

1. 封装, 继承, 多态
2. **PackageUser类仅包内可见因此, 其他包, 即使是子包中也无法使用**
3. **Package级允许源文件名与类型名称不同Public级别则不允许**
4. package级类允许文件名与类名不同, 但是禁止使用
5. 当类显式声明了有参构造函数, 编译器将不再自动创建无参构造函数
6. 静态导入语句提供了一种导入要使用的常量和静态方法的方法, 以便无需为其类的名称添加前缀
7. 垃圾回收器会在合适的时间, 自动的/周期性的释放没有引用对象占用的内存
8. ☐静态代码块中的变量是局部变量
9. 一个类中可以有多静态代码块
10. 静态内部类, **不持有外部类的引用, 静态内部类与外部类完全是独立的仅在代码层面嵌套**
11. **将方法声明为default方法则支持在接口中直接实现该方法**
12. 声明字面量字符串变量时, JVM先在字符串常量池中查找是否有相同字面量的字符串对象。有则返回引用, 没有则创建字符串对象并置于池中, String nString = "110";
13. Collection接口。用于描述, 最具通用性的集合。因此, 也包含了最具通用性的集合操作方法
14. Collection接口继承自Iterable接口。即, 所有集合类型均支持foreach循环语句
15. Lambda表达式语法
 - 箭头, 函数参数列表与表达式/函数主体的分隔符
 - Lambda表达式可包含0或多个参数
 - 参数列表, 当参数为空时, 需声明空括号; 当只有一个参数时, 可省略括号; 参数类型可省略, 编译器自动完成类型推导
16. 函数式接口, 能且只能包含1个抽象方法的接口
17. 只要类被加载, 即执行static相关操作, **第二次调用getS()方法, 直接返回static变量的值/引用, 而不会再次执行create()方法**
18. 只要类被加载, 无论是由于被实例化, 还是被调用static成员(变量/常量/方法), 均按声明的先后顺序, 初始化static成员或执行static代码块, 后实例化类。**调用父类A构造函数, 必先调用父类A构造函数**
19. **锁定基本类型**的包装对象无效!
20. 当重写了带参数的构造方法, 系统默认的不带参数的构造方法不存在
21. **方法重载属于编译时多态, 可以在编译阶段确定执行重载方法中的哪一个**
22. 抽象类不能实例化, 即不能通过构造方法直接创建对象。
23. 接口可以只定义常量, 而没有任何方法。接口类型数组存在
24. 接口中所有抽象/默认/静态方法都是隐式public, 故重写的子类方法只能public, 因为不能缩小
25. **内部类可以直接访问外部类的成员, 静态内部类不可以直接访问外部类的成员, 外部类可通过创建内部类的实例来访问内部类的成员**
26. Java集合框架的顶级接口Collection, ☐Iterable接口不属于Java集合框架
27. Iterator是迭代接口, 不是实现类
28. LinkedList不是集合类型, 而是List集合类型的一种基于双向链表的实现
29. Map不是集合, 没有索引, 不支持foreach语句遍历, 支持foreach方法

- 30. 集合是存储元素对象的容器，而Stream(集合流)，并不是存储元素的数据结构，而是操作集合元素的管道
- 31. 当线程调用start()后，其所处状态为就绪状态
- 32. Java中的资源对象与创建的普通对象不同，不会在失去引用后自动销毁
- 33. int read() throws IOException，如果到达流末没有可读字节，返回-1，不会抛异常
- 34. OutputStream > FileOutputStream