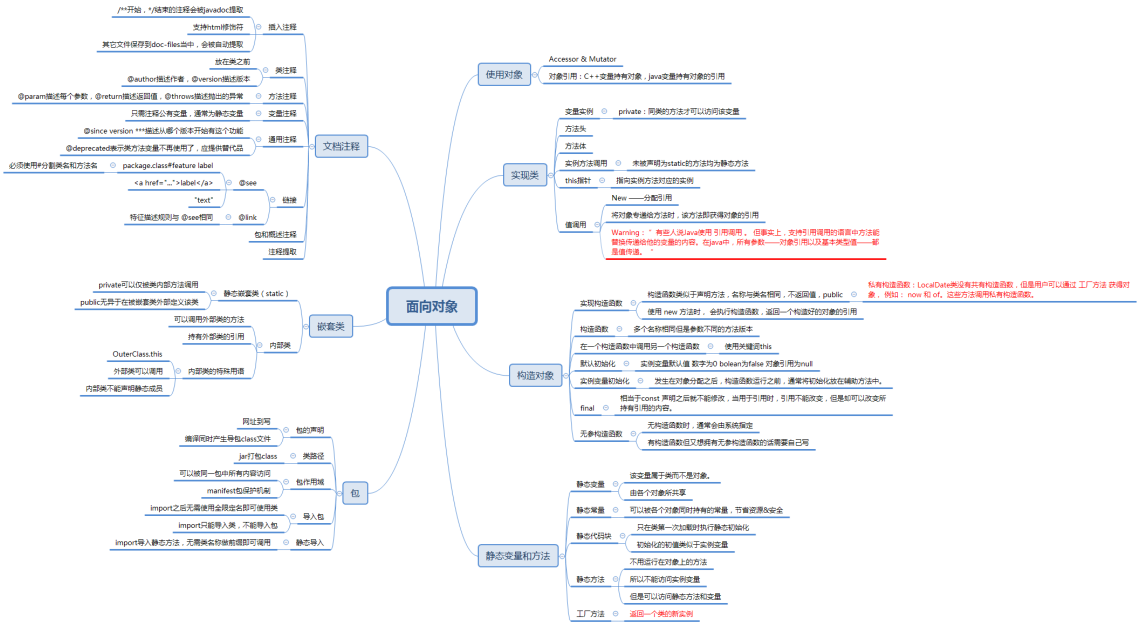
面向对象

面向对象 1
1. 使用对象 4
1.1. Accessor & Mutator 4
1.2. 对象引用：C++变量持有对象，java变量持有对象的引用 4
2. 实现类 4
2.1. 变量实例 4
2.1.1. private：同类的方法才可以访问该变量 4
2.2. 方法头 4
2.3. 方法体 4
2.4. 实例方法调用 4
2.4.1. 未被声明为static的方法均为静态方法 4
2.5. this指针 4
2.5.1. 指向实例方法对应的实例 5
2.6. 值调用 5
2.6.1. New ——分配引用 5
2.6.2. 将对象专递给方法时，该方法即获得对象的引用 5
2.6.3. Warning： ”有些人说Java使用 引用调用 。
但事实上，支持引用调用的语言中方法能替换传递给他的变量的内容。在java中，所有参数——对象引用以及基本类型值——都是值传递。“ 5
3. 构造对象 5
3.1. 实现构造函数 5
3.1.1. 构造函数类似于声明方法，名称与类名相同，不返回值，public 5
3.1.2. 使用 new 方法时， 会执行构造函数，返回一个构造好的对象的引用 5
3.2. 构造函数 5
3.2.1. 多个名称相同但是参数不同的方法版本 5
3.3. 在一个构造函数中调用另一个构造函数 5
3.3.1. 使用关键词this 5
3.4. 默认初始化 5
3.4.1. 实例变量默认值 数字为0 bolean为false 对象引用为null 5
3.5. 实例变量初始化 6
3.5.1.
发生在对象分配之后，构造函数运行之前，通常将初始化放在辅助方法中。 6
3.6. final 6

3.6.1. 相当于const
声明之后就不能修改，当用于引用时，引用不能改变，但是却可以改变所持有引用的内容。 6
3.7. 无参构造函数 6
3.7.1. 无构造函数时，通常会由系统指定 6
3.7.2. 有构造函数但又想拥有无参构造函数的话需要自己写 6
4. 静态变量和方法 6
4.1. 静态变量 6
4.1.1. 该变量属于类而不是对象。 6
4.1.2. 由各个对象所共享 6
4.2. 静态常量 6
4.2.1. 可以被各个对象同时持有的常量，节省资源&安全 6
4.3. 静态代码块 6
4.3.1. 只在类第一次加载时执行静态初始化 6
4.3.2. 初始化的初值类似于实例变量 6
4.4. 静态方法 6
4.4.1. 不用运行在对象上的方法 7
4.4.2. 所以不能访问实例变量 7
4.4.3. 但是可以访问静态方法和变量 7
4.5. 工厂方法 7
4.5.1. 返回一个类的新实例 7
5. 包 7
5.1. 包的声明 7
5.1.1. 网址到写 7
5.1.2. 编译同时产生导包class文件 7
5.2. 类路径 7
5.2.1. jar打包class 7
5.3. 包作用域 7
5.3.1. 可以被同一包中所有内容访问 7
5.3.2. manifest包保护机制 7
5.4. 导入包 7
5.4.1. import之后无需使用全限定名即可使用类 7
5.4.2. import只能导入类，不能导入包 7
5.5. 静态导入 7
5.5.1. import导入静态方法，无需类名称做前缀即可调用 7
6. 嵌套类 8
6.1. 静态嵌套类（static） 8

6.1.1. private可以仅被类内部方法调用 8
6.1.2. public无异于在被嵌套类外部定义该类 8
6.2. 内部类 8
6.2.1. 可以调用外部类的方法 8
6.2.2. 持有外部类的引用 8
6.2.3. 内部类的特殊用语 8
7. 文档注释 8
7.1. 插入注释 8
7.1.1. /**开始，*/结束的注释会被javadoc提取 8
7.1.2. 支持html修饰符 8
7.1.3. 其它文件保存到doc-files当中，会被自动提取 8
7.2. 类注释 8
7.2.1. 放在类之前 8
7.2.2. @author描述作者，@version描述版本 9
7.3. 方法注释 9
7.3.1. @param描述每个参数，@return描述返回值，@throws描述抛出的异常 9
7.4. 变量注释 9
7.4.1. 只需注释公有变量，通常为静态变量 9
7.5. 通用注释 9
7.5.1. @since version ***描述从哪个版本开始有这个功能 9
7.5.2. @deprecated表示类方法变量不再使用了，应提供替代品 9
7.6. 链接 9
7.6.1. @see 9
7.6.2. @link 9
7.7. 包和概述注释 9
7.8. 注释提取 9



1. 使用对象

1.1. Accessor & Mutator

1.2. 对象引用：C++变量持有对象，java变量持有对象的引用

2. 实现类

2.1. 变量实例

2.1.1. private：同类的方法才可以访问该变量

2.2. 方法头

2.3. 方法体

2.4. 实例方法调用

2.4.1. 未被声明为static的方法均为静态方法

2.5. this指针

2.5.1. 指向实例方法对应的实例

2.6. 值调用

2.6.1. New ——分配引用

2.6.2. 将对象专递给方法时，该方法即获得对象的引用

2.6.3. Warning： ”有些人说**Java**使用 引用调用 。
但事实上，支持引用调用的语言中方法能替换传递给他的变量的内容。在**java** 中，所有参数——对象引用以及基本类型值——都是值传递。“

3. 构造对象

3.1. 实现构造函数

3.1.1. 构造函数类似于声明方法，名称与类名相同，不返回值，**public**

私有构造函数：**LocalDate**类没有共有构造函数，但是用户可以通过 工厂方法获得对象， 例如： **now** 和 **of**。这些方法调用私有构造函数。

3.1.2. 使用 **new** 方法时， 会执行构造函数，返回一个构造好的对象的引用

3.2. 构造函数

3.2.1. 多个名称相同但是参数不同的方法版本

3.3. 在一个构造函数中调用另一个构造函数

3.3.1. 使用关键词**this**

3.4. 默认初始化

3.4.1. 实例变量默认值 数字为**0 bolean**为**false** 对象引用为**null**

3.5. 实例变量初始化

3.5.1. 发生在对象分配之后，构造函数运行之前，通常将初始化放在辅助方法中。

3.6. final

3.6.1. 相当于const

声明之后就不能修改，当用于引用时，引用不能改变，但是却可以改变所持有引用的内容。

3.7. 无参构造函数

3.7.1. 无构造函数时，通常会由系统指定

3.7.2. 有构造函数但又想拥有无参构造函数的话需要自己写

4. 静态变量和方法

4.1. 静态变量

4.1.1. 该变量属于类而不是对象。

4.1.2. 由各个对象所共享

4.2. 静态常量

4.2.1. 可以被各个对象同时持有的常量，节省资源&安全

4.3. 静态代码块

4.3.1. 只在类第一次加载时执行静态初始化

4.3.2. 初始化的初值类似于实例变量

4.4. 静态方法

4.4.1. 不用运行在对象上的方法

4.4.2. 所以不能访问实例变量

4.4.3. 但是可以访问静态方法和变量

4.5. 工厂方法

4.5.1. 返回一个类的新实例

5. 包

5.1. 包的声明

5.1.1. 网址到写

5.1.2. 编译同时产生导包class文件

5.2. 类路径

5.2.1. jar打包class

5.3. 包作用域

5.3.1. 可以被同一包中所有内容访问

5.3.2. manifest包保护机制

5.4. 导入包

5.4.1. import之后无需使用全限定名即可使用类

5.4.2. import只能导入类，不能导入包

5.5. 静态导入

5.5.1. import导入静态方法，无需类名称做前缀即可调用

6. 嵌套类

6.1. 静态嵌套类（static）

6.1.1. private可以仅被类内部方法调用

6.1.2. public无异于在被嵌套类外部定义该类

6.2. 内部类

6.2.1. 可以调用外部类的方法

6.2.2. 持有外部类的引用

6.2.3. 内部类的特殊用语

OuterClass.this

外部类可以调用

内部类不能声明静态成员

7. 文档注释

7.1. 插入注释

7.1.1. /**开始，*/结束的注释会被javadoc提取

7.1.2. 支持html修饰符

7.1.3. 其它文件保存到doc-files当中，会被自动提取

7.2. 类注释

7.2.1. 放在类之前

7.2.2. @author描述作者， @version描述版本

7.3. 方法注释

7.3.1. @param描述每个参数， @return描述返回值， @throws描述抛出的异常

7.4. 变量注释

7.4.1. 只需注释公有变量，通常为静态变量

7.5. 通用注释

**7.5.1. @since version ***描述从哪个版本开始有这个功能

7.5.2. @deprecated表示类方法变量不再使用了，应提供替代品

7.6. 链接

7.6.1. @see

package.class#feature label

必须使用#分割类名和方法名

label

"text"

7.6.2. @link

特征描述规则与 **@see**相同

7.7. 包和概述注释

7.8. 注释提取