

月考题：

1. 50道单选题，几乎都是代码分析题，不考单词，考到7月25号的内容为止，满分100分，60分及格

常见面试题：共7题，必须背下来

1. java的八种基本数据类型是什么？

- byte:字节型，用于存储整数的，占用1个字节，范围-128到127
- short:短整型，用于存储整数的，占用2个字节，范围-32768到32767
- int:整型，用于存储整数的，占用4个字节，范围-2³¹到2³¹-1
- long:长整型，用于存储整数的，占用8个字节，范围-2⁶³到2⁶³-1
- float:单精度浮点型，用于存储小数的，占用4个字节，不能表示精确的值
- double:双精度浮点型，用于存储小数的，占用8个字节，不用表示精确的值
- boolean:布尔型，用于存储true或false的，占用1个字节
- char:字符型，采用Unicode字符编码格式，用于存储单个字符，占用2个字节

2. switch能作用在哪些类型的变量上？

- 可以作用于:byte,short,int,char,String,枚举类型，其它类型都是不允许的

3. 重写与重载的区别:

- 重写:发生在父子类中，方法名相同，参数列表相同

当派生类觉得超类的行为不好时，可以通过重写来修改超类中的方法

- 重载:发生在同一类中，方法名相同，参数列表不同

是完全不同的方法，只是正好方法名相同而已

4. 实例变量与静态变量的区别:

- 实例变量:属于对象的，在创建对象时存储在堆中，创建了多少个对象，则实例变量就会在堆中存储几份，需要通过引用打点来访问
- 静态变量:属于类的，在类被加载时存储在方法区中，无论创建了多少个对象，静态变量在内存中都只有一份，常常通过类名打点来访问

5. 问: java是值传递还是引用传递？

- java中无论是基本类型还是引用类型，都是值传递
 - 对于基本类型而言，传递的是具体的值的副本
 - 对于引用类型而言，传递的是具体的地址的副本

注意:网站上说java有引用传递的文章都是错误的~!

6. String str = new String("abc"); 创建了几个字符串对象？

答: 2个。一个是"abc"字面量对象并缓存在了字符串常量池中。另一个是new出来的String对象。

7. ==和equals()的区别:

- ==: 若为基本类型则比较的是具体的值是否相同，若为引用类型则比较的是引用的地址是否相同。

```
int a=5,b=5;
System.out.println(a==b); //true, 比较a与b的值是否相同
Student zs = new Student();
Student ls = new Student();
System.out.println(zs==ls); //比较zs与ls的地址是否相同
```

- equals(): 只能用于引用类型之间, 是在Object类中定义的, 默认还是比较==(即比较地址), 一般没有参考意义, 常常被重写来比较具体的属性值是否相同。String类重写了equals()来比较字符串的内容是否相同。

```
String s1 = new String("hello");
String s2 = new String("hello");
System.out.println(s1==s2); //false, 比较的是地址
System.out.println(s1.equals(s2)); //true, 比较的是字符串的内容
```

语言基础内容列表:

1. java开发环境、JVM、JRE、JDK
2. 变量、八种基本数据类型、类型间的转换
3. 运算符、分支结构(上)
4. 分支结构(下)、循环结构(上)
5. 循环结构(下)、数组(上)
6. 数组(下)、方法

语言基础知识点:

1. 标识符的命名规则:
2. 八种基本数据类型有哪些? 每种类型所占的字节数?
 - 整数直接量默认为int类型、浮点数直接量默认为double类型
3. 字符串连接符运算符: +

```
System.out.println(5+10+"你"); //15你
System.out.println(5+"你"+10); //5你10
```

4. 算术运算符: +、-、*、/、%、++、--

```
int a = 5;
int b = a++;
System.out.println(a); //6
System.out.println(b); //5
```

5. 逻辑运算符:
 - &&: 短路与(并且)----见false则false
 - ||: 短路或(或者)-----见true则true
 - !: 逻辑非(取反)-----非真则假, 非假则真

6. 分支结构: switch...case

- 看到break会停, 看不到就往下一直串, 一直串到底
- default可以写在switch里的任何地方, 但无论写在哪儿, 都一定是先匹配case, 当所有case都未匹配时才执行

```
int num = 6;
switch(num){
    default: //所有case都未匹配时才执行
        .... //此处若没有break, 依然会继续穿到下面的代码去
    case 1:
        ....
    case 2:
        ....
}
```

7. 三种循环结构的执行规则: 带数

- while: 先判断后执行, 有可能一次都不执行
- do...while: 先执行后判断, 至少执行一次

8. 第一阶段共三种异常:

- ArrayIndexOutOfBoundsException: 数组下标越界异常
 - 数组下标从0开始, 最大到(数组的长度-1), 超出范围时会发生下标越界异常
- NullPointerException: 空指针异常
 - 用null值做操作时会发生空指针异常
- ClassCastException: 类型转换异常
 - 强制类型转换时若不符合那两个成功的条件时会发生类型转换异常

9. 数组下标从0开始, 最大到(数组的长度-1)

数组的长度是length属性、字符串的长度是length方法

一旦new数组了, 则数组元素都会有默认值

```
byte, short, int, long, char-----0
float, double-----0.0
boolean-----false
引用类型-----null
```

10. 关键字:

- break: 可以用在switch和循环中, 用在switch中用于跳出switch, 用在循环中用于跳出循环
- continue: 只能用在循环中, 跳过循环体中剩余语句而进入下一次循环
- void: 方法无返回值
- return: 结束方法
 - return 值; //1)结束方法 2)返回结果给调用方-----用在有返回值的方法中
 - return; //1)结束方法-----用在无返回值的方法中

11. System.out.println(2+2); //4

System.out.println('2'+2); //100, '2'的码50, 加上, '2'的码50

12. Math.random(): 生成0.0到0.999999999999.....的随机数

13. while的语法、do...while的语法、for的语法：

```
while(boolean){
    语句块
}

do{
    语句块
}while(boolean);

for(要素1;要素2;要素3){
    语句块
}
```

面向对象、API基础内容列表：

1. 类和对象、方法的重载
2. 构造方法、this
3. 引用类型数组、继承、super
4. 向上造型、方法的重写
5. package和import、访问控制修饰符、final、static
6. static final常量、抽象方法、抽象类
7. 成员内部类、匿名内部类
8. 接口
9. 多态
10. 内存管理、面向对象三大特征总结
11. String、String常用方法、StringBuilder、StringBuilder常用方法
12. 正则表达式、String支持与正则相关的方法、Object、包装类

面向对象知识点：

11. 实例成员是由引用名打点来访问的，静态成员是由类名打点来访问的
静态方法中只能直接访问静态成员，实例方法中可以直接访问实例的也可以直接访问静态
静态变量常常也称为类变量-----属于类的叫类变量(也叫做静态变量)
12. 访问权限从低到高依次为: private，默认的(default), protected，public
类/接口的访问权限只能是public或默认的，类中成员的访问权限如上4种都可以
13. 向上造型：前面是超类型/接口，后面是派生类型

```
//假设Aoo是超类，Boo是派生类
Aoo o1 = new Aoo(); //没造
Boo o2 = new Boo(); //没造
Aoo o3 = new Boo(); //向上造型
o1 = o2; //向上造型
```

14. final的变量不能被改变、final的方法不能被重写、final的类不能被继承
15. 程序的执行过程(若有的话):

- 超类的static块
- 衍生类的static块
- 超类的构造方法
- 衍生类的构造方法

```
//如果一个类中出现多个静态块，则会由上到下执行----如下代码先执行静态块1，再执行静态块2
class Aoo{
    static{
        System.out.println("静态块1");
    }
    static{
        System.out.println("静态块2");
    }
}
```

16. 重写方法被调用时，一定是看对象的类型

- 重写：方法名和参数列表必须与超类完全相同
 - 派生类方法的访问权限大于或等于超类的
 - 派生类方法的返回值类型小于或等于超类的(若不理解，参考方法重写那天的内容)
 - 派生类方法抛出的异常小于或等于超类的(现在还没讲)
- 重载：方法名相同，参数列表不同(参数个数不同、参数类型不同)

17. 构造方法(constructor)：构造器、构建器、构造函数

- 作用：给实例变量赋初始值
- 若自己不写则默认有一个无参构造，若自己写了则不再默认提供
- 构造方法也属于类中成员，可以用4种访问修饰符来修饰

```
class Aoo{
    private Aoo(){ //构造方法也可以私有，若构造私有了，则别类的中不能new Aoo对象了
    }
}
```

18. 类是由class来定义的，接口是由interface来定义的，抽象类是由abstract修饰的

19. 包含抽象方法的类必须是抽象类/接口，派生类继承后必须重写所有抽象方法

重写接口中的方法时，必须加public权限

20. 代码顺序:

- 先package
- 而后import
- 然后class

21. ==对于基本类型而言，是判断值是否相同，==对于引用类型而言，是判断引用的地址是否相同

String中重写了Object的equals()来判断字符串内容是否相同

22. String字符串一旦定义好，对象内容不能再改变了，但是引用可以重新赋值(指向新的对象)，字符串字面量会存储在字符串常量池中，当下次内容相同的字符串被使用，将直接从常量池中获取

```
String s1 = "123abc";
String s2 = new String("123abc");
String s3 = "123"+"abc"; //编译器直接运算好并保存，相当于String s3 = "123abc"
String s4 = s1;
System.out.println(s1==s2); //false
System.out.println(s1==s3); //true
System.out.println(s1==s4); //true
System.out.println(s2==s4); //false
System.out.println(s2.equals(s4)); //true
```

一定要看StringDemo里面的那些代码、要看常量池图、面试题图

23. Integer类：是一个引用类型(包装类)，也能装整数

```
Integer i = 5; //自动装箱了，自动补全代码Integer i = Integer.valueOf(5);
int j = i; //自动拆箱了，自动补全代码int j = i.intValue();
```

补充：

```
public static void main(String[] args) {
    String s = "hello";
    testString(s); //String ss = s;
    System.out.println(s); //hello
}

public static void testString(String ss){
    ss = "hello world";
}
```

```
/*
    final修饰实例变量，必须声明同时初始化或在构造方法中初始化，否则编译错误
    final修饰局部变量，只要在使用之前初始化即可
*/
class Aoo{
    final int a = 5;
    final int b;
    Aoo(){
        b = 6;
    }
    void show(){
        final int c;
    }
}
```

