

JAVA-基础语法

数据类型

字符串操作

字符串基本操作

字符串相加

字符串访问单个字符

字符串长度

字符串拼接操作

类型转换

隐式转换

取值范围：数据类型

强制转换

Scanner键盘输入

方法（函数）

定义格式

方法调用内存图解

带参数方法的定义和调用

方法的重载

循环

Random产生随机数

基本语法

小项目：猜数字小游戏

数组

数组初始化

动态获取数组长度

数组遍历快捷键

数组引入方法（函数）

数组内存图和方法参数传递

两个数组指向相同内存图

二维数组

格式(静态初始化)

格式(动态初始化)

将创建好的一维数组存入二维数组

数组项目：产生随机验证码

数据类型

表数范围

整数类型

byte类型：-128~127

long类型：很大

short类型：正负三万左右

布尔类型

boolean

浮点类型

float类型：用float表示小数后面需要加f（3.14f）

double类型：与c相同

字符串类型

String

字符串操作

字符串基本操作

字符串相加

字符串相加

Java

复制代码

```
1 public static void main(String[] args)
2 {
3     String code = "rs";
4     String code1 = "op";
5     String code2 = code1+code;
6     System.out.println(code2);//rsop
7 }
```

字符串访问单个字符

在Java中访问字符串中的字符要使用`charAt()`方法，比如`s.charAt(0)`。

字符串单独访问某个字符

Java

复制代码

```
1 public static void main(String[] args)
2 {
3     String code = "rs";
4     String code1 = "op";
5     System.out.println(code.charAt(1)); //s
6 } //charAt(index)
```

字符串长度

字符串长度

Java

复制代码

```
1 public static void main(String[] args)
2 {
3     String code = "rs";
4     String code1 = "op";
5     System.out.println(code.length()); //2
6 }
```

字符串拼接操作

当 + 操作中，遇到了字符串，这时+就是字符串连接符

字符串

Java

复制代码

```
1 public static void main(String[] args) {
2     System.out.println(1+23); //输出24
3     System.out.println("年龄为："+23); //输出年龄为:23
4     System.out.println(1+99+"年黑马"); //输出100年黑马
5     System.out.println("年龄为：" + 23 + 1); //输出年龄为: 231
6 }
```

类型转换

隐式转换

取值范围：数据类型

把一个取值范围小的数值或变量，赋值给另一个取值范围大的变量

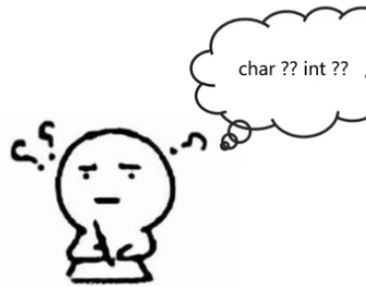
取值范围从小到大：

byte→short→int→long→float→double

运算过程中的隐式转换

- 取值范围小的数据，和取值范围大的数据进行运算，小的会先提升为大的之后，再进行运算
- byte short char 三种数据在运算的时候，都会提升为int，然后再进行运算

```
public class Test {  
    public static void main(String[] args) {  
        int a = 1;  
        char b = 'a';  
        int c = a + b;  
        System.out.println(c);  
    }  
}
```



▼ 隐式转换

Java

📄 复制代码

```
1 public static void main(String[] args) {  
2     int a=10;  
3     double b=a;  
4     System.out.println(b);//10.0  
5     long num1=123L;  
6     float num2=num1;  
7     System.out.println(num2);//123.0  
8 }
```

强制转换

把一个取值范围大的数值，赋值给另一个取值范围小的变量

不允许直接赋值，需要加入强制转换

```
1 public static void main(String[] args) {
2     double a = 12.3;
3     int b = (int)a;
4     System.out.println(b); //12
5     int c=141;
6     byte d=(byte) c;
7     System.out.println(d); // -116, 计算机组成原理
8 }
```

扩展运算符类似+=, -=等自带强制转换

Scanner键盘输入

```
1 import java.util.Scanner;
2 public class helloworld {
3
4     public static void main(String[] args) {
5         Scanner sc = new Scanner(System.in);
6         System.out.println("请输入您的名字");
7         String name = sc.next();
8         System.out.println("请输入您的身高");
9         int height = sc.nextInt();
10        System.out.println("请输入您的名言");
11        String profile = sc.next();
12        System.out.println("请输入您的婚宴状况");
13        boolean marrige = sc.nextBoolean();
14    }
15 }
```

方法（函数）

定义格式

```
1 public static 返回值类型 方法名(参数){  
2     //方法体  
3     return 数据; //取决于返回值类型  
4 }  
5
```

方法调用内存图解



多一句没有，少一句不行，用最短时间，教会最实用的技术！

方法调用内存图解

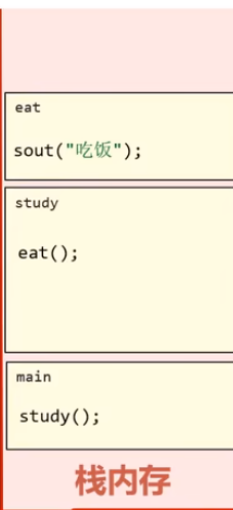
```
public class MethodDemo2 {  
    public static void main(String[] args) {  
        study();  
    }  
  
    public static void sleep(){  
        System.out.println("睡觉");  
    }  
  
    public static void eat(){  
        System.out.println("吃饭");  
    }  
  
    public static void study(){  
        eat();  
  
        System.out.println("学习");  
  
        sleep();  
    }  
}
```

吃饭

MethodDemo2.class

main
study
sleep
eat

方法区



高级软件人才培养专家

带参数方法的定义和调用

与c相同

方法的重载

- 简单记：同一个类中，方法名相同，参数不同的方法
参数不同：个数不同、类型不同、顺序不同

注意：识别方法之间是否是重载关系，只看方法名和参数，跟返回值无关。

循环

▼ 循环截止的break方式

Java

📄 复制代码

```
1 public class helloworld {
2
3     public static void main(String[] args) {
4         load();
5     }
6
7     public static void load() {
8         Scanner sc = new Scanner(System.in);
9
10        lo://标号，给循环起名字
11        while(true)
12        {
13            System.out.println("请输入您的选择: ");
14            int choice = sc.nextInt();
15            switch (choice){
16                case 1:
17                    System.out.println("添加学生");
18                    break;
19                case 2:
20                    System.out.println("删除学生");
21                    break;
22                case 3:
23                    System.out.println("感谢您的使用");
24                    break lo;
25
26            }
27        }
28    }
29 }
```

其他循环处与c相同

Random产生随机数

基本语法

random里的数是从0开始左闭右开区间

需求：产生三个1-100随机数

▼ 1-100

Java

📄 复制代码

```
1 public static void main(String[] args) {  
2     for(int i=1;i<4;i++)  
3         randomdemo();  
4 }  
5 public static void randomdemo(){  
6     Random r = new Random();  
7     int num = r.nextInt(100)+1;  
8     System.out.println(num);  
9 }
```

需求：产生一个20-80随机数

▼ 20-80

Java

📄 复制代码

```
1 public static void main(String[] args) {  
2     // for(int i=1;i<4;i++)  
3     randomdemo();  
4 }  
5 public static void randomdemo(){  
6     Random r = new Random();  
7     int num = r.nextInt(61)+20;  
8     System.out.println(num);  
9 }
```

小项目：猜数字小游戏


```
1  import java.util.Random;
2  import java.util.Scanner;
3  public class helloworld {
4
5      public static void main(String[] args) {
6          randomdemo();
7      }
8      public static void randomdemo(){
9          Random r = new Random();
10         Scanner sc = new Scanner(System.in);
11         int randomnum = r.nextInt(100)+1;
12
13         while(true){
14             System.out.println("请输入");
15             int scnumber=sc.nextInt();
16             if(scnumber>randomnum)
17                 System.out.println("猜大了");
18             else if(scnumber<randomnum)
19                 System.out.println("猜小了");
20             else{
21                 System.out.println("猜对了");
22                 break;
23             }
24         }
25         System.out.println("恭喜您");
26
27     }
28 }
```

数组

数组初始化

1: `int[] arr = new int[3]`

2: `int array[]= new int[3]`

动态获取数组长度

数组名.length: 元素的个数

数组遍历快捷键

数组名.fori

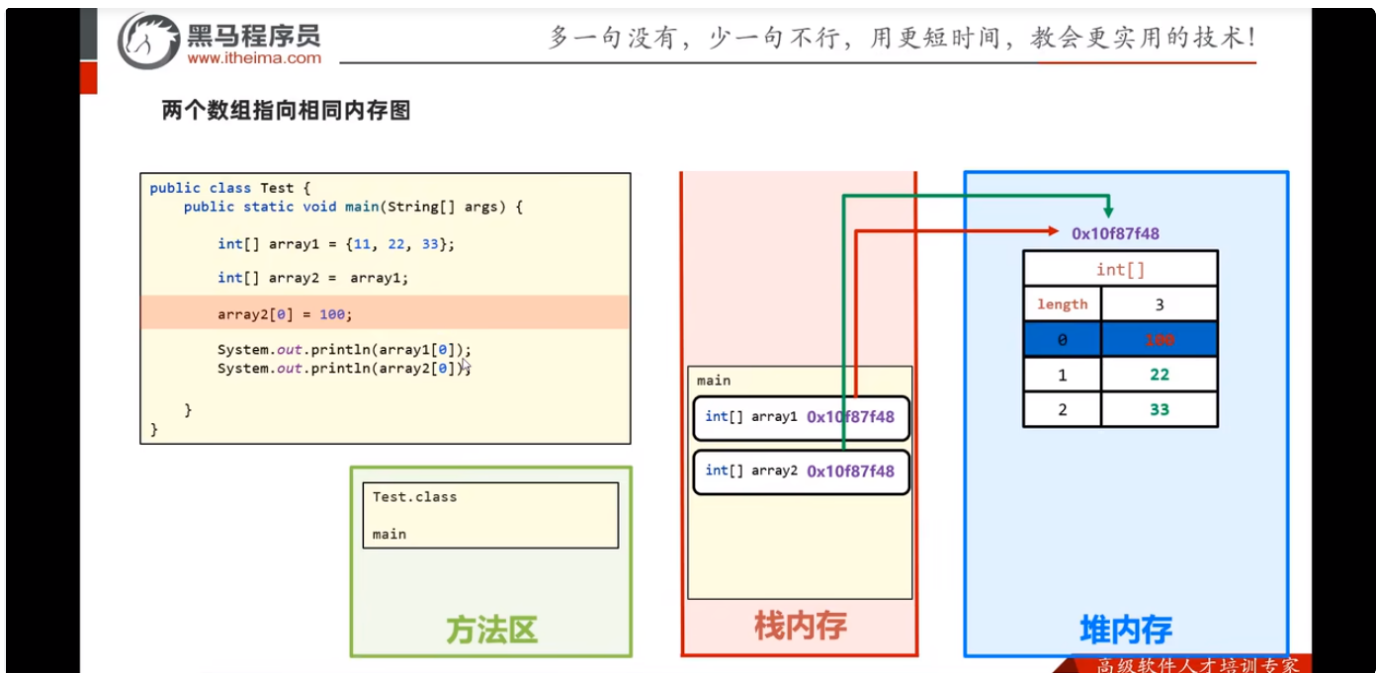
自动生成数组遍历for循环框架

数组引入方法（函数）

```
public static int arraytest(int[] arr1){}
```

数组内存图和方法参数传递

两个数组指向相同内存图



Java

复制代码

```
1 public static void main(String[] args) {  
2     int[] array1={11,22,33};  
3     int[] array2=array1;  
4     array2[0]=100;  
5     System.out.println(array1[0]);  
6     System.out.println(array2[0]);  
7 }
```

二维数组

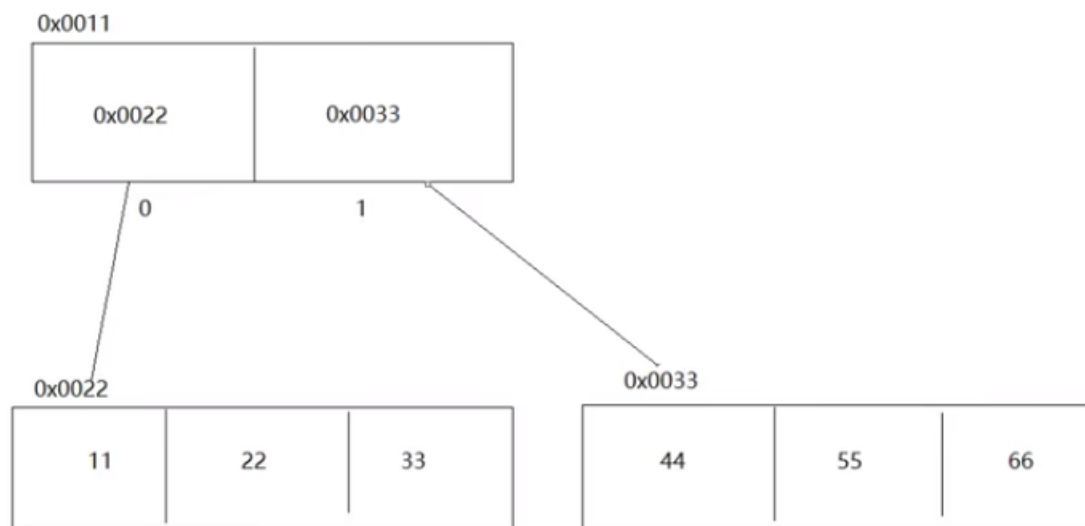
格式(静态初始化)

```
int[][] arr = {{1,2,3},{5,6,7}};
```

格式(动态初始化)

```
int[][] arr = new int[2][3];
```

二维数组存储一维数组时，具体存储是每个一维数组的地址值



将创建好的一维数组存入二维数组

```

1 public static void main(String[] args) {
2     int[] array1={11,22,33};
3     int[] array2={44,55,66};
4     int[][] arr = new int[2][3];
5     arr[0]=array1;
6     arr[1]=array2;
7     for (int i = 0; i < arr.length; i++) {
8         for (int j = 0; j < arr[i].length; j++) {
9             System.out.println(arr[i][j]);
10        }
11    }
12
13 }

```

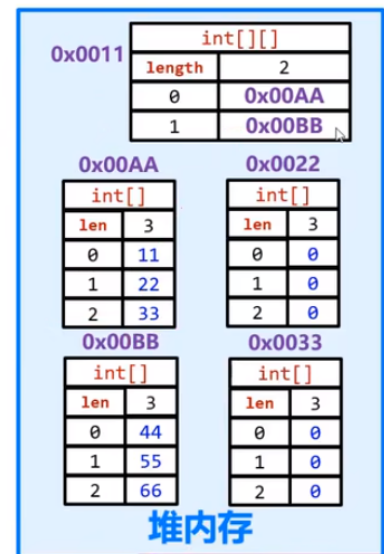
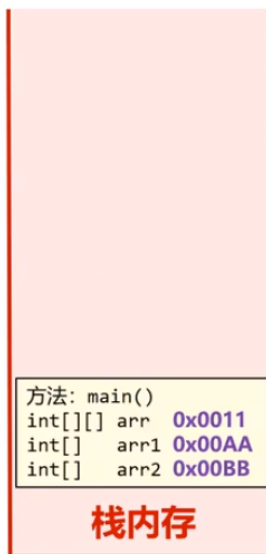
内存图

```

public static void main(String[] args) {
    int[][] arr = new int[2][3];
    int[] arr1 = {11, 22, 33};
    int[] arr2 = {44, 55, 66};

    arr[0] = arr1;
    arr[1] = arr2;
}

```



高级软件人才培养专家

数组项目：产生随机验证码

[b站讲解随机产生验证码](#)

```
1 public static void main(String[] args)
2 {
3     char[] chs=randomchs();
4     input(chs);
5 }
6 public static char[] randomchs(){
7     int index=0;
8     char[] chs=new char[26+26+10];
9     for(char c= 'a';c<='z';c++)
10 {
11     chs[index++]=c;
12 }
13     for(char c= 'A';c<='Z';c++)
14 {
15     chs[index++]=c;
16 }
17     for(char c='0';c<='9';c++)
18 {
19     chs[index++]=c;
20 }//所有验证码组成存入数组chs中
21     return chs;
22 }
23 }
24 public static void input(char chs[]){
25     Random r = new Random();
26     String checkcode = "";
27     for(int i=0;i<5;i++)
28 {
29         int index1=r.nextInt(chs.length);
30         checkcode += chs[index1];
31     }
32     System.out.println("产生的验证码为: "+checkcode);
33 }
```