●10.01_面向对象(package 关键字的概述及作用)(了解)

- A:为什么要有包
  - 将字节码(.class)进行分类存放
  - 包其实就是文件夹
- B:包的概述
-

举例：

学生：增加，删除，修改，查询

老师：增加，删除，修改，查询

方案 1：按照功能分

com.cssxt.add

AddStudent

AddTeacher

com.cssxt.delete

DeleteStudent

DeleteTeacher

com.cssxt.update

UpdateStudent

UpdateTeacher

com.cssxt.find

FindStudent

FindTeacher

方案 2：按照模块分（开发中推荐）

com.cssxt.teacher

AddTeacher

DeleteTeacher

UpdateTeacher

FindTeacher

com.cssxt.student

AddStudent

DeleteStudent

UpdateStudent

FindStudent

●10.02_面向对象(包的定义及注意事项)(掌握)

- A:定义包的格式
  - package 包名;
  - 多级包用.分开即可
- B:定义包的注意事项
  - A:package 语句必须是程序的第一条可执行的代码
  - B:package 语句在一个 java 文件中只能有一个
  - C:如果没有 package，默认表示无包名
- C:案例演示
  - 包的定义及注意事项

```
package com.baidu.add;//package 一定是在第一行
import com.gogoole.add.*;//引入类的全路径名,*表示引入 com.gogoole.add 目录下所有的类,会有一个索引的过程,会消耗一些时间
//import com.gogoole.add.Person//开发中推荐引入到指明的类名

public class Student extends Person{

    public Student(String name,int age){
        super(name,age);
    }
    public static void main(String[] args) {
        Student stu = new Student("小明",19);
        stu.show();

        //如果不引入包只能这么写,也是可以的,但不推荐
       com.gogoole.add.Person p = new  com.gogoole.add.Person("张三",20);
        p.show();
    }
}
```

●10.03_面向对象(带包的类编译和运行)(了解开发中不会这么用)

- A:如何编译运行带包的类
  - a:javac 编译的时候带上-d 即可
    - javac -d . HelloWorld.java
  - b:通过 java 命令执行。
    - java 包名.HellWord

●10.04_面向对象(不同包下类之间的访问)(掌握)

- A:案例演示

- 不同包下类之间的访问

●10.05_面向对象(import 关键字的概述和使用)(掌握)

- A:案例演示
  - 为什么要有 import
    - 其实就是让有包的类对调用者可见,不用写全类名了

- B:导包格式
  - import 包名;
  - 注意：
  - 这种方式导入是到类的名称。
  - 虽然可以最后写*号，但是不建议。
- C:package,import,class 有没有顺序关系(面试题)

有顺序，顺序为：package,import,class(interface)

●10.06_面向对象(四种权限修饰符的测试)(掌握)

- A:案例演示
  - 四种权限修饰符
- B:结论
-

	本类	同一个包下(子类和无关类)	不同包下(子类)	不同包下(无关类)
private	Y			
默认	Y	Y		
protected	Y	Y	Y	
public	Y	Y	Y	Y

默认修饰符演示

在同一个类中:

```
package com.sst;

public class Pojo {
    String name;
    String sex;
    void print(){  // 没有修饰符的就是默认(缺省)修饰符
      System.out.println(name+sex);
    }

    public static void main(String[] args) {
        Pojo p = new Pojo();
        p.name = "zhagsan";
        p.sex ="男";
        p.print();
```

```
    }
}
```

在同一个包中:

```
package com.sst;

public class Student extends Pojo{
    public static void main(String[] args) {
        Student stu = new Student();
        stu.name ="李四";
        stu.sex = "女";
        stu.print();
    }
}
```

```
package com.sst;

public class Pojo {
    String name;
    String sex;
    void print(){  // 没有修饰符的就是默认(缺省)修饰符
      System.out.println(name+sex);
    }

    public static void main(String[] args) {
        Pojo p = new Pojo();
        p.name = "zhagsan";
        p.sex ="男";
        p.print();

    }
}
```

在不同的包下:

```
package com.com.ssh;

import com.sst.Pojo;

public class Worker extends Pojo {
    public static void main(String[] args) {
        Worker w = new Worker();
        w.name="";//在不同的报下,默认修饰符修饰的变量访问不到
   }
}
```

protected演示

在不同包下的子类

```java
package com.sst;

public class Pojo {
    protected String name;
    protected String sex;
    protected void print(){  // 没有修饰符的就是默认(缺省)修饰符
       System.out.println(name+sex);
    }
}

package com.com.ssh;

import com.sst.Pojo;

public class Worker extends Pojo {
    public static void main(String[] args) {
        Worker w = new Worker();
        w.name="小红";
        w.sex ="女";
        w.print();
    }
}
```

在不同的包下没有关系的类中:

```java
package com.com.ssh;

        import com.sst.Pojo;

public class Worker{
    public static void main(String[] args) {
        Pojo p = new Pojo();
        p.name="";// 报错了
   }
}


package com.sst;

public class Pojo {
    protected String name;
    protected String sex;
    protected void print(){  // 没有修饰符的就是默认(缺省)修饰符
       System.out.println(name+sex);
    }
}
```

重点掌握:public \private

●10.07_面向对象(类及其组成所使用的常见修饰符)(掌握)

- A:修饰符：
  - 权限修饰符：private，默认的，protected，public
  - 状态修饰符：static，final
  - 抽象修饰符：abstract
- B:类：
  - 权限修饰符：默认修饰符，public
  - 状态修饰符：final
  - 抽象修饰符：abstract

  - 用的最多的就是：public

- C:成员变量：
  - 权限修饰符：private，默认的，protected，public
  - 状态修饰符：static，final

  - 用的最多的就是：private

- D:构造方法：
  - 权限修饰符：private，默认的，protected，public

  - 用的最多的就是：public

- E:成员方法：
  - 权限修饰符：private，默认的，protected，public
  - 状态修饰符：static，final
  - 抽象修饰符：abstract

  - 用的最多的就是：public,private

- F:除此以外的组合规则：
  - 成员变量：public static final 变量名
  - 成员方法：
    - public static
    - public abstract
    - public final

●10.08_面向对象(内部类概述和访问特点)(了解)

- A:内部类概述
- B:内部类访问特点
  - a:内部类可以直接访问外部类的成员，包括私有。
  - b:外部类要访问内部类的成员，必须创建对象。
  - 外部类名.内部类名 对象名 = 外部类对象.内部类对象;
- C:案例演示
  - 内部类极其访问特点

```java
package com.sxt.interclass;

public class Outer {
    public static void main(String[] args) {
        //内部类创建实例对象的方式一:
        Outer.Inter in = new Outer().new Inter();
        in.getMsg();

        //方式二:
        Outer out = new Outer();
        Outer.Inter in2 = out.new Inter();
        in2.getMsg();
    }


    private int a = 10;
    int b = 20;
    public void show1(){
        System.out.println("外部类中的 show1");
    }
    private void show2(){
        System.out.println("外部类中的 show2");
    }

    //成员内部类
   class Inter{
        int c = 30;
        public void getMsg(){
            System.out.println("a:"+a);//在内部类中可以方法外部类的私有成员属性
          System.out.println("b:"+b);
            show1();
            show2();
        }
    }
}
```