1. JDK ： Java Development kit （Java 开发工具包）
   1. Jar ：Java Runtime Environment（Java运行环境）
      1. Jvm ：Java Virtual Machine（Java 虚拟机）
2. Java运行过程：
   1. 编译（Javac）：将Java文件编译成class字节码文件
   2. 运行（Java）：Java虚拟机运行class文件
3. 变量：
   1. 局部变量：定义在方法或语句块内，**从属于方法**
   2. 成员变量（实例变量）：定义在类里面、方法外面**，从属于对象（如未初始化系统自动初始化 数字：0，0.0 布尔 ：false ；char ；\u0000 ；String :“”）(只能被对象调用)**
   3. 静态变量：定义在方法外面类里面，使用static修饰，**从属于类如未初始化系统自动初始化 数字：0，0.0 布尔 ：false ；char ；\u0000 ；String :“”）（可以被对象调用，还可以被类名调用。）**
4. **数据类型：**
   1. **基本数据类型：**
      1. **数字：**
         1. **整型：byte（1byte）、 short（2byte）、int（4字节，默认） 、long（8字节）**

**4.1.1.2浮点型：float（4byte） 、double（8byte）**

* + 1. **字符：char**
    2. **布尔：Boolean**
    3. **字符串：String**
  1. **引用类型：**
     1. **：数组**
     2. **：对象**
     3. **：接口**
  2. **类型转换**
  3. **运算符：**
     1. **算术运算符：+ - \* / %**
     2. **逻辑运算符：&& || ！ > < >= <=**

**按位与、或： & | <(左移一位 相当于\*2) >(右移一位 相当于/2)**

**Eg ( 3&4 == 011&100=000)**

* + 1. **三目运算**

**Return（a>b）?2:1**

* 1. **控制语句**
     1. **顺序结构**
     2. **选择结构**

**If**

**else if**

**else**

**多值选择，表达式：int short char**

**switch(表达式){**

**case :**

**break;**

**case:**

**break;**

**default:**

**}**

* + 1. **循环结构 ：**

**While{**

**。。。。。。。。。。。**

**}**

**for （int i=0 ；i< ; i++）{**

**}**

**For(Object obj : Object){**

**}**

**Do{**

**至少循环一次**

**}while();**

**Break；结束循环**

**Continue：结束本次循环**

1. **面向对象**
   1. **对象的进化史**
      1. **基本数据类型阶段：数据少 无数据管理时代**
      2. **数组：数据多了，将同类型的数据放到一起，若管理**
      3. **结构体：可以将不同类型的数据放到一起**
      4. **对象：数据复杂，行为复杂，不同类型的数据放一起**
   2. **对象和类：**
      1. **对象是具体的，类是抽象的。类也是模板 可以通过类来new 对象**
      2. **定义类：**

**public class （类名）**

**//属性**

**private +数据类型 + 变量名（增加set get方法）**

**//方法**

**构造方法：方法名和类名相同；无返回类型；通过new调用；没有构造器系统自动添加无参构造器**

**Super（调用被子类覆盖的父类方法） 构造方法第一句总是super**

**-------------------------------------------------------------------------**

**方法的重载：**

**同一类**

**同意方法名**

**参数的类型不同**

**参数的顺序不同**

**参数的个数不同**

* + 1. **关键字**

**this：**

**普通方法中：调用本方法内的对象**

**构造器中：正要初始化的对象**

**还可以代用其他的构造方法**

**super ：**

**static：被它修饰的方法和变量就变成了静态方法和静态属性，从属于类**

**package ：java包**

**import ：引入外部类**

**final ：**

**修饰变量：常量**

**修饰方法：不能被重写**

**修饰类不能被继承**

* 1. **面向对象三大特征：**
     1. **继承**

**类只能单继承**

* + 1. **封装**

**通过 privat 、protecte 、default、public 对属性或方法进行封装**

**高内聚、低耦合**

* + 1. **多态**

**三个必要条件：继承、对父类方法的重写、父类引用指向子类对象**

* + 1. **抽象类**
       1. **包含抽象方法**
       2. **抽象类不能被new**
       3. **抽象类可以包含：普通方法，成员变量、构造器（不能被调用）**
    2. **接口**
       1. **interface**
       2. **可以多继承**
       3. **implements可以实现多借口**
    3. **内存机制**
       1. **栈**

1. **存放局部变量**
2. **不可以被多线程共享**
3. **空间连续**
   * + 1. **堆**
4. **存放对象**
5. **可以被多线程共享**
6. **空间不连续**
   * + 1. **方法区**
7. **存放代码 静态变量 字符串长量等。**
8. **可以被多线程共享**
9. **空间不连续**
10. **异常处理**
    1. **try ………catch………finally**
    2. **抛出异常throws**
    3. **手动抛出异常 throw**
    4. **自定义异常**
11. **extends Exception或其子类**
12. **数组**

**7.1一维数组**

1. **数组长度不可变**
2. **所有元素类型可以不一样**
3. **通过下标进行索引**
4. **数组也是对象**

**7.2 多维数组**

**a) 数组的元素是数组**

1. **数组初始化的方式**
2. **默认初始化**
3. **容器**
   1. **Connection**
4. **List 有序（可重复）**

**1. Arraylist：底层用数组实现，线程不安全，效率高**

**2.Linkledist：底层用双向链表实现，线程不安全，效率高**

**3.Vector：底层用数组实现，线程安全，效率低**

1. **Set 无序（不可重复）**
2. **Hashset 用hashmap实现**
   1. **Map**
3. **Hashmap ：线程不安全，效率高**
4. **HashTable ： 线程安全，效率低**
   1. **Iterator**

**遍历容器中的元素**

* 1. **泛型**

1. **常用类**
   1. **包装类**
   2. **String**
2. **StringBuffer:可变字符序列，线程安全**
3. **StringBuilder：可变字符序列，线程不安全**
   1. **日期类**
4. **Date：使用毫秒来表示事件**
5. **Dateformat：日期格式化**
   1. **File**

**代表文件或目录**

* 1. **Math**
  2. **枚举**

1. **定义一组枚举使用枚举**
2. **IO流**
   1. **Inputstream**
3. **FileInputStream**
4. **ByteArrayInputStream**
5. **BufferedInputStream**
   1. **OutputStream**
6. **FileOutPutStream**
7. **ByteArrayOutputStream**
8. BufferedoutputStream
   1. **Reader**
   2. **Writer**
9. **多线程**
   1. **实现方法**
10. **Extends Threa**
11. **Implement Runnable**
    1. **线程的状态**
12. **New**
13. **就绪：start后**
14. **运行：**
15. **阻塞**

**1. 睡眠 :sleep，不会释放持有的对象锁**

**2. join 插队 会释放持有的对象锁**

**3.wait 等待另一个线程执行完成**

1. **终止**
2. **stop方法是过时的**
3. **使用Boolean变量终止**
4. **线程的优先级**
5. **1-10默认是5**
6. **优先级代表的是一种概率而不是先后**
7. **资源同步**
8. **Synchronized:修饰方法语句块（重量级）**
9. **Volatile ：**
10. **保证此变量对所有的线程的可见性**
11. **禁止指令重排序优化**
12. **死锁**

**不要锁套锁**

1. **网络编程**
2. **InetAddress: 封装了IP和DNS 无端口**
3. **InetSocketAddress：封装了IP和端口**
4. **URL**
5. **UDP：**
   1. **不面向连接，可能会丢数据，发送效率高，不安全**
   2. **Dategrampacket：数据打包**
   3. **dategramSocket：数据发送和接收**
6. **TCP：**
   1. **面向连接安全稳定，三次握手，效率较低，安全**
   2. **ServeSocket：发送**
   3. **Socket：接收**
7. **GUI**
   1. **组件**
8. **Component（普通组件、容器）**
   1. **布局管理器**
9. **flowLayout**
10. **BorderLayout**
11. **cardLayout**
12. **GridLayout**
    1. **事件**
13. **事件对象**
    * 1. **keyEvent**
      2. **MouseEvent**
      3. **。。。。。。。。。。。。**
14. **事件源对象**
15. **监听器**
    * + 1. **KeyListener 、KeyAdapter**
        2. **MouseListener、MouseMotionListener、MouseAdapter**
        3. **AddWindowsListener**
    1. **菜单制作**

**MeanBar**

**Mean**

**MeanItem**