1. 面向对象的理解并举例？

卖东西:

面向过程：到商店-->找商户-->找商品-->买商品

面向过程：找老妈-->老妈买商品

2. 类与对象之间的关系？

类是一类具有相同特性事物的抽象集合，是对象的模版与蓝图。

3. 如何对类进行分析，如果创建自定义类对象，并如何指挥对象做事情？

类具有公有的属性及公共的方法，如果创建自定义对象，可以创建该类公共的方法

4. 对象的内存分布图？

创建对象时首先会在方法区加载该类class文件，解析该类具有的公有属性及公共行为，之后在栈内存中创建该类的引用类型变量，同时会在堆内存中创建该类的一个对象，栈内存中的引用类型变量指向堆内存中的对象。

5. 成员变量和局部变量的区别？

1.定义位置不同

成员变量：类之内，方法之外

局部变量：类之内，方法之内

2.作用范围不同

成员变量:在该类内部任何位置可以随意引用

局部变量:只在其方法定义的有效作用域内有效，一旦超出其作用域立即失效

3.生命周期不同

成员变量：随着对象的产生而产生，随着对象的消亡而消亡

局部变量：只在其方法定义的有效作用域内有效，一旦超出其作用域立即失效

4.初始值不同

成员变量：具有初始值，int 0，Boolean false

局部变量：没有初始值，必须初始化赋值才能使用

5.储存位置不同

成员变量:存储在堆内存中(对象存在的空间)

局部变量:存储在栈内存中

6. 私有的使用。

private关键字的使用：一般情况下对于一个类中的公有属性来讲，只要是实体类就应该进行private封装，并按需求创建公有的访问器及修改器。而对于该类的方法来说，如果只是给该类内部进行使用可以使用private关键字进行封装。对于一个类来说，如果只是为同一个包中使用，可以使用private进行封装。

7. 构造函数和一般函数的区别？

1.构造函数没有返回值类型

2.构造函数的函数名与该类的类名相同

3.构造函数不是手动调用的，是创建对象值JVM主动调用的，每创建一个对象就会调用一次构造函数

4.构造函数可以通过重载的方式多个出现

5.当一个类没有创建造函数是编译器会自动给其添加一个无参的构造函数(继承自object类)

6.当一个类显现创建一个构造方法时编译器不会个其添加无参的构造方法。

8. 构造函数什么时候用？

创建对象时调用该类的构造方法，对成员变量进行初始化赋值。

9. 构造代码块的作用？

对该类的对象进行统一的初始化赋值

10. this关键字的特点和使用以及应用场景？

this关键字指向本类对象

1.this关键字引用本类的构造方法时必须出现在第一句。

2.使用this关键字引用本类构造方法时要避免各自调用

3.this关键字代表本类对象，只能在非静态成员中使用

4.如果在一个函数中访问一个成员变量，而且没有存在同名的局部变量 时，那么 java编译器会默认在变量的前面加上 this关键字的。

5.作业

5．作业

5.1.静态和非静态的区别。说一下内存

数量上的区别

存储位置上的区别

生命周期的区别

作用域与作用上的区别

访问方式上的区别

调用方式：

静态成员 ：可以通过类名.静态成员及对象.静态成员访问

非静态成员 ：可以通过对象.非静态成员访问

访问限制：

静态 ：静态成员函数不能够直接使用非静态成员变量，能够直接使用静态成员变量

非静态 :非静态成员函数能够直接使用非静态成员及静态成员。

this,super关键字的使用：

静态 ：静态函数不能够使用this与super关键字

非静态 ：非静态函数能够调用thi与super关键字

一个类文件加载到内存中，JVM会对该类文件进行解析,并将分析出该类文件具有哪些成员及哪些静态数据。 一个方法在运行时JVM会在栈内存中分配对应的内存空间

1.当前内存由栈内存、堆内存、方法区(java总共有5片内存)

2.类加载时先加载class文件进入共享区

其中该类的main方法加载进方法栈内存

.class 加载进方法区

堆内存开辟空间，实例变量进行默认初始化，显示初始化。

内存地址传给变量 p1，栈和堆建立连接

3.创建该类对象

1：堆内存开辟空间，实例变量进行默认初始化，显示初始化。

2：内存地址传给变量 p2，栈和堆建立连接

4.建立多个该类对象

每一个该类对象都可以访问Static中的静态变量

5.2. 成员变量和静态变量的区别？

访问方式：

静态成员变量 ：可以通过类名.静态成员及对象.静态成员访问

非静态成员变量 ：可以通过对象.非静态成员访问

数量差距：

静态成员变量 ：只会在方法区中存在一份，同类对象共同引用相同的静态成员变量

非静态成员变量 ：每个对象中都会存在一份自己的非静态成员变量

储存位置:

静态成员变量 ：在方法区中有且仅存在一份

非静态成员变量 ：在堆内存中的每一个对象空间中存在一份

作用区别：

静态成员变量 ：静态成员变量的作用是提供一个共享数据给所有对象使用的。

非静态成员变量 ：非静态成员变量的作用是用于描述一类事物的属性。

生命周期：

静态成员变量 ：随着该类的加载而产生，随着类的消失而消失

非静态成员变量 ：随着对象的产生而产生，随着对象的消失而消失

5.3. 静态的特点以及注意事项？

5.4. 什么时候使用静态？

对于成员变量来说，如果该成员变量对于该类的所有对象来说只需要存在一份即可时(共享给该类使用对象共同使用)可以使用static进行修饰

对于成员函数来说，如果该成员函数没有直接使用该类的非静态成员变量时可以使用static进行修饰。

5.5. 继承的好处？

1.减少重复代码

2.描述类与类之间的相互关系

5.6. java改良多继承的原因？

如果一个类能够继承多个父类会造成继承关系复杂，当一个类能够继承多个父类时构造函数的调用难以确定。

5.7. 当使用一个已存在的继承体系时，该如何更快应用

直接使用子类，如果子类的方法不能够满足需求，可以对其非静态函数进行重写

5.8. 什么时候用继承？

当类与类之间真实存在继承关系时使用继承

5.9. super 和 this 的特点？

super This

super指向父类的引用空间

super引用父类构造函数时需要出现在构造函数的第一句。

super引用父类构造函数时不能够与this关键字一起使用。

this与super关键字都不能与static共同使用 this代表该类对象

this关键字调用本类构造函数时需要出现在构造函数的第一句

this关键字调用本类构造函数时不能够与super共同使用

this关键调用本类构造函数时要避免重复调用

5.10. 覆盖的特点，何时应用，注意事项？

当父类的方法不能够满足需求是，需要在子类中重写父类的方法

前提：类与类之间存在继承或者实现的关系

1.函数名一致

2.形参列表一致

3.子类方法的返回值类型小与或者等于父类方法的返回值

4.子类方法的访问权限修饰符要大于或者等于父类方法的访问权限修饰符

5.子类的异常报错类型要小与父类方法的异常报错类型

5.11. 子类的实例化过程？为什么是这样的实例化过程？

1.子类class对象加载到方法区

2.栈内存创建子类的引用类型变量

3.堆内存创建子类对象的内存空间

4.父类class对象加载到方法区

5.解析父类有哪些成员变量及公共静态方法

6.子类使用super方法引用父类的空间

5.12. super语句，和 this语句为什么不能同时存在，super为什么要定义在第一行？

因为this与super都要出现在构造函数的第一句，这要就相互矛盾了。

5. 作业

5.1.为什么要将一个类定义成内部类？

内部类能够减少代码量，简化书写

5.2.匿名内部类的使用和细节(面试题)

public void 方法(){

class Inner(){

内部类属性;

内部类方法;

}.内部类方法;

};

public void 方法(){

Father f =new Father(){

内部类属性;

内部类方法一;

内部类方法二;

};

f.内部类方法一;

f.内部类方法二;

};

public void 方法(){

new Father(){

内部类属性;

重写父类方法;

内部类方法(return this);

}.内部类方法.重写父类方法;

};

5.3.异常的思想和体系特点？

|throwable

|error(JVM或者硬件导致的错误，不能够进行处理只能够尽量避免)

|exception(可以通过java代码进行处理)

|运行时异常

|编译时异常

5.4.throws和 throw的如何使用？

throws在方法声明使用，用于声明抛出异常

throw在方法中使用，用于抛出具体的异常

5.5.什么时候 try什么时候 throws？

需要捕获异常，对异常进行处理时使用try-catch关键字对异常进行捕获处理；如果不处理异常，需要将异常抛出是可以使用throws抛出异常。

5.6.编译时被检测异常和运行时异常的区别？

编译时异常，编译不能同过，体现在程序中就是由红色感叹号。

运行时异常，编译可以通过，程序没有红色感叹号，但是在运行时会有错误发生。

5.7.异常的所有细节？

5.8.finally的应用？

try-catch捕获异常的最后步骤，通常用于关闭资源。

5.9.包的作用，名称空间的定义和理解？

1 把功能相似或相关的类或接口组织在同一个包中，方便类的查找和使用。

2 如同文件夹一样，包也采用了树形目录的存储方式。同一个包中的类名字是不同的，不同的包中的类的名字是可以相同的，当同时调用两个不同包中相同类名的类时，应该加上包名加以区别。因此，包可以避免名字冲突。

3 包也限定了访问权限，拥有包访问权限的类才能访问某个包中的类。

在一个.java文件中可以一个public类和多个非public类，如果要将这些类组织在一个包当中，则在.java文件中除注释以外的第一行使用关键字package即可实现。当需要调用此包中的类时，就可以使用关键字import进行导入。在定义包的时候，应该注意几点:

1 为了尽量使包名保持唯一性，包名通常采用小写、按倒写互联网址的形式进行定义。

2 在进行命名包时，应该避免使用与系统发生冲突的名字。

5.10.jar包的基本使用

只要将类和包都存储到jar中，方便于使用。只要将jar配置到classpath路径下

6.作业

6.1.抽象类的特点，以及细节？

1. 如果一个方法没有方法体，那么该方法必须使用 abstract 修饰。

2. 如果一个类有抽象的方法，那么该类也必须使用 abstract 修饰。

3. 非抽象类继承抽象类的时候，必须要把抽象类中的所有抽象方法全部实现。

4. 抽象类可以定义抽象方法以及非抽象方法。

5. 抽象类是可以不存在抽象方法的。

6. 抽象类不能创建对象。为什么？

疑问： 为什么抽象类不能创建对象呢？因为抽象类一旦创建了对象就可以使用对象去调用方法了，一旦调用了抽象方法就没有任何的意义了。java是不支持废话的。

7. 抽象类是存在构造方法的。

疑问：既然抽象类不能创建对象，那么存在构造方法的意义在那？抽象类的构造方法是留给子类调用初始化从父类继续下去的成员变量的。

8．抽象类不能创建对象，那么抽象类中是否有构造函数？抽象类中一定有构造函数。主要为了初始化抽象类中的属性。通常由子类实现。

9．final 和 abstract 是否可以同时修饰一个类？一定不能同时修饰。

原因：final修饰的对象成员不能够被重写，重新赋值、继承，abstract修饰的对象，子类一定要实现父类的abstract修饰的方法。

1. abstract 不能与 private 配合使用。

private修饰的父类方法不能够被子类继承，只能够在本类中使用

2. abstract 不能与 final 关键字配合使用。

final修饰的类不能够被继承

final:它的作用修饰类代表不可以继承 修饰方法不可重写

abstract 修饰类就是用来被继承的，修饰方法就是用来被重写的。

3. abstract 不能与 static 关键字配合使用。

6.2.接口的表现形式的特点。

接口的定义：

interface 接口{

public abstract void 方法();

}

接口的实现

class 实现类 implement 接口{

}

1.接口类没有构造方法

2.接口类并创建对象

3.接口类实现时需要使用implement关键字，一个类可以实现多个接口

4.接口中的常量属性默认是public static final，即所有常量都是静态常量，定义时必须进行初始化赋值

5.有抽象函数的类可以是抽象类也可以是抽象方法，抽象类不一定有抽象方法。

6.由于接口中的方法默认都是抽象的，所以不能被实例化。

static修饰的方法可以被子类继承，但是在继承时进行了隐藏，不能够被子类重新。

abstract修饰的方法需要被被子类重写以实现该方法。

7.对于接口而言，可以使用子类来实现接口中未被实现的功能函数。

8.如果实现类中要访问接口中的成员，不能使用super关键字。因为两者之间没有显示的继承关系，况且接口中的成员成员属性是静态的。可以使用接口名直接访问。

6.3.接口的思想特点，要举例。

车具有空调的功能

6.4.多实现和多继承的区别？

多实现：实现多个接口

多继承：继承多个父类

6.5.抽象类和接口的区别？

抽象类是一类事物的抽象，是从类中抽取出来的

接口是功能方法的抽象

6.6.多态的体现，前提，好处，弊端。

体现：父类的引用类型变量指向子类对象，父类引用也可以接受自己的子类对象

前提：必须存在继承或者实现关系

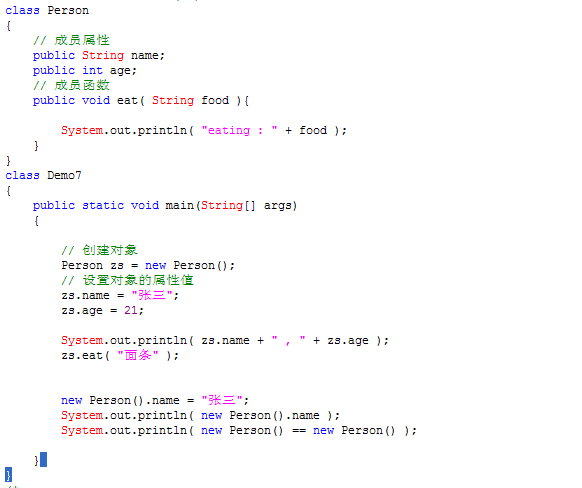
好处：提高扩展性，方法可以接收及返回更多的对象类型

1. 应用程序不必为每一个派生类编写功能调用，只需要对抽象基类进行处理即可。大大提高程序的可复用性。//继承

2. 派生类的功能可以被基类的方法或引用变量所调用，这叫向后兼容，可以提高可扩充性和可维护性。 //多态的真正作用

弊端：父类的引用类型变量不能够使用子类特有的方法，如果需要使用需要进行数据类型转化

1. 请问输出的name属性是什么? new Perosn == new Person()相等吗？



}

1. 面向对象的理解并举例？

卖东西:

面向过程：到商店-->找商户-->找商品-->买商品

面向过程：找老妈-->老妈买商品

2. 类与对象之间的关系？

类是一类具有相同特性事物的抽象集合，是对象的模版与蓝图。

3. 如何对类进行分析，如果创建自定义类对象，并如何指挥对象做事情？

类具有公有的属性及公共的方法，如果创建自定义对象，可以创建该类公共的方法

4. 对象的内存分布图？

创建对象时首先会在方法区加载该类class文件，解析该类具有的公有属性及公共行为，之后在栈内存中创建该类的引用类型变量，同时会在堆内存中创建该类的一个对象，栈内存中的引用类型变量指向堆内存中的对象。

5. 成员变量和局部变量的区别？

1.定义位置不同

成员变量：类之内，方法之外

局部变量：类之内，方法之内

2.作用范围不同

成员变量:在该类内部任何位置可以随意引用

局部变量:只在其方法定义的有效作用域内有效，一旦超出其作用域立即失效

3.生命周期不同

成员变量：随着对象的产生而产生，随着对象的消亡而消亡

局部变量：只在其方法定义的有效作用域内有效，一旦超出其作用域立即失效

4.初始值不同

成员变量：具有初始值，int 0，Boolean false

局部变量：没有初始值，必须初始化赋值才能使用

5.储存位置不同

成员变量:存储在堆内存中(对象存在的空间)

局部变量:存储在栈内存中

6. 私有的使用。

private关键字的使用：一般情况下对于一个类中的公有属性来讲，只要是实体类就应该进行private封装，并按需求创建公有的访问器及修改器。而对于该类的方法来说，如果只是给该类内部进行使用可以使用private关键字进行封装。对于一个类来说，如果只是为同一个包中使用，可以使用private进行封装。

7. 构造函数和一般函数的区别？

1.构造函数没有返回值类型

2.构造函数的函数名与该类的类名相同

3.构造函数不是手动调用的，是创建对象值JVM主动调用的，每创建一个对象就会调用一次构造函数

4.构造函数可以通过重载的方式多个出现

5.当一个类没有创建造函数是编译器会自动给其添加一个无参的构造函数(继承自object类)

6.当一个类显现创建一个构造方法时编译器不会个其添加无参的构造方法。

8. 构造函数什么时候用？

创建对象时调用该类的构造方法，对成员变量进行初始化赋值。

9. 构造代码块的作用？

对该类的对象进行统一的初始化赋值

10. this关键字的特点和使用以及应用场景？

this关键字指向本类对象

1.this关键字引用本类的构造方法时必须出现在第一句。

2.使用this关键字引用本类构造方法时要避免各自调用

3.this关键字代表本类对象，只能在非静态成员中使用

4.如果在一个函数中访问一个成员变量，而且没有存在同名的局部变量 时，那么 java编译器会默认在变量的前面加上 this关键字的。

}

5 作业

1. 静态和非静态的区别。说一下内存

调用方式：

静态成员 ：可以通过类名.静态成员及对象.静态成员访问

非静态成员 ：可以通过对象.非静态成员访问

访问限制：

静态 ：静态成员函数不能够直接使用非静态成员变量，能够直接使用静态成员变量

非静态 :非静态成员函数能够直接使用非静态成员及静态成员。

this,super关键字的使用：

静态 ：静态函数不能够使用this与super关键字

非静态 ：非静态函数能够调用thi与super关键字

一个类文件加载到内存中，JVM会对该类文件进行解析,并将分析出该类文件具有哪些成员及哪些静态数据。

一个方法在运行时JVM会在栈内存中分配对应的内存空间

1.当前内存由栈内存、堆内存、方法区(java总共有5片内存)

2.类加载时先加载class文件进入共享区

其中该类的main方法加载进方法栈内存

.class 加载进方法区

堆内存开辟空间，实例变量进行默认初始化，显示初始化。

内存地址传给变量 p1，栈和堆建立连接

3.创建该类对象

1：堆内存开辟空间，实例变量进行默认初始化，显示初始化。

2：内存地址传给变量 p2，栈和堆建立连接

4.建立多个该类对象

每一个该类对象都可以访问Static中的静态变量

2. 成员变量和静态变量的区别？

访问方式：

静态成员变量 ：可以通过类名.静态成员及对象.静态成员访问

非静态成员变量 ：可以通过对象.非静态成员访问

数量差距：

静态成员变量 ：只会在方法区中存在一份，同类对象共同引用相同的静态成员变量

非静态成员变量 ：每个对象中都会存在一份自己的非静态成员变量

储存位置:

静态成员变量 ：在方法区中有且仅存在一份

非静态成员变量 ：在堆内存中的每一个对象空间中存在一份

作用区别：

静态成员变量 ：静态成员变量的作用是提供一个共享数据给所有对象使用的。

非静态成员变量 ：非静态成员变量的作用是用于描述一类事物的属性。

生命周期：

静态成员变量 ：随着该类的加载而产生，随着类的消失而消失

非静态成员变量 ：随着对象的产生而产生，随着对象的消失而消失

3. 静态的特点以及注意事项？

4. 什么时候使用静态？

对于成员变量来说，如果该成员变量对于该类的所有对象来说只需要存在一份即可时(共享给该类使用对象共同使用)可以使用static进行修饰

对于成员函数来说，如果该成员函数没有直接使用该类的非静态成员变量时可以使用static进行修饰。

5. 继承的好处？

1.减少重复代码

2.描述类与类之间的相互关系

6. java改良多继承的原因？

如果一个类能够继承多个父类会造成继承关系复杂，当一个类能够继承多个父类时构造函数的调用难以确定。

7. 当使用一个已存在的继承体系时，该如何更快应用

直接使用子类，如果子类的方法不能够满足需求，可以对其非静态函数进行重写

8. 什么时候用继承？

当类与类之间真实存在继承关系时使用继承

9. super 和 this 的特点？

super：

super指向父类的引用空间

super引用父类构造函数时需要出现在构造函数的第一句。

super引用父类构造函数时不能够与this关键字一起使用。

this:

this代表该类对象

this关键字调用本类构造函数时需要出现在构造函数的第一句

this关键字调用本类构造函数时不能够与super共同使用

this关键调用本类构造函数时要避免重复调用

this与super关键字都不能与static共同使用

10. 覆盖的特点，何时应用，注意事项？

当父类的方法不能够满足需求是，需要在子类中重写父类的方法

前提：类与类之间存在继承或者实现的关系

1.函数名一致

2.形参列表一致

3.子类方法的返回值类型小与或者等于父类方法的返回值

4.子类方法的访问权限修饰符要大于或者等于父类方法的访问权限修饰符

5.子类的异常报错类型要小与父类方法的异常报错类型

11. 子类的实例化过程？为什么是这样的实例化过程？

1.子类class对象加载到方法区

2.栈内存创建子类的引用类型变量

3.堆内存创建子类对象的内存空间

4.父类class对象加载到方法区

5.解析父类有哪些成员变量及公共静态方法

6.子类使用super方法引用父类的空间

12. super语句，和 this语句为什么不能同时存在，super为什么要定义在第一行？

因为this与super都要出现在构造函数的第一句，这要就相互矛盾了。

6.作业

1.抽象类的特点，以及细节？

1.接口类没有构造方法

2.接口类并创建对象

3.接口类实现时需要使用implement关键字，一个类可以实现多个接口

4.接口中的常量属性默认是public static final，即所有常量都是静态常量，定义时必须进行初始化赋值

5.有抽象函数的类可以是抽象类也可以是抽象方法，抽象类不一定有抽象方法。

6.由于接口中的方法默认都是抽象的，所以不能被实例化。

static修饰的方法可以被子类继承，但是在继承时进行了隐藏，不能够被子类重新。

abstract修饰的方法需要被被子类重写以实现该方法。

7.对于接口而言，可以使用子类来实现接口中未被实现的功能函数。

8.如果实现类中要访问接口中的成员，不能使用super关键字。因为两者之间没有显示的继承关系，况且接口中的成员成员属性是静态的。可以使用接口名直接访问。

2.接口的表现形式的特点。

接口的定义：

interface 接口{

public abstract void 方法();

}

接口的实现

class 实现类 implement 接口{

}

3.接口的思想特点，要举例。

车具有空调的功能

4.多实现和多继承的区别？

多实现：实现多个接口

多继承：继承多个父类

5.抽象类和接口的区别？

抽象类是一类事物的抽象，是从类中抽取出来的

接口是功能方法的抽象

6.多态的体现，前提，好处，弊端。

体现：父类的引用类型变量指向子类对象，父类引用也可以接受自己的子类对象

前提：必须存在继承或者实现关系

好处：提高扩展性，方法可以接收及返回更多的对象类型

1. 应用程序不必为每一个派生类编写功能调用，只需要对抽象基类进行处理即可。大大提高程序的可复用性。//继承

2. 派生类的功能可以被基类的方法或引用变量所调用，这叫向后兼容，可以提高可扩充性和可维护性。 //多态的真正作用

弊端：父类的引用类型变量不能够使用子类特有的方法，如果需要使用需要进行数据类型转化

5. 作业

1.为什么要将一个类定义成内部类？

内部类能够减少代码量，简化书写

2.匿名内部类的使用和细节(面试题)

public void 方法(){

class Inner(){

内部类属性;

内部类方法;

}.内部类方法;

};

public void 方法(){

Father f =new Father(){

内部类属性;

内部类方法一;

内部类方法二;

};

f.内部类方法一;

f.内部类方法二;

};

public void 方法(){

new Father(){

内部类属性;

重写父类方法;

内部类方法(return this);

}。内部类方法.重写父类方法;

};

3.异常的思想和体系特点？

|throwable

|error(JVM或者硬件导致的错误，不能够进行处理只能够尽量避免)

|exception(可以通过java代码进行处理)

|运行时异常

|编译时异常

4.throws和 throw的如何使用？

throws在方法声明使用，用于声明抛出异常

throw在方法中使用，用于抛出具体的异常

5.什么时候 try什么时候 throws？

需要捕获异常，对异常进行处理时使用try-catch关键字对异常进行捕获处理

如果不处理异常，需要将异常抛出是可以使用throws抛出异常。

6.编译时被检测异常和运行时异常的区别？

编译时异常，编译不能同过，体现在程序中就是由红色感叹号。

运行时异常，编译可以通过，程序没有红色感叹号，但是在运行时会有错误发生。

7.异常的所有细节？

8.finally的应用？

try-catch捕获异常的最后步骤，通常用于关闭资源。

9.包的作用，名称空间的定义和理解？

1 把功能相似或相关的类或接口组织在同一个包中，方便类的查找和使用。

2 如同文件夹一样，包也采用了树形目录的存储方式。同一个包中的类名字是不同的，不同的包中的类的名字是可以相同的，当同时调用两个不同包中相同类名的类时，应该加上包名加以区别。因此，包可以避免名字冲突。

3 包也限定了访问权限，拥有包访问权限的类才能访问某个包中的类。

在一个.java文件中可以一个public类和多个非public类，如果要将这些类组织在一个包当中，则在.java文件中除注释以外的第一行使用关键字package即可实现。当需要调用此包中的类时，就可以使用关键字import进行导入。在定义包的时候，应该注意几点:

1 为了尽量使包名保持唯一性，包名通常采用小写、按倒写互联网址的形式进行定义。

2 在进行命名包时，应该避免使用与系统发生冲突的名字。

10.jar包的基本使用。只要将类和包都存储到jar中，方便于使用。只要将jar配置到classpath路径下

6.作业

1.抽象类的特点，以及细节？

1.接口类没有构造方法

2.接口类并创建对象

3.接口类实现时需要使用implement关键字，一个类可以实现多个接口

4.接口中的常量属性默认是public static final，即所有常量都是静态常量，定义时必须进行初始化赋值

5.有抽象函数的类可以是抽象类也可以是抽象方法，抽象类不一定有抽象方法。

6.由于接口中的方法默认都是抽象的，所以不能被实例化。

static修饰的方法可以被子类继承，但是在继承时进行了隐藏，不能够被子类重新。

abstract修饰的方法需要被被子类重写以实现该方法。

7.对于接口而言，可以使用子类来实现接口中未被实现的功能函数。

8.如果实现类中要访问接口中的成员，不能使用super关键字。因为两者之间没有显示的继承关系，况且接口中的成员成员属性是静态的。可以使用接口名直接访问。

2.接口的表现形式的特点。

接口的定义：

interface 接口{

public abstract void 方法();

}

接口的实现

class 实现类 implement 接口{

}

3.接口的思想特点，要举例。

车具有空调的功能

4.多实现和多继承的区别？

多实现：实现多个接口

多继承：继承多个父类

5.抽象类和接口的区别？

抽象类是一类事物的抽象，是从类中抽取出来的

接口是功能方法的抽象

6.多态的体现，前提，好处，弊端。

体现：父类的引用类型变量指向子类对象，父类引用也可以接受自己的子类对象

前提：必须存在继承或者实现关系

好处：提高扩展性，方法可以接收及返回更多的对象类型

1. 应用程序不必为每一个派生类编写功能调用，只需要对抽象基类进行处理即可。大大提高程序的可复用性。//继承

2. 派生类的功能可以被基类的方法或引用变量所调用，这叫向后兼容，可以提高可扩充性和可维护性。 //多态的真正作用

弊端：父类的引用类型变量不能够使用子类特有的方法，如果需要使用需要进行数据类型转化

5. 作业

1.为什么要将一个类定义成内部类？

内部类能够减少代码量，简化书写

2.匿名内部类的使用和细节(面试题)

public void 方法(){

class Inner(){

内部类属性;

内部类方法;

}.内部类方法;

};

public void 方法(){

Father f =new Father(){

内部类属性;

内部类方法一;

内部类方法二;

};

f.内部类方法一;

f.内部类方法二;

};

public void 方法(){

new Father(){

内部类属性;

重写父类方法;

内部类方法(return this);

}。内部类方法.重写父类方法;

};

3.异常的思想和体系特点？

|throwable

|error(JVM或者硬件导致的错误，不能够进行处理只能够尽量避免)

|exception(可以通过java代码进行处理)

|运行时异常

|编译时异常

4.throws和 throw的如何使用？

throws在方法声明使用，用于声明抛出异常

throw在方法中使用，用于抛出具体的异常

5.什么时候 try什么时候 throws？

需要捕获异常，对异常进行处理时使用try-catch关键字对异常进行捕获处理

如果不处理异常，需要将异常抛出是可以使用throws抛出异常。

6.编译时被检测异常和运行时异常的区别？

编译时异常，编译不能同过，体现在程序中就是由红色感叹号。

运行时异常，编译可以通过，程序没有红色感叹号，但是在运行时会有错误发生。

7.异常的所有细节？

8.finally的应用？

try-catch捕获异常的最后步骤，通常用于关闭资源。

9.包的作用，名称空间的定义和理解？

1 把功能相似或相关的类或接口组织在同一个包中，方便类的查找和使用。

2 如同文件夹一样，包也采用了树形目录的存储方式。同一个包中的类名字是不同的，不同的包中的类的名字是可以相同的，当同时调用两个不同包中相同类名的类时，应该加上包名加以区别。因此，包可以避免名字冲突。

3 包也限定了访问权限，拥有包访问权限的类才能访问某个包中的类。

在一个.java文件中可以一个public类和多个非public类，如果要将这些类组织在一个包当中，则在.java文件中除注释以外的第一行使用关键字package即可实现。当需要调用此包中的类时，就可以使用关键字import进行导入。在定义包的时候，应该注意几点:

1 为了尽量使包名保持唯一性，包名通常采用小写、按倒写互联网址的形式进行定义。

2 在进行命名包时，应该避免使用与系统发生冲突的名字。

10.jar包的基本使用。只要将类和包都存储到jar中，方便于使用。只要将jar配置到classpath路径下

异常体系