富 学号： 8002119097 专业班级： 计算机193

实验类型：□ 验证 □ 综合 □ 设计 □ 创新 实验日期： 实验成绩：

**一、实验目的**

本实验的目的用于指导读者掌握Java语言面向对象的程序设计方法，理解对象的封装等概念，要求读者在调试实例程序后，总结出面向对象的事务定义以及以对象的形式进行封装等内容。

**二、实验内容**

1、编写一个名为“复数”的类，包含复数的实部和虚部(数据成员），以及复数之间的基本算术运算：加、减（方法成员），并要求复数加减运算，程序运行中能对给定的复数进行运算，并打印运算的结果。

2、编写一个小游戏实现两人（Titan和Zues）格斗。

基本要求（Version1.0）：

（1）     创建一个Titan类,有int型数据成员Energy；无参构造方法，可将Energy初始化为800；带参构造方法，可将Energy初始化为参数指定值；数据成员Energy的设置器和读取器；成员方法fight(Zues z)，每次调用此方法，随机生成一个10~100之间的整数攻击值，减少参数z的Energy值，并将结果输出，格式为“Titan攻击Zues，产生\*\*\*点攻击值，Zues当前Energy值为\*\*\*”。

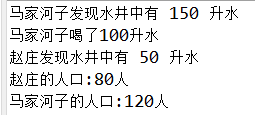
（2）     创建一个Zues类,有int型数据成员Energy；无参构造方法，可将Energy初始化为1000；带参构造方法，可将Energy初始化为参数指定值；数据成员Energy的设置器和读取器；成员方法fight(Titan t)，每次调用此方法，随机生成一个0~70之间的整数攻击值，减少参数z的Energy值，并将结果输出，格式为“Zues攻击Titan，产生\*\*\*点攻击值，Titan当前Energy值为\*\*\*”。

（3）     构建测试类，分别创建Titan对象和Zues对象，使用循环结构让Titan对象和Zues对象相互攻击，每次攻击完毕，判断Titan对象和Zues对象的Energy值，如果一方的Energy值小于0则停止循环，并输出结果：“Titan/Zues的Energy值为\*\*\*，已经失败，获胜者是Titan/Zues！”

扩展要求（Version2.0）：在类中添加游戏角色的交手次数、经验值、生命值之间的关系，并断定角色决斗的胜负，游戏若能采用GUI界面设计更佳。

3、（共饮同井水）编写程序模拟两个村庄共用同一口井水。编写一个Village 类(详见附件1：Village.java)，该类有一个静态的int型成员变量waterAmount，用于模拟井水的水量。在主类Land 的main()方法中创建两个村庄，一个村庄改变了waterAmount 的值，另一个村庄查看waterAmount 的值，源文件如下所示：

（1）画出Village类的UML类图。

（2）请将Land.java源文件中**[代码X]**(X:1--5)部分补充完整，使得程序运行效果如下图所示：

（3）在Land类的main方法中可否不定义name变量也能获得查看井水村庄的名字？若可行，修改程序使得能获得上图相同的运行结果。

(4) 【代码3】是否可以是Village.drinkWater(50)?为什么？

(5) 【代码4】是否可以是Village.getWaterAmount()?村庄的人数是否也可以采用Village.getPeopleNumber（）方式获得？

**三、实验要求**

1、加减运算能够接收不同类型的参数既可以实现复数与实数的加减、复数与复数的加减运算。

2、两游戏角色决斗。习题2中扩展部分规则参考：角色1（Titan）交手次数＋1，生命值－1,经验值＋2；角色2（Zue）交手次数＋1,生命值－2,经验值＋3。经验值每增加50时,生命值＋1；生命值<0判为负。生命值初始为1000,经验值初始为0。

3、给定二个不同的角色，判定交手的胜负关系。

4、实验报告给出决斗结果和交手次数

5、实验报告给出相关类图和所有源代码。

**四、实验环境**

1、PC微机；

2、DOS操作系统或 Windows 操作系统；

3、Eclipse程序集成环境。

**五、实验步骤**

内容一：

1、创建“复数”类Complex，定义实部、虚部成员

2、定义构造函数接收二个double参数用来初始化数据成员

3、定义二个复数运算函数plus()以及minus()各接收一个复数类型以实现复数与复数的加减运算。

4、定义二个复数运算函数plus()以及minus()各接收一个double类型以实现复数与与实数的加减运算。

4、定义一个打印方法。

5、在main()方法中创建复数对象并调用相关方法来验证。

内容二：

1、建立角色类，给出相应的成员，并能以生命值、经验值、能量值初始化角色对象。

2、在角色类中建立fight方法，接收一个角色类型的参数并与之“战斗”，返回胜者信息。

3、在主函数中初始化二个角色，调用fight方法。

**六、实验数据及处理结果**写出各个习题的算法（设计思路）、程序、结果并能对结果的正确性及产生的问题进行适当分析。

实验报告应包括以下几个部分：

1、程序流程图、UML类图；

2、程序的数据结构设计；

3、程序的源代码及相关注释

4、程序运行结果的分析。

题目一：

[源代码]

1. **package** exp4;
3. **import** java.util.Scanner;
4. **class** complexCalc{
5. **private** **double** real;
6. **private** **double** virtual;
8. **public** complexCalc(){real=0;virtual=0;};
9. **public** complexCalc(**double** real,**double** virtual){
10. **this**.real=real;
11. **this**.virtual=virtual;
12. }
14. **public** **void** addComplex(complexCalc obj){
15. real+=obj.real;
16. virtual+=obj.virtual;
17. }
19. **public** **void** subComplex(complexCalc obj){
20. real-=obj.real;
21. virtual-=obj.virtual;
22. }
24. **public** **void** showResult(){
25. **if** (virtual>0) System.out.print(real+"+"+virtual+"j");
26. **else** **if** (virtual<0) System.out.print(real+""+virtual+"i");
27. **else** System.out.print(real);
28. }
30. }

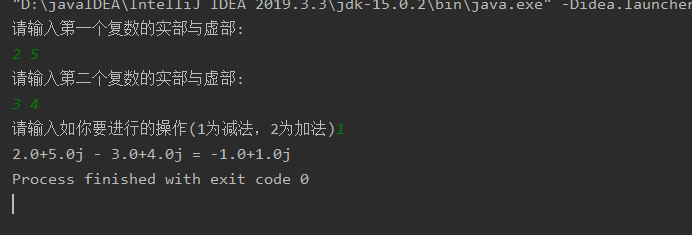
33. **public** **class** Complex {
34. **static** Scanner input=**new** Scanner(System.in);
35. **private** **static** **double** real1,real2,vir1,vir2;
36. **public** **static** **void** inputComplex(){
37. System.out.println("请输入第一个复数的实部与虚部:");
38. real1=input.nextDouble(); vir1=input.nextDouble();
39. System.out.println("请输入第二个复数的实部与虚部:");
40. real2=input.nextDouble(); vir2=input.nextDouble();
41. }
42. **public** **static** **void** main(String[] args){
43. inputComplex();
44. complexCalc c1=**new** complexCalc(real1,vir1);
45. complexCalc c2=**new** complexCalc(real2,vir2);
46. System.out.print("请输入如你要进行的操作(1为减法，2为加法)");
47. **int** choice=input.nextInt();
48. c1.showResult();
49. **if** (choice==1) {
50. System.out.print(" - ");
51. c1.subComplex(c2);
52. }**else**{
53. System.out.print(" + ");
54. c1.addComplex(c2);
55. }
56. c2.showResult();
57. System.out.print(" = ");
58. c1.showResult();
59. }
60. }

[设计思路及算法]

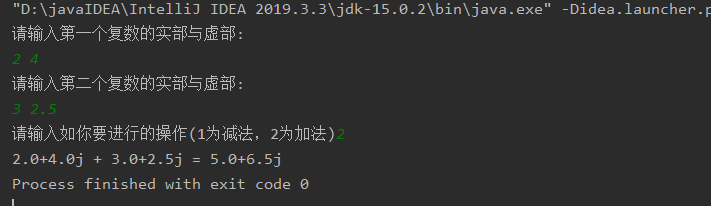
**思路：设计一个复数类，用有real,virtual两个Double型私有变量，分别代表复数的实部和虚部。然后再定义成员函数add(OBJ)表示两个复数类型的数相加，成员函数sub(obj)表示两个复数的相减。相加或者相减后的结果都存在调用成员函数的对象中，运算完成后再调用show函数显示复数的值即可。**

**[运行结果及截图]**

**1.两个复数相减**

****

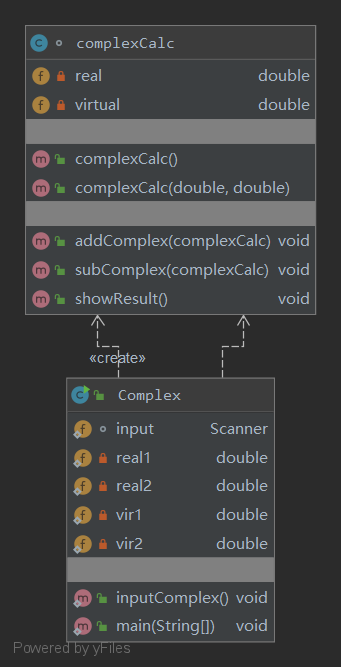
**2.两个复数相加**

****

**[分析]**

**这题主要是一个复数类的设计，思路和实现上都很容易，没有什么特殊注意的方面，逻辑也很简单。所以在实现的过程中并没有运到什么困难。**

**类图如下：**

****

题目二：

[源代码]

Titan.java

1. **package** exp4;
3. **import** java.util.Random;
5. **public** **class** Titan {
6. **private**  **int** Energy;
7. **private** **int** exp=0; //初始经验值,规定每次交手产生伤害的10%为经验值，向下取整
8. **private** **int** life=3; //初始生命值，当能量减为0时耗费一条生命，当经验值达到200时，增加一条生命
9. **private** **int** fightCount=0; //交手次数
11. **public** Titan(){Energy=80;}
12. **public** Titan(**int** En){Energy=En;}
14. **public** **void** setEnergy(**int** m){Energy=m;}
15. **public** **int** getEnergy() {**return** Energy;}
16. **public** **void** beAttacked(**int** m){Energy-=m;}
17. **public** **int** getLife(){**return** life;}
18. **public** **void** lifeDec(){life--;}
19. **public** **int** getFightCount(){**return** fightCount;}
21. **public** **void** fight(Zues z){
22. Random rand=**new** Random();  //随机产生攻击力
23. **int** f=rand.nextInt(90)+10;
24. exp+=f/10; //经验增加
25. **if**(exp>=200) {
26. life++; //生命增加
27. System.out.println("Titan等级提升，生命增加一条");
28. exp-=200;
29. }
30. fightCount++;//交手次数增加
31. z.beAttacked(f);   //对手掉血
32. **if** (z.getLife()>0 &&z.getEnergy()<=0) {
33. z.lifeDec(); //生命减一
34. z.setEnergy(1500);
35. }
36. System.out.println("Titan攻击了Zues,产生了"+f+"点伤害"
37. +" Zues当前的血量为"+z.getEnergy()+" Zues还剩下"+z.getLife()+"条命");
38. }
40. }

Zues.java

1. **package** exp4;
3. **import** java.util.Random;
5. **public** **class** Zues {
6. **private**  **int** Energy;
7. **private** **int** exp=0; //初始经验值,规定每次交手产生伤害的10%为经验值，向下取整
8. **private** **int** life=3; //初始生命值，当能量减为0时耗费一条生命，当经验值达到200时，增加一条生命
9. **private** **int** fightCount=0; //交手次数
11. **public** Zues(){Energy=1000;}
12. **public** Zues(**int** e){Energy=e;}
14. **public** **void** beAttacked(**int** attack){Energy-=attack;}
15. **public** **int** getEnergy(){**return** Energy;}
16. **public** **void** setEnergy(**int** m){Energy=m;}
17. **public** **int** getLife(){**return** life;}
18. **public** **void** lifeDec(){life--;}
19. **public** **int** getFightCount(){**return** fightCount;}
21. **public** **void** fight(Titan t){
22. Random rand=**new** Random();  //随机产生攻击力
23. **int** f=rand.nextInt(70);
24. exp+=f/10;
25. **if** (exp>=200) {
26. exp-=200;
27. life++;
28. System.out.println("Zues等级提升，生命增加一条");
29. }
30. fightCount++;
31. t.beAttacked(f);   //对手掉血
32. **if** (t.getLife()>0&&t.getEnergy()<=0) {
33. t.lifeDec();
34. t.setEnergy(1000);
35. }
36. System.out.println("Zues攻击了Titan,产生了"+f+"点伤害"
37. +" Titan当前的血量为"+t.getEnergy()+" Titan还剩下"+t.getLife()+"条命");
38. }

41. }

TextGame.java

1. **package** exp4;
3. lic **class** TextGame {
4. **public** **static** **void** main(String [] args){
5. Titan t=**new** Titan();
6. Zues z=**new** Zues();
7. t.setEnergy(1000);
8. z.setEnergy(1500);
9. **while** (!(t.getEnergy()<=0||z.getEnergy()<=0)){
10. **if** (t.getEnergy()>0) t.fight(z);
11. **if** (z.getEnergy()>0) z.fight(t);
12. }
13. **if** (t.getEnergy()<=0)System.out.println("Titan的Energy值为"+t.getEnergy()+"，已经失败，获胜者是Zues！");
14. **else** System.out.println("Zues的Energy值为"+z.getEnergy()+"，已经失败，获胜者是Titan！");
15. System.out.println("两人的交手次数为"+Math.min(t.getFightCount(),z.getFightCount()));
16. }

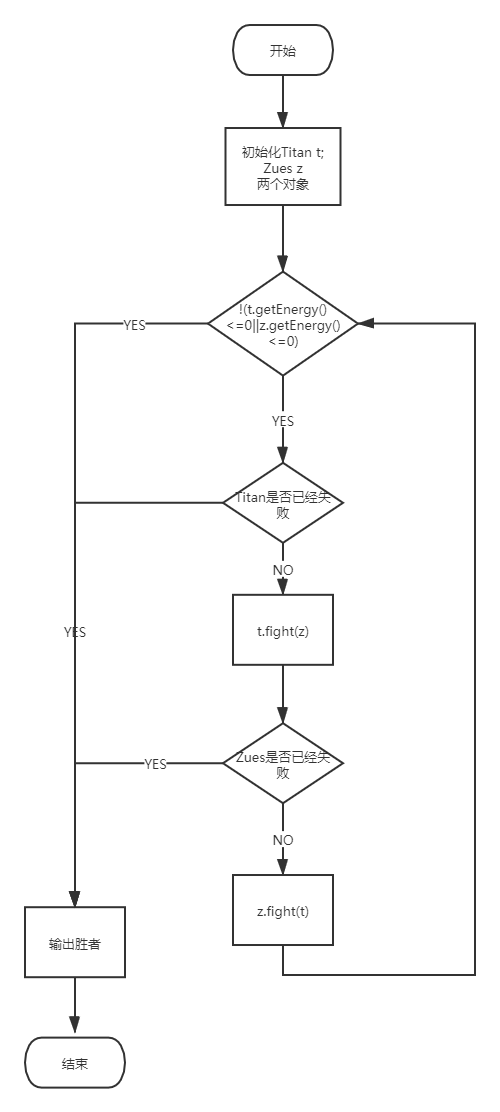
[设计算法和思路]

**思路：**设计Titan类和Zues类表示两个对手。每个类都有能量，生命，经验，交手次数这四个数据域。

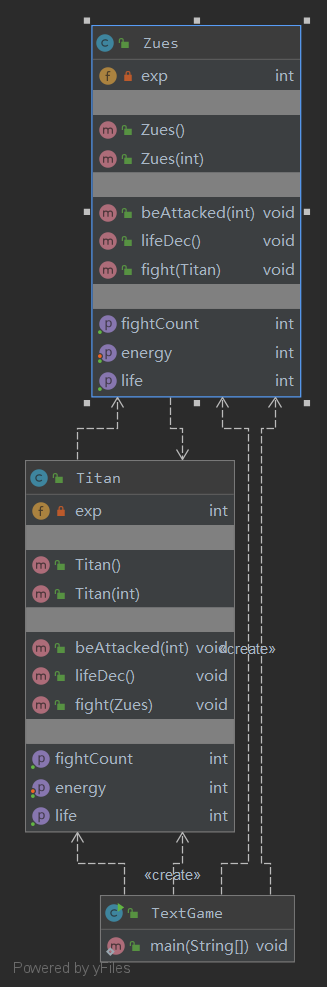
我们规定游戏规则：起初每个人物都有自己的生命值和能量值。经验和交手次数的初始值均为零。每次交手随机产生伤害，取当次伤害的10%（向下取整）作为该次出击所获得的经验值,当经验值满200时升级，增加一条生命值，当能量降为零时，生命值减1，能量恢复。双方依次出击，知道一方没有了生命值和能量则分出胜负。

该游戏的设计思路都很简单，实现也不难(非GUI界面),每个类都有fight(obj),beAttack(int)等一些基础成员函数来达到攻击和被攻击的效果。

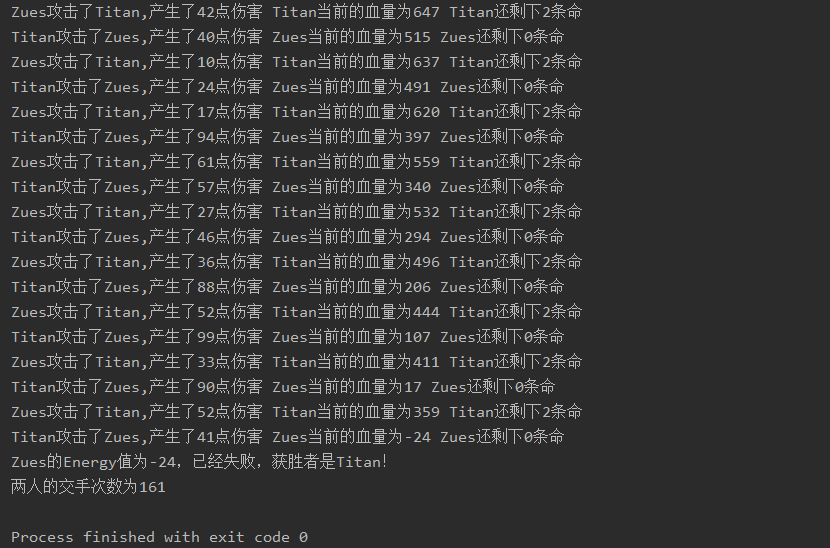
**流程图如下**：



**类图如下：**



[结果及截图]



[分析]

这题如果不做GUI界面的话还是很简单的，只要搞清楚要实现什么功能，有哪些数据需要处理，根据自己的思路或者也可以先画出一个流程图，依据流程图，实现起来其实就很简单，因此我在实现的过程中没有遇到什么困难。真正需要思考的还是GUI界面怎么处理的问题，这也是我需要改善的问题。

题目三

[源代码]

Village.java

1. **package** exp4;
3. lic **class** Village {
4. **static** **int** waterAmount;   //模拟水井的水量
5. **private** **int** peopleNumber;        //村庄的人数
6. String name;            //村庄的名字
7. Village(String s) {
8. name = s;
9. }
10. **static** **void** setWaterAmount(**int** m) {
11. **if**(m>0)
12. waterAmount = m;
13. }
14. **void** drinkWater(**int** n){
15. **if**( waterAmount-n>=0) {
16. waterAmount = waterAmount-n;
17. System.out.println(name+"喝了"+n+"升水");
18. }
19. **else**
20. waterAmount = 0;
21. }
22. **static** **int** getWaterAmount() {
23. **return** waterAmount;
24. }
25. **void** setPeopleNumber(**int** n) {
26. peopleNumber = n;
27. }
28. **int** getPeopleNumber() {
29. **return** peopleNumber;
30. }

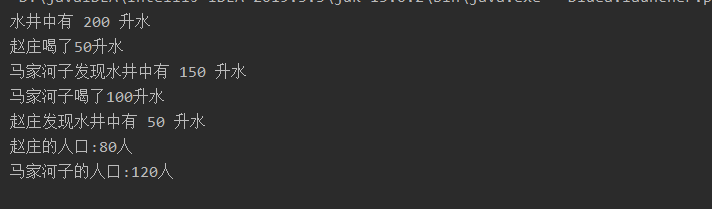
Land.java

1. **package** exp4;
3. **public** **class** Land {
4. **public** **static** **void** main(String args[]) {
5. Village.setWaterAmount(200);//用类名调用setWaterAmount(int m),并向参数传值200
6. **int** leftWater =Village.getWaterAmount(); //用Village类的类名访问waterAmount
7. System.out.println("水井中有 "+leftWater+" 升水");
8. Village zhaoZhuang,maJiaHeZi;
9. zhaoZhuang = **new** Village("赵庄");
10. maJiaHeZi = **new** Village("马家河子");
11. zhaoZhuang.setPeopleNumber(80);
12. maJiaHeZi.setPeopleNumber(120);
14. zhaoZhuang.drinkWater(50);//zhaoZhuang调用drinkWater(int n),并向参数传值50
15. leftWater =maJiaHeZi.getWaterAmount();//maJiaHeZi调用getWaterAmount()方法
16. String name=maJiaHeZi.name;
17. System.out.println(name+"发现水井中有 "+leftWater+" 升水");
18. maJiaHeZi.drinkWater(100);
19. leftWater =zhaoZhuang.getWaterAmount(); //zhaoZhuang调用getWaterAmount()方法
20. name=zhaoZhuang.name;
21. System.out.println(name+"发现水井中有 "+leftWater+" 升水");
23. **int** peopleNumber = zhaoZhuang.getPeopleNumber();
24. System.out.println("赵庄的人口:"+peopleNumber+"人");
25. peopleNumber = maJiaHeZi.getPeopleNumber();
26. System.out.println("马家河子的人口:"+peopleNumber+"人");
27. }
29. }

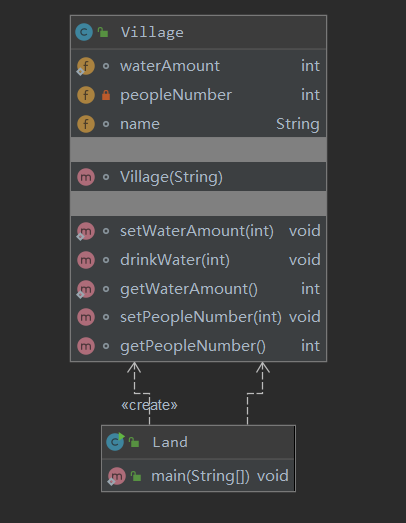
[设计思路计算法]

这题是一个代码填空题，能够正确填出答案需要正确理解类中静态变量和成员变量的区别。

[运行结果及截图]



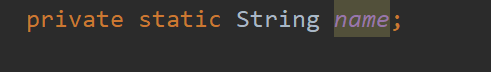
这题涉及的类的类图如下：



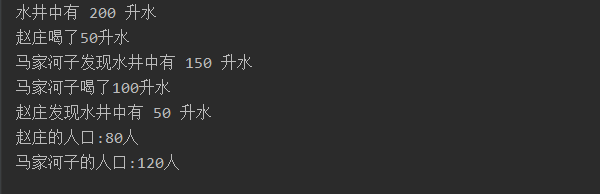
[分析]

（3）在Land类的main方法中可否不定义name变量也能获得查看井水村庄的名字？若可行，修改程序使得能获得上图相同的运行结果。

可以的，不过要设计为静态变量。如下图：



改变后运行结果如下：



与源程序相同。

(4) 【代码3】是否可以是Village.drinkWater(50)?为什么？

答：不能，因为drinkWater()函数不是静态成员函数，需要通过对象来调用。

(5) 【代码4】是否可以是Village.getWaterAmount()？来获得村庄的人数是否也可以采用Village.getPeopleNumber（）方式获得？

答：1.可以使用Village.getWaterAmount()来返回村庄的人数，因为getWaterAmount（）函数是静态函数可以用类名来调用。

2.不能使用Village.getPeopleNumber（）来获得，因为getPeopleNumber()是实例函数，不能通过类名调用。

**七、选做题（二选一）**

1. 课本第10章习题347页10.4（Mypoint类）。编程定义一个点类“MyPoint”。编写一个测试类PointTest 来创建MyPoint 类的对象，创建两个点的（0，0）和（10，30.5）并显示它们之间的距离，测试该类。

2. 课本第10章习题349页10.10（Queue类）。编程定义一个队列类（Queue类），添加从1到20的20个数字到队列中，然后将这些数字移除并显示它们。

题二：

[源代码]

Queue.java

1. **package** exp4;
3. **public** **class** Queue {
4. **private** **int**[] element;
5. **private** **int** maxLenght;
6. **private** **int** size=0;
8. **public**  Queue(){
9. maxLenght=8;
10. element=**new** **int**[maxLenght];
11. }
12. **public** **void** enqueue(**int** n){
13. **if** (size>=maxLenght) {
14. **int**[] newElement = **new** **int**[maxLenght \* 2];
15. System.arraycopy(element,0,newElement,0,element.length);
16. element = newElement;
17. maxLenght\*=2;
18. }
19. element[size++]=n;
20. }
21. **boolean** empty(){
22. **if** (size==0) **return** **true**;
23. **else** **return** **false**;
24. }
25. **public** **void** dequeue(){
26. **if** (empty()) {
27. System.out.println("对列为空，没有数值");
28. }**else**{
29. **int** pop=element[0];
30. **for** (**int** i=1;i<element.length;i++) element[i-1]=element[i];
31. size--;
32. System.out.println("弹出的值为:"+pop);
33. }
34. }
35. **public** **int** getSize() {**return** size;}
36. **public** **void** showSize(){
37. **if** (size==0) System.out.println("队列中没有数");
38. **else** {
39. System.out.println("现在队列中的数依次为");
40. **for** (**int** i=0;i<size;i++) System.out.print(element[i]+" ");
41. System.out.println();
42. }
43. }

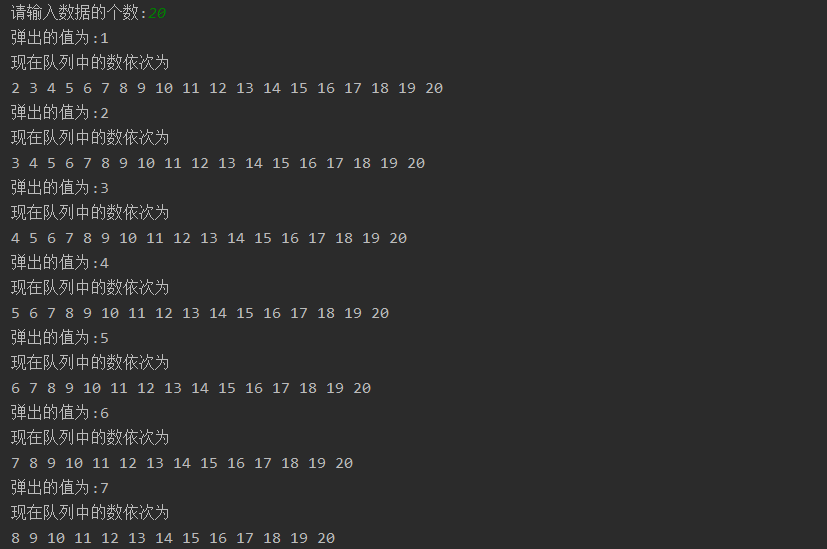
TextQueue.java

1. **package** exp4;
3. **import** java.util.Scanner;
5. **public** **class** TextQueue {
6. **public** **static** **void** main(String[] args){
7. Scanner input=**new** Scanner(System.in);
8. Queue q=**new** Queue();
9. System.out.print("请输入数据的个数:");
10. **int** n=input.nextInt();
11. **for** (**int** i=1;i<=n;i++) q.enqueue(i);
12. **while** (!q.empty()){
13. q.dequeue();
14. q.showSize();
15. }
16. }
18. }

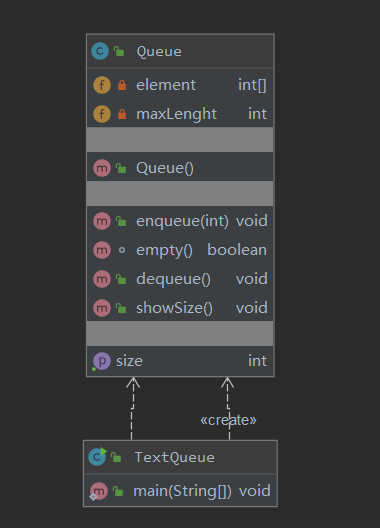
[设计思路计算法]

**思路：**我们要实现的是输入一个N，输出1-N的数在队列中依次弹出的结果。在队列类中，可以定义一个int element[]属组来存放数据。定义一个int size 来记录当前数组中数据的个数。再定义一个int maxLenght来记录当前数组的最大存放个数。在加入数据时，如果size.maxLenght是，马上申请一个最大长度为之前的两倍的数组，在通过System.arraycopy()函数将原来数组的数据复制到新数组里。再通过element=newElement 就可以相当于原数组的长度扩大了两倍。在弹出数据时，弹出element[0]，通过for循环将后面的数据往前移一位位置，在将size减一即可。所以我们设计一个enqueue()函数来增加元素，一个dequeue()函数来弹出数据，一个show()函数来显示数据即可。逻辑简单，实现也很简单。因为这题就是一个for循环的问题，所以这里不给出流程图了。

[运行结果及截图]



类图如下：



**八、实验总结与体会**

这次实验是对类和对象的使用，需要我们对类和对象有一个清楚的认识，理解静态变量和其他成员常量的区别等知识点。实验题目我觉得第二题的难度最大，主要体现在GUI界面的设计，这需要花大量时间去学习去思考学习。其他题目难度不大，在实现的过程中并没有遇到什么困难。因此我觉得这次实验有基础也有拓展(GUI设计)，是很符合我本人的对学习的安排的，在这次实验中，我的收获也是很多的！

//附件1：Village.java

public class Village {

static int waterAmount; //模拟水井的水量

private int peopleNumber; //村庄的人数

String name; //村庄的名字

Village(String s) {

name = s;

}

static void setWaterAmount(int m) {

if(m>0)

waterAmount = m;

}

void drinkWater(int n){

if( waterAmount-n>=0) {

waterAmount = waterAmount-n;

System.out.println(name+"喝了"+n+"升水");

}

else

waterAmount = 0;

}

static int getWaterAmount() {

return waterAmount;

}

void setPeopleNumber(int n) {

peopleNumber = n;

}

int getPeopleNumber() {

return peopleNumber;

}

}

//附件2：Land.java

public class Land {

public static void main(String args[]) {

【代码1】 //用类名调用setWaterAmount(int m),并向参数传值200

int leftWater =【代码2】; //用Village类的类名访问waterAmount

System.out.println("水井中有 "+leftWater+" 升水");

Village zhaoZhuang,maJiaHeZi;

zhaoZhuang = new Village("赵庄");

maJiaHeZi = new Village("马家河子");

zhaoZhuang.setPeopleNumber(80);

maJiaHeZi.setPeopleNumber(120);

【代码3】//zhaoZhuang调用drinkWater(int n),并向参数传值50

leftWater =【代码4】//maJiaHeZi调用getWaterAmount()方法

String name=maJiaHeZi.name;

System.out.println(name+"发现水井中有 "+leftWater+" 升水");

maJiaHeZi.drinkWater(100);

leftWater =【代码5】//zhaoZhuang调用getWaterAmount()方法

name=zhaoZhuang.name;

System.out.println(name+"发现水井中有 "+leftWater+" 升水");

int peopleNumber = zhaoZhuang.getPeopleNumber();

System.out.println("赵庄的人口:"+peopleNumber+"人");

peopleNumber = maJiaHeZi.getPeopleNumber();

System.out.println("马家河子的人口:"+peopleNumber+"人");

}

}