

作业二：

### 1、简述重载和重写的区别。

区别：

(1) 重载是指不同的函数使用相同的函数名，但是函数的参数个数或类型不同。调用的时候根据函数的参数来区别函数。而重写是指派生类重新对基类的中的虚函数重新实现，即函数名和参数一样，只是函数的实现体不一样。

(2) 重载和重写的规则也不一样，重载：必须具有不同的参数列表，可以有不同的访问的修饰符，可以抛出不同的异常。重写：参数列表必须完全与被重写的方法相同，否则不能视为重写而是重载。返回类型必须是一直与被重写的方法的返回类型相同，否则不能称其为重写。访问修饰符限制一定大于被重写方法的访问修饰符。重写方法一定不能抛出心得异常或者比被重写方法申明更加宽泛的检查异常。

(3) 重写是子类与父类之间的关系，属于垂直关系

### 2、简述 final 关键字修饰变量、方法和类的不同作用。

修饰类：该类不能被继承，该类的方法不会被覆盖，在设计时，此类不需要有子类，不必要被扩展，类的实现细节不需要子类，不必要被扩展，类的实现细节不允许被改变。

修饰方法：该方法可以被继承，但是不能被覆盖，可以防止继承它的子类修改该方法的意义和实现更为高效提高执行效率。

修饰变量：用 final 修饰后变为常量，可以先申明不给初值，但是使用前必须被初始化，一旦被赋值，将不能被改变。

修饰参数：用 final 修饰参数，可以读取参数，但是不能对其做其修改。

### 3、Java 中子类重新覆盖（重写）父类的方法的两个原则是什么？你理解这两个原则？

原则：1.子类的方法不能比父类权限小 2.不能比父类抛出的异常多

理解：

要遵循开闭原则，如果权限比父类少，父类可以调用任意方法，但是子类却不能，这样就不用创建子类了，只需要一个父类解决一切问题，通过使用父类调用方法一切完美，但是这样

### 4、简述抽象类和接口的相同点、不同点。

相同点：

(1) 不能直接实例化，如果要实例化，抽象类变量必须指向实现所有抽象方法的子类对象，接口变量必须指向实现所有接口方法的类对象。

(2) 抽象类里的抽象方法必须全部被子类所实现，子类不能全部实现父类抽象方法，那么该子类还只能是抽象类。一个类实现接口的时候，如不能全部实现接口方法，那么该类也只能为抽象类。

不同点：

(1) 抽象类可以有默认的方法实现；接口，Java8 规定可以为接口提供默认实现的方法并且不用强制子类来重写

(2) 抽象类可以有构造器，接口不能有构造器

(3) 抽象方法可有 public.protected 和 default、abstract 修饰符。不能 private static.synchronize、 native 修饰；接口属性默认修饰符是 public static final

(必须赋初值)，方法默认修饰符是 `public abstract`. Java8 开始出现静态方法，多加 `static` 关键字。

(4) 抽象类可以继承一个类 and 实现多个接口；接口只可以继承一个或多个接口。

5、分别使用 `String`，`StringBuffer` 完成从“1”，“2”，...，“1000”的字符串连接，然后记录连接操作的耗时，并说明你所观察到的现象。

```
package ex2;
```

```
public class ex22 {
```

```
    public static void main(String[] args) {
```

```
        long a1=System.currentTimeMillis();
```

```
        String s = "";
```

```
        for(int i = 0;i <= 10000;i++) {
```

```
            s = s + i;
```

```
        }
```

```
        System.out.println("String:"+(System.currentTimeMillis()-a1));
```

```
        StringBuffer sb = new StringBuffer("");
```

```
        a1=System.currentTimeMillis();
```

```
        for(int i = 0;i <= 10000;i++) {
```

```
            sb.append(i);
```

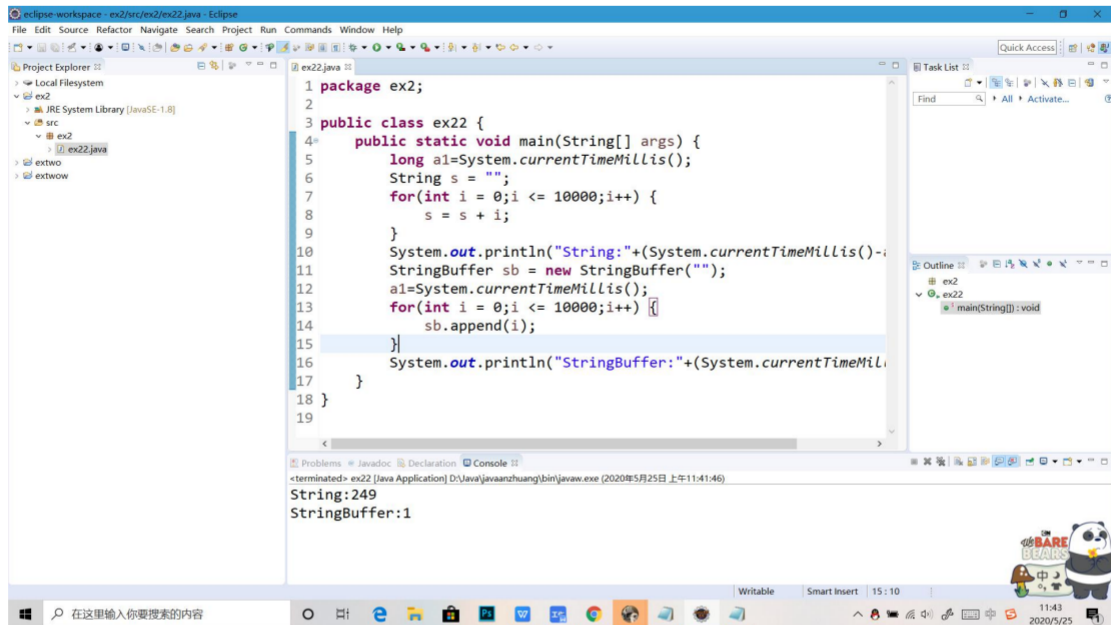
```
        }
```

```
        System.out.println("StringBuffer:"+(System.c
```

```
urrentTimeMillis()-a1));
```

```
}
```

```
}
```



现象：StringBuffer 插入字符串的速度明显快于 String 字符串插入的速度