目录

[Day06.    Java](http://code.tarena.com.cn/CGBCode/cgb1903/01-java/%b1%ca%bc%c7/day06.htm#_Toc5722937)

[练习1   士兵](http://code.tarena.com.cn/CGBCode/cgb1903/01-java/%b1%ca%bc%c7/day06.htm#_Toc5722938)

[练习2   士兵的武器](http://code.tarena.com.cn/CGBCode/cgb1903/01-java/%b1%ca%bc%c7/day06.htm#_Toc5722939)

[1     二维数组](http://code.tarena.com.cn/CGBCode/cgb1903/01-java/%b1%ca%bc%c7/day06.htm#_Toc5722940)

[1.1       创建二维数组](http://code.tarena.com.cn/CGBCode/cgb1903/01-java/%b1%ca%bc%c7/day06.htm#_Toc5722941)

[1.2       遍历二维数组](http://code.tarena.com.cn/CGBCode/cgb1903/01-java/%b1%ca%bc%c7/day06.htm#_Toc5722942)

[练习3   二维数组](http://code.tarena.com.cn/CGBCode/cgb1903/01-java/%b1%ca%bc%c7/day06.htm#_Toc5722943)

[2     java.util.Arrays](http://code.tarena.com.cn/CGBCode/cgb1903/01-java/%b1%ca%bc%c7/day06.htm#_Toc5722944)

[3     数组的复制](http://code.tarena.com.cn/CGBCode/cgb1903/01-java/%b1%ca%bc%c7/day06.htm#_Toc5722945)

[练习4   数组复制](http://code.tarena.com.cn/CGBCode/cgb1903/01-java/%b1%ca%bc%c7/day06.htm#_Toc5722946)

[4     变量](http://code.tarena.com.cn/CGBCode/cgb1903/01-java/%b1%ca%bc%c7/day06.htm#_Toc5722947)

[4.1       局部变量](http://code.tarena.com.cn/CGBCode/cgb1903/01-java/%b1%ca%bc%c7/day06.htm#_Toc5722948)

[4.2       成员变量](http://code.tarena.com.cn/CGBCode/cgb1903/01-java/%b1%ca%bc%c7/day06.htm#_Toc5722949)

[5     方法递归](http://code.tarena.com.cn/CGBCode/cgb1903/01-java/%b1%ca%bc%c7/day06.htm#_Toc5722950)

[练习5   递归求阶乘](http://code.tarena.com.cn/CGBCode/cgb1903/01-java/%b1%ca%bc%c7/day06.htm#_Toc5722951)

[练习6   递归求斐波那契数](http://code.tarena.com.cn/CGBCode/cgb1903/01-java/%b1%ca%bc%c7/day06.htm#_Toc5722952)

[6     作业](http://code.tarena.com.cn/CGBCode/cgb1903/01-java/%b1%ca%bc%c7/day06.htm#_Toc5722953)

**Day06.  Java**

**package** day0504;

**import** java.util.Arrays;

**import** java.util.Random;

**public** **class** Test1 {

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

**int**[] a = **new** **int**[10];

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

        a[i] = i+1;

    }

    System.***out***.println(Arrays.*toString*(a));

    //把数组 a 传递到 shuffle() 方法打乱顺序

*shuffle*(a);

    System.***out***.println(Arrays.*toString*(a));

}

**private** **static** **void** shuffle(**int**[] a) {

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

**int** j = **new** Random().nextInt(a.length);

**int** t = a[i];

        a[i] = a[j];

        a[j] = t;

    }

}

}

**练习1  士兵**

Soldier士兵类

{

    id,

    blood = 100,

    go(),

    attack()

}

s1 = {

   id: 9527,

   blood: 100

}

s2 = {

   id: 9528,

   blood: 100

}

项目：day0601\_士兵

类：day0601.Test1

             Soldier

Soldier

**package** day0601;

**import** java.util.Random;

/\*

 \* 封装

 \* 把士兵相关的数据、运算方法，封装成一个“类”

 \*/

**public** **class** Soldier {

//成员变量、属性变量

**int** id; //默认值0

**int** blood = 100;

//成员方法

**public** **void** go() {

    System.***out***.println(**this**.id+"号士兵前进");

}

**public** **void** attack() {

**if**(**this**.blood == 0) {

        System.***out***.println("这是"+id+"号士兵的尸体");

**return**;

    }

    System.***out***.println(**this**.id+"号士兵进攻");

    //模拟受到伤害，产生一个随机的减血量

**int** d = **new** Random().nextInt(10);

    //如果减血量比当前血量大

**if**(d > **this**.blood) {

        d = **this**.blood;//有多少减多少

    }

**this**.blood -= d;

    System.***out***.println("血量："+**this**.blood);

**if**(**this**.blood == 0) {

        System.***out***.println(id+"号士兵阵亡");

    }

}

}

Test1

**package** day0601;

**public** **class** Test1 {

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

    /\*

     \* s1 = {

     \*     id: 9527,

     \*     blood: 100

     \* }

     \* s2 = {

     \*     id: 9528,

     \*     blood: 100

     \* }

     \*/

    //新建士兵对象，把它的内存地址保存到s1

    Soldier s1 = **new** Soldier();

    Soldier s2 = **new** Soldier();

    s1.id = 9527;

    s2.id = 9528;

    s1.go();

    s2.go();

    s2.attack();

    s2.attack();

    s2.attack();

    s2.attack();

    s2.attack();

}

}

**练习2  士兵的武器**

day0601\_士兵

复制

day0602\_今晚吃鸡

新建类： Weapon 武器类

{

    name,

    bullets,

    fire()

}

**package** day0602;

**import** java.util.Random;

**public** **class** Weapon {

String name;// 武器名称

**int** bullets;// 子弹数量

**public** Weapon(String name) {

**this**(name, 100);//构造方法之间调用

}

**public** Weapon(String name, **int** bullets) {

**this**.name = name;

**this**.bullets = bullets;

}

**public** **void** fire() {

    System.***out***.println("开火");

    //产生随机的发射子弹数量

**int** r = 3 + **new** Random().nextInt(10);

**if**(r > **this**.bullets) {

        r = **this**.bullets;

    }

**this**.bullets -= r;

**for** (**int** i = 0; i < r; i++) {

        System.***out***.print("突");

    }

    System.***out***.println("~");

    System.***out***.println("剩余子弹："+bullets);

}

**public** **void** reload() {

**this**.bullets = 100;

}

}

Test2

**package** day0602;

**import** java.util.Scanner;

**public** **class** Test2 {

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

    //新建两个武器实例，

    //调用发射方法、装载方法测试

    Weapon w1 = **new** Weapon("AKM");

    Weapon w2 = **new** Weapon("M416", 80);

    System.***out***.println("回车发射，输入r装载子弹");

**while**(**true**) {

        String s = **new** Scanner(System.***in***).nextLine();

**if** (s.equals("r")) {

           w1.reload();

           w2.reload();

           System.***out***.println("弹夹已装满");

**continue**;

        }

        w1.fire();

        w2.fire();

    }

}

}

Soldier

**package** day0602;

**import** java.util.Random;

/\*

 \* 封装

 \* 把士兵相关的数据、运算方法，封装成一个“类”

 \*/

**public** **class** Soldier {

//成员变量、属性变量

**int** id; //默认值0

**int** blood = 100;

Weapon weapon; // 默认值 null

//成员方法

**public** **void** go() {

    System.***out***.println(**this**.id+"号士兵前进");

}

**public** **void** attack() {

**if**(**this**.blood == 0) {

        System.***out***.println("这是"+id+"号士兵的尸体");

**return**;

    }

    System.***out***.println(**this**.id+"号士兵进攻");

    //如果有枪

**if** (**this**.weapon != **null**) {

**if** (**this**.weapon.bullets == 0) {

**this**.weapon.reload();//装载子弹

        }

**this**.weapon.fire(); //开火

    }

    //模拟受到伤害，产生一个随机的减血量

**int** d = **new** Random().nextInt(10);

    //如果减血量比当前血量大

**if**(d > **this**.blood) {

        d = **this**.blood;//有多少减多少

    }

**this**.blood -= d;

    System.***out***.println("血量："+**this**.blood);

**if**(**this**.blood == 0) {

        System.***out***.println(id+"号士兵阵亡");

    }

}

}

Test3

**package** day0602;

**import** java.util.Scanner;

/\*

 \* 100个士兵进入刺激战场

 \*

 \* Soldier[] 数组

 \* [

 \*     {

 \*         id: 1,

 \*         blood: 100,

 \*         weapon: {...}

 \*     },

 \*     {

 \*         id: 2,

 \*         blood: 100,

 \*         weapon: {...}

 \*     }

 \*     ....

 \* ]

 \*/

**public** **class** Test3 {

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

    System.***out***.println("欢迎进入刺激战场");

    //100个士兵，每个士兵拿一把枪

    Soldier[] a = **new** Soldier[100];

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

        //新建士兵实例，为id和weapon赋值

        Soldier s = **new** Soldier();

        s.id = i+1;

        s.weapon = **new** Weapon("AK"+(i+1));

        //把士兵放入数组

        a[i] = s;

    }

    System.***out***.println("按回车执行");

**while**(**true**) {

        //暂停等待，按回车后才会继续向后执行

**new** Scanner(System.***in***).nextLine();

        //每个士兵进攻一轮

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

**if** (a[i].blood != 0) {//如果有血

              a[i].attack();    //执行进攻

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

           }

        }

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

    }

}

}

**1   二维数组**

存放数组的数组

**1.1     创建二维数组**

int[][] a = new int[3][2];

  外围数组长度 3

  内部的三个数组长度 2

  共创建了4个数组

  内部数组中是默认值0

  外围数组中保存内部数组的地址

int[][] a = new int[3][];

  只创建一个外围数组，长度3

  三个位置都是 null 值

  可以之后再新建数组，放入

a[0] = new int[4];

a[1] = new int[]{5,1,5};

a[2] = new int[]{7,3,7,5,1};

int[][] a = {

{6,2,5,7},

{1,2,5},

{7,3,4,8,3}

};

**1.2     遍历二维数组**

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

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

    a[i][j]

}

}

**练习3  二维数组**

项目：day0603\_二维数组

类：day0603.Test1

**package** day0603;

**public** **class** Test1 {

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

**char**[][] a = {                     //j

           {'道','路','千','万','条'}, //0

           {'安','全','第','一','条'}, //1

           {'行','车','不','规','范'}, //2

           {'亲','人','两','行','泪'}  //3

    };   //i  0    1    2    3    4

    /\*

     \* 亲行安道

     \* 人车全路

     \* 两不第千

     \* 行规一万

     \* 泪范条条

     \*/

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

**for** (**int** j = a.length-1; j >=0; j--) {

           System.***out***.print(a[j][i]);

        }

        System.***out***.println();

    }

}

}

**2   java.util.Arrays**

数组的工具类，提供几个数组操作的工具方法

  **Arrays.toString(数组)**

把数组数据，连接成字符串

  **Arrays.sort(数组)**

数组排序

基本类型数组，优化的快速排序

引用类型数组，优化的合并排序

  **Arrays.binarySearch(数组, 目标值)**

二分法查找、折半查找

在有序数组中，查找目标值的位置下标

找不到，返回 -(插入点+1)

  **Arrays.copyOf(数组, 长度)**

把数组复制成一个指定长度的新数组

**3   数组的复制**

  **Arrays.copyOf(数组, 长度)**

会创建新数组

  **System.arraycopy(**

**原数组,**

**原数组起始位置,**

**目标数组,**

**目标数组起始位置,**

**复制的数据数量)**

   不会新建数组，目标数组要事先存在

**练习4  数组复制**

day0602\_今晚吃鸡

Test3

复制

Test4

阵亡士兵从数组删除，复制一个少一个位置的数组

a = [{},{},{x},{},{},{},{}]

             i

a = [{},{},{},{},{},{}]

         i

**package** day0602;

**import** java.util.Scanner;

/\*

 \* 100个士兵进入刺激战场

 \*

 \* Soldier[] 数组

 \* a = [

 \*     {

 \*         id: 1,

 \*         blood: 100,

 \*         weapon: {...}

 \*     },

 \*     {

 \*         id: 2,

 \*         blood: 100,

 \*         weapon: {...}

 \*     }

 \*     ....

 \* ]

 \*/

**public** **class** Test4 {

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

    System.***out***.println("欢迎进入刺激战场");

    //100个士兵，每个士兵拿一把枪

    Soldier[] a = **new** Soldier[100];

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

        //新建士兵实例，为id和weapon赋值

        Soldier s = **new** Soldier();

        s.id = i+1;

        s.weapon = **new** Weapon("AK"+(i+1));

        //把士兵放入数组

        a[i] = s;

    }

    System.***out***.println("按回车执行");

**while**(**true**) {

        //暂停等待，按回车后才会继续向后执行

**new** Scanner(System.***in***).nextLine();

        //每个士兵进攻一轮

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

           //if (a[i].blood != 0) {//如果有血

              a[i].attack();    //执行进攻

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

              /\*

              a = [{},{},{x},{},{},{},{}]

                           i

              a = [{},{},{},{},{},{}]

                       i

               \* \*/

**if** (a[i].blood == 0) {

                 Soldier[] b = **new** Soldier[a.length-1];

                 System.*arraycopy*(a,0,b,0,i);

                 System.*arraycopy*(a,i+1,b,i,a.length-i-1);

                 a = b;

                 i--;

              }

           //}

        }

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

    }

}

}

**4   变量**

**4.1     局部变量**

定义在方法中，或局部代码块中

  局部变量必须手动初始化（第一次赋值时，分配内存空间）

int a;

print(a); //错，没有初始化

a = 6; //初始化，分配内存空间

print(a);

  局部变量的作用域、作用范围，在它定义的大括号内有效

  在作用范围内，不能重复定义

void f() {

int a = 6;

if(...) {

    int a = 8;//错

    print(a);

    int b = 7;

}

print(b);//错

int b = 9;

}

**4.2     成员变量**

定义在类中

  自动初始化成默认值

  访问范围，受访问控制符的控制

  局部变量，可以和成员变量同名

**5   方法递归**

在方法中，调用自身

void f() {

f();

}

例如用 f() 方法求阶乘

f(5)

f(4)

f(3)

f(2)

f(5)

  5\*f(4)

      4\*f(3)

          3\*f(2)

              2\*f(1)

                  1\*f(0)

递归算法，一步一步地简化问题，简化成最简问题，再反向倒推求出结果

一般不同时做多次递归调用，否则运算量会急剧增加，这时考虑是否能用循环来代替递归

**练习5  递归求阶乘**

项目：day0604\_递归求阶乘

类：day0604.Test1

**package** day0604;

**import** java.util.Scanner;

**public** **class** Test1 {

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

    System.***out***.print("输入整数求阶乘：");

**int** n = **new** Scanner(System.***in***).nextInt();

**long** r = *f*(n);

    System.***out***.println(r);

}

**private** **static** **long** f(**int** n) {

    //最简问题

**if** (n == 0) {

**return** 1;//从最简问题向前返回

    }

**return** n \* *f*(n-1); //递归调用

}

}

**练习6  递归求斐波那契数**

1 1 2 3 5 8 13 21 34 55 89 144 233 ...

项目：day0605\_递归求斐波那契数

类：day0605.Test1

**package** day0605;

**import** java.util.Scanner;

**public** **class** Test1 {

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

    System.***out***.print("求第几个斐波那契数：");

**int** n = **new** Scanner(System.***in***).nextInt();

**long** r = *g*(n);

    System.***out***.println(r);

}

**private** **static** **long** g(**int** n) {

    /\*

     \* 1 1 2 3 5 8

     \*         a b

     \*

     \* b=a+b

     \* a=b-a

     \*/

**long** a = 1;

**long** b = 1;

    //从第3个求到第n个

**for** (**int** i = 3; i <= n; i++) {

        b=a+b;

        a=b-a;

    }

**return** b;

}

//反例，多次递归，运算量大

**private** **static** **long** f(**int** n) {

    //最简问题

**if** (n==1 || n==2) {

**return** 1;

    }

**return** *f*(n-1) + *f*(n-2);

}

}

**6   作业**

  重写

  day0601\_士兵

  day0602\_今晚吃鸡