

task1

1.

```
public class Person {
    private String name;
    private int age;
    private int sex;

    public Person(String name, int age, int sex) {
        this.name = name;
        this.age = age;
        this.sex = sex;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    private int getAge() {
        return age;
    }

    private void setAge(int age) {
        this.age = age;
    }

    public int getSex() {
        return sex;
    }

    public void setSex(int sex) {
        this.sex = sex;
    }

    public String getAddress() {
        return address;
    }

    public void setAddress(String address) {
        this.address = address;
    }

    private void eat() {
        System.out.println(name + "正在吃东西");
    }

    private void sleep() {
```

```

        System.out.println(name + "正在睡觉");
    }

    private void dadoudou() {
        System.out.println(name + "正在打豆豆");
    }

    public void dailyActivity() {
        eat();
        sleep();
        dadoudou();
    }
}

```

this的作用是：

**表示成员变量**，避免与方法的参数名命名冲突

**调用本类的其他构造方法**，避免代码重复，复用已有的构造逻辑，当需要修改属性初始化逻辑时，只需修改普通构造方法即可

可以在方法中**返回当前对象本身**，常用于链式调用

2.

```

public class Main {
    public static void main(String[] args) {
        Person person = new Person("微光", 18, 1); // 1表示男性
        System.out.println("对象: ");
        System.out.println("姓名: " + person1.getName());
        System.out.println("年龄: " + person1.getAge());
        System.out.println("性别 (1=男, 2=女): " + person1.getSex());
        person1.dailyActivity();
    }
}

```

总体上**先有类，后有对象**；**一个类可以创建多个对象**：

类是对同一类事物的**抽象描述**（对属性和行为的规定），对象必须在定义类后才能创建**具体属性值**，并调用类定义的方法来执行行为；

所有通过同一个类创建的对象，都会**继承类定义的方法**，但各自的属性值独立

3

```

public class Person {
    private String name;
    private int age;
    protected int sex;
    String address;

    public Person(String name, int age, int sex, String address) {
        this.name = name;
        this.age = age;
        this.sex = sex;
    }
}

```

```
        this.address = address;
    }

    public Person(Person other) {
        this(other.name, other.age, other.sex, other.address);
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    protected int getAge() {
        return age;
    }

    protected void setAge(int age) {
        this.age = age;
    }

    public int getSex() {
        return sex;
    }

    public void setSex(int sex) {
        this.sex = sex;
    }

    public String getAddress() {
        return address;
    }

    public void setAddress(String address) {
        this.address = address;
    }

    private void eat() {
        System.out.println(name + "正在吃东西");
    }

    private void sleep() {
        System.out.println(name + "正在睡觉");
    }

    public void doDailyActivities() {
        eat();
        sleep();
        sex();
    }
}
```

```
}

private void sex() {
    if ( sex == 1){
        System.out.println("男") ;
    } else {
        System.out.println("女");
    }
}

private void dadoudou() {
    System.out.println(name + "正在打豆豆");
}

public void dailyActivity() {
    eat();
    sleep();
    dadoudou();
}
}
```

### 1.**public (公共的)**

访问范围：所有地方都可以访问

适用对象：类、字段、方法、构造方法

### 2.**protected (受保护的)**

访问范围：本类中可以访问，同一包中的其他类可以访问，不同包中的子类可以访问

适用对象：字段、方法、构造方法（不能修饰类）

### 3.**默认 (包访问权限，无修饰符)**

访问范围：本类中可以访问，同一包中的其他类可以访问，不同包中的类（包括子类）不能访问

适用对象：类、字段、方法、构造方法

### 4.**private (私有的)**

访问范围：只能在本类中访问，其他任何地方（包括同包类和子类）都不能直接访问

适用对象：字段、方法、构造方法（不能修饰类）