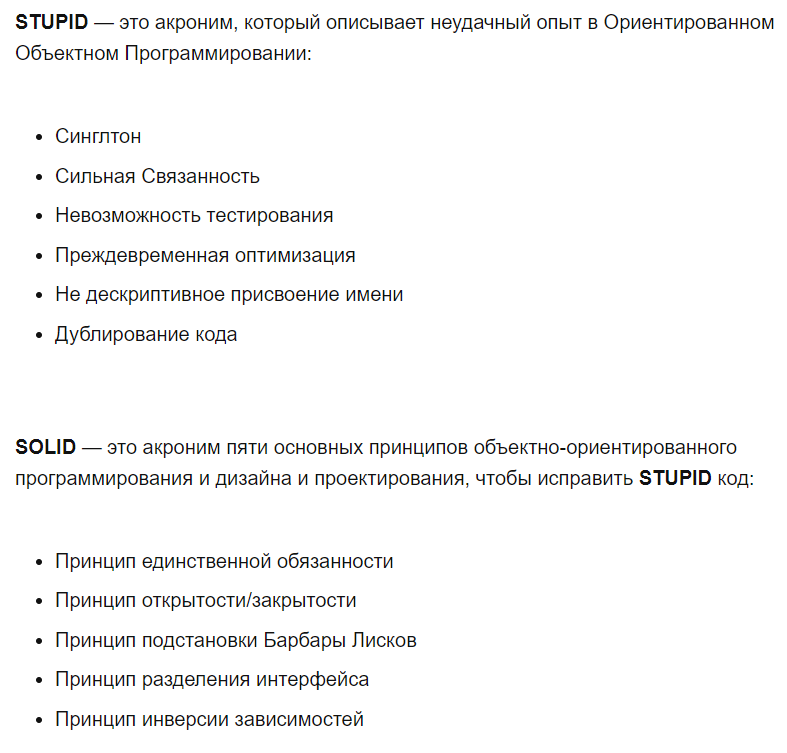
**Вопросы к защите лабораторной работы:**

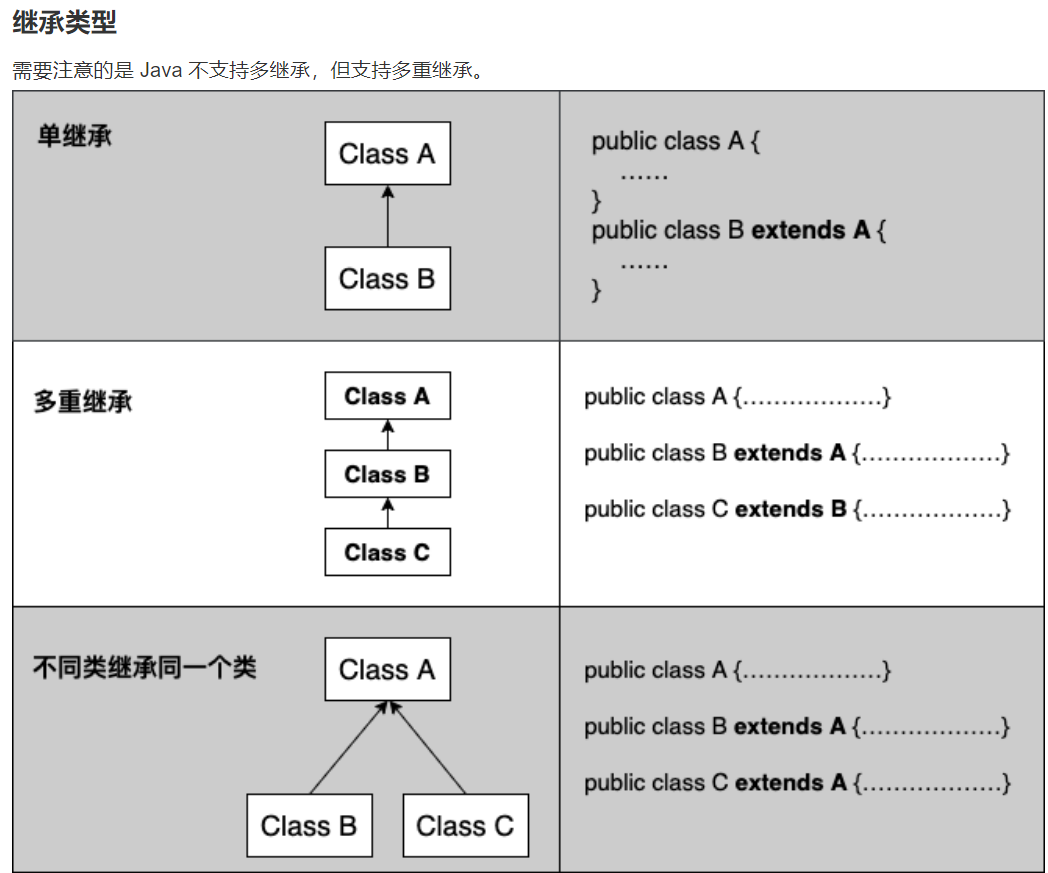
1. Принципы объектно-ориентированного программирования SOLID и STUPID.



1. Класс Object. Реализация его методов по умолчанию. 类Object。其方法的默认实现。

Object – базовый класс для всех классов.

对象是所有类的基类。

1. protected Object clone() //创建并返回此对象的一个副本。
2. boolean equals(Object obj) //指示某个其他对象是否与此对象“相等”。
3. protected void finalize() //当垃圾回收器确定不存在对该对象的更多引用时，由对象的垃圾回收器调用此方法。
4. Class<? extends Object> getClass() //返回一个对象的运行时类。
5. int hashCode() //返回该对象的哈希码值。
6. void notify() //唤醒在此对象监视器上等待的单个线程。
7. void notifyAll() //唤醒在此对象监视器上等待的所有线程。
8. String toString() //返回该对象的字符串表示。
9. void wait() //导致当前的线程等待，直到其他线程调用此对象的 notify() 方法或 notifyAll() 方法。
10. void wait(long timeout) //导致当前的线程等待，直到其他线程调用此对象的 notify() 方法或notifyAll() 方法，或者超过指定的时间量。
11. void wait(long timeout, int nanos) //导致当前的线程等待，直到其他线程调用此对象的notify() 方法或 notifyAll() 方法，或者其他某个线程中断当前线程，或者已超过某个实际时间量。
12. Особенности реализации наследования в Java. Простое и множественное наследование.(Java中继承实现的特点。简单和多重继承。)
13. Понятие абстрактного класса. Модификатор abstract.

抽象类概念。修饰符abstract。

1. Абстрактные классы в Java представляют собой отношения наследования, где класс может наследоваться только от одного абстрактного класса, в то время как класс может реализовывать несколько интерфейсов.
2. Используйте абстрактный класс для определения абстрактного класса.
3. Понятие интерфейса. Реализация интерфейсов в Java, методы по умолчанию. Отличия от абстрактных классов.

https://www.runoob.com/java/java-interfaces.html

1. Перечисляемый тип данных (enum) в Java. Особенности реализации и использования.

Использование перечислений во внутренних классах

Итерация над элементами перечисления

Использование классов перечисления в переключателе

1. Методы и поля с модификаторами static и final.
2. Перегрузка и переопределение методов. Коварианты возвращаемых типов данных.
3. Элементы функционального программирования в синтаксисе Java. Функциональные интерфейсы, лямбда-выражения. Ссылки на методы.

**为实验室工作辩护的问题：**

1. 面向对象编程的原理 SOLID 和 STUPID。



1. 类Object。其方法的默认实现。

Object – базовый класс для всех классов.

protected Object clone() //创建并返回此对象的一个副本。

boolean equals(Object obj) //指示某个其他对象是否与此对象“相等”。

protected void finalize() //当垃圾回收器确定不存在对该对象的更多引用时，由对象的垃圾回收器调用此方法。

Class<? extends Object> getClass() //返回一个对象的运行时类。

int hashCode() //返回该对象的哈希码值。

void notify() //唤醒在此对象监视器上等待的单个线程。

void notifyAll() //唤醒在此对象监视器上等待的所有线程。

String toString() //返回该对象的字符串表示。

void wait() //导致当前的线程等待，直到其他线程调用此对象的 notify() 方法或 notifyAll() 方法。

void wait(long timeout) //导致当前的线程等待，直到其他线程调用此对象的 notify() 方法或notifyAll() 方法，或者超过指定的时间量。

void wait(long timeout, int nanos) //导致当前的线程等待，直到其他线程调用此对象的notify() 方法或 notifyAll() 方法，或者其他某个线程中断当前线程，或者已超过某个实际时间量。

1. Java中继承实现的特点。简单和多重继承。
2. 抽象类概念。修饰符abstract。

在 Java 中抽象类表示的是一种继承关系，一个类只能继承一个抽象类，而一个类却可以实现多个接口。

使用 abstract class 来定义抽象类。

Abstract 关键字同样可以用来声明抽象方法，抽象方法只包含一个方法名，而没有方法体。

1. 接口概念。Java 中接口的实现，默认方法。与抽象类的区别。
2. 接口概念：

接口是一个抽象类型，是抽象方法的集合

1. 接口的实现：

当类实现接口的时候，类要实现接口中所有的方法。否则，类必须声明为抽象的类。

1. 接口默认方法：

类使用implements关键字实现接口。在类声明中，Implements关键字放在class声明后面

1. 与抽象类的区别:
2. 抽象类中的方法可以有方法体
3. 抽象类中的成员变量可以是各种类型的，而接口中的成员变量只能是 public static final 类型的。
4. 接口中不能含有静态代码块以及静态方法(用 static 修饰的方法)，而抽象类是可以有静态代码块和静态方法
5. 一个类只能继承一个抽象类，而一个类却可以实现多个接口。
6. Java 中的枚举数据类型（enum）。实现和使用的特点。
7. 带有 "static"和 "final"修饰语的方法和字段。2022/1/13
8. 方法的重载和重写。返回数据类型的协变量。
9. Java 语法中的函数式编程元素。函数式接口，lambda 表达式。方法参考。