面向对象第一天：

回顾：

1. 数组：
   - 数组的复制：

     ```
     System.arraycopy(a,1,b,0,4);
     int[] b = Arrays.copyOf(a,6);
     a = Arrays.copyOf(a,a.length+1); //数组的扩容
     ```

   - 数组的排序：

     ```
     Arrays.sort(arr); //升序排列
     ```

2. 方法：

   封装一段特定的业务逻辑功能，尽可能的独立，只干一件事，

   可以被反复调用，可以减少代码重复，有利于代码维护

3. 方法的定义：五要素

   修饰词　返回值类型　方法名(参数列表){

   　　方法体

   }

4. 方法的调用：
   - 无返回值：方法名(有参传参);
   - 有返回值：数据类型 变量 = 方法名(有参传参);

     System.out.println(方法名(有参传参));

5. return：
   - return 值; //1)结束方法 2)返回结果给调用方
   - return; //1)结束方法

精华笔记：

1. 什么是类？什么是对象？
   - 现实生活中是由很多很多对象组成的，基于对象抽出了类
   - 对象：软件中真实存在的单个个体/东西

     类：类型/类别，代表一类个体
   - 类是对象的模板/模子，对象是类的具体的实例
   - 类中可以包含：
     - 对象的属性/特征----------成员变量
     - 对象的行为/动作/功能----------方法
   - 一个类可以创建多个对象

2. 如何创建类？如何创建对象？如何访问成员？
3. 方法的签名：方法名+参数列表
4. 方法的重载(overload/overloading)：----------方便用户的调用
    - 发生在同一类中，方法名相同，参数列表不同
    - 编译器在编译时会根据方法的签名自动绑定方法

笔记：

1. 什么是类？什么是对象？
    - 现实生活中是由很多很多对象组成的，基于对象抽出了类
    - 对象：软件中真实存在的单个个体/东西

      类：类型/类别，代表一类个体
    - 类是对象的模板/模子，对象是类的具体的实例
    - 类中可以包含：
        - 对象的属性/特征----------成员变量
        - 对象的行为/动作/功能----------方法
    - 一个类可以创建多个对象
2. 如何创建类？如何创建对象？如何访问成员？

```
package ooday01;
//学生类
public class Student {
    //成员变量----对象的属性
    String name;
    int age;
    String address;

    //方法-----对象的行为/功能
    void study(){
        System.out.println(name+"在学习...");
    }
    void sayHi(){
        System.out.println("大家好，我叫"+name+"，今年"+age+"岁了，家
住"+address);
    }
}

package ooday01;
//学生类的测试类
public class StudentTest {
    public static void main(String[] args) {
        //创建一个学生对象
        Student zs = new Student();
        //给成员变量赋值
        zs.name = "zhangsan";
        zs.age = 18;
        zs.address = "河北廊坊";
        //调用方法
        zs.study();
```

```
        zs.sayHi();

        Student ls = new Student();
        ls.name = "lisi";
        ls.age = 25;
        ls.address = "黑龙江佳木斯";
        ls.study();
        ls.sayHi();

        //1)创建了一个学生对象
        //2)给所有成员变量赋默认值
        Student ww = new Student();
        ww.study();
        ww.sayHi();

    }
}
```

3. 方法的签名：方法名+参数列表
4. 方法的重载(overload/overloading)：----------方便用户的调用
    - 发生在同一类中，方法名相同，参数列表不同

```
public class Aoo{
    void show(){}
    void show(String name){}
    void show(int age){}
    void show(String name,int age){}
    void show(int age,String name){}

    //int show(){ return 1; } //编译错误，重载与返回值类型无关
    //void show(String address){} //编译错误，重载与参数名称无关
}
```

    - 编译器在编译时会根据方法的签名自动绑定方法

```
public class OverloadDemo {
    public static void main(String[] args) {
        Aoo o = new Aoo();
        o.show();
        o.show("zhangsan");
        o.show(25);
        o.show("zhangsan",25);
        o.show(25,"zhangsan");
    }
}
```

补充：

1. OO：面向对象

   OOA：面向对象分析

   OOD：面向对象设计

OOP：面向对象编程----------你们以后所参与的

2. 高质量的代码：----------想拿年薪
   - 复用性好、扩展性好、维护性好、移植性好、健壮性好、可读性好、效率好.....
3. 类：是一种引用数据类型，是我们自己创造的一种数据类型
4.
```
    引用
数据类型   引用类型变量   指向       对象
Student      zs         =    new Student();
//读作:声明一个Student类型的引用zs，指向了一个学生对象
```

5. 默认值：

```
byte,short,int,long,char-----------------0
float,double-----------------------------0.0
boolean----------------------------------false
引用类型----------------------------------null
```

6. 潜艇游戏需求：
   - 所参与的角色：战舰、炸弹、侦察潜艇、鱼雷潜艇、水雷潜艇、水雷
   - 角色间的关系：
     - 战舰发射炸弹
     - 炸弹打潜艇(侦察潜艇、鱼雷潜艇、水雷潜艇)，若打中了：
       - 潜艇消失、炸弹消失
       - 若打中的是侦察潜艇，则玩家得10分

         若打中的是鱼雷潜艇，则玩家得40分

         若打中的是水雷潜艇，则战舰增1条命
     - 水雷潜艇发射水雷
     - 水雷打战舰，若打中了：
       - 水雷消失
       - 战舰减1条命(命数为0时游戏结束)
7. 明日单词：

```
1)this:这个
2)Pointer:指针
3)random:随机
```

作业：需要上传的，项目经理会检查的

1. Student类与StudentTest类

晚上的练习任务：----------每人最少写2次

1. 将潜艇游戏的代码最少写两次(建项目、建包、建类...)

> main中：还是得创建11个对象，但是只需要测试3、4个对象即可

扩展任务：----------选做

1. 飞机大战/射击游戏项目