

# AP Computer Science A (AP计算机科学A) 考前冲刺资料整理



## # 主方法

```
JAVA
1 public static void main(String[] args) {
2     // 在这里输入代码
3 }
4 }
```

## # 优先级

```
JAVA
1 优先级如下，优先级从高往低排列
2 a) ()
3 b) ! , ++ , -
4 c) * , / , %
5 d) + , -
6 e) < , > , <= , >=
7 f) == , !=
8 g) &&
9 h) ||
10 i) = , += , -= , *= , /= , %=
```

## # 随机数

```
JAVA
1 math.random();
2
3 随机数公式:
4 lowValue ≤ x < highValue:
5 double x = (highValue - lowValue) * Math.random() + lowValue;
```

## # 类

```
JAVA
1 public class 类名 {
2     //类的内容
3 }
```

## # 方法

```
JAVA
1 public/private + 可能有 static 关键字 + 返回值类型 + 方法名(形式参数列表){
2     方法体
3 }
```

### static关键字

带有static关键字定义的方法被称为静态方法（static method）实际上也被称为类方法（class method）。相反，不带有static方法的就被称为实例方法了。静态方法并不随着我们的对象的创建才能使用，静态方法是与对象无关的。静态方法能够直接通过类的调用来使用。换句话说，实例方法是面向实例，也就是对象的（对象就是类的实例），而静态方法是面向类的。

## # 对象

```
JAVA
类名 变量名 = new 类名();
```

## # 多态

当Java程序在执行时，具体执行多态方法的哪一种形态，则取决于，调用这一方法的对象，实际指向的对象类型（也就是在new语句后紧跟着的类型），而不是这一对象引用的对象类型（也就是在new语句中，实际对象前面的那个类型）。

//假设getID()方法是GradStudent类中特有的一个public实例方法，没有在Student类中被定义，GradStudent类继承自Student类。

```
JAVA
1 Student student = new GradStudent();
2 GradStudent gradstudent = new GradStudent();
```

```
3 int x = student.getID(); //编译时错误
4 int y = gradstudent.getID(); //正确student对象无getID() 方法，这不是多态
```

## # 接口

```
JAVA
1 //创建接口
2 public interface 接口名
3 {
4     //内容
5 }
6
7 //实现接口
8 public class 类名 implements 接口名{
9     //类的内容
10 }
```

## # Object类

```
JAVA
1 toString()
2 //把对象内容以String型返回的方法。
3
4 equals()
5 //比较内容的方法
```

## # String类

```
JAVA
1 int length()
2 //以int型返回字符串的长度。
3
4 String substring(int startIndex)
5 //抽取startIndex位置到字符串末尾的字符串
6
7 String substring(int startIndex, int endIndex)
8 //抽取startIndex位置到endIndex的字符串（包括startIndex位置的字母，不包括endI
9
10 int indexOf(String str)
11 //该方法的作用是，返回一个子字符串的第一个字符在原字符串中的位置。
```

```
12
13 compareTo()
14 //该方法用于按字典顺序比较两个字符串。
15 //先比较对应字符的大小(ASCII编码),如果第一个字符和参数的第一个字符不等,结束比较,
16 //如果参数字符串等于此字符串,则返回值 0; 如果此字符串小于字符串参数,则返回一个
```

## # 瀑布模型

- 1.程序需求分析
- 2.程序设计
- 3.程序编写
- 4.调试和测试

## # bug类型

编译错误(compile-time error)指的是在编译时的错误。比如语言使用错误,调用错误,缺少符号等。

运行错误(run-time error)是指编译时未报错,但是运行时出现的错误。比如子类函数在调用父类函数时出现的问题,数据中分母为0,无限循环。在Java中,这些错误都会以Exception的方式抛出(throws)。

逻辑错误(intent or logic error)是并没有明显提示,程序可以正常运行。但结果并不是我们想要的。

## # 开发方式

自下而上开发(bottom-up):写多个零散的代码块,再逐渐组合连接关系。

自上而下开发(top-down):写出主体块,再分解为多个零散的代码。

## # 数组

```
JAVA
1 //定义数组
2 double data[];
3 double[] data;
4 //赋值
5 data = new double[15];
6 //定义并赋值
7 double[] data = new double[15];
8 //数组长度
9 arrayName.length
```

## # 动态数组

```
JAVA
1 //构建
2 ArrayList<E> name = new ArrayList<E>();
3
4 //增加数据
5 boolean add(E obj);
6
7 int size();
8 //返回list中的元素数量
9
10 E get(int index);
11 //返回在index的元素
12
13 E set(int index, E element);
14 //把在index位置的元素换成element, 返回原先在index的元素
15
16 void add(int index,E element);
17 //在index处加入element, 把index处及其之后的元素都会向后移动1个位置。List的大小
18
19 E remove(int index)
20 //删除在index位置的元素, 返回原来index位置的元素。在index位置后的元素会向前移动。
```

## # 二维数组

```
JAVA
1 E[][] name = new E[a][b];
2 //其中, E是类型名, a、b是数组的长度。
3
4 name[r][c];
5 //r代表row行, c代表列。
```

## # 包装类

```
JAVA
1 //声明一个Integer对象
2 Integer num = 10;
3 //以上声明就是使用了自动封包, 等价于:
4 Integer num = new Integer(10);
```

## # 词汇表

英文

中文

Assume	假设
Expression	表达
represents	代表
correct	正确
executing	实施
parameters	参数
evaluating	估计
segment	分割
declaration	声明
private	私有的
protected	受保护的
public	公共的
abstract	声明抽象
class	类
extends	继承
final	最终值,不可改变的
implements	实现 (接口)
interface	接口
new	新,创建
static	静态
break	跳出循环
continue	继续
else	否则
for	循环
if	如果
instanceof	实例

英文

中文

return	返回
while	循环
import	引入
package	包
boolean	布尔型
byte	字节型
char	字符型
double	双精度浮点
float	单精度浮点
int	整型
long	长整型
short	短整型
super	父类
this	本类
void	无返回值
null	空

ROY'S BLOG  
blog.ranfu.net