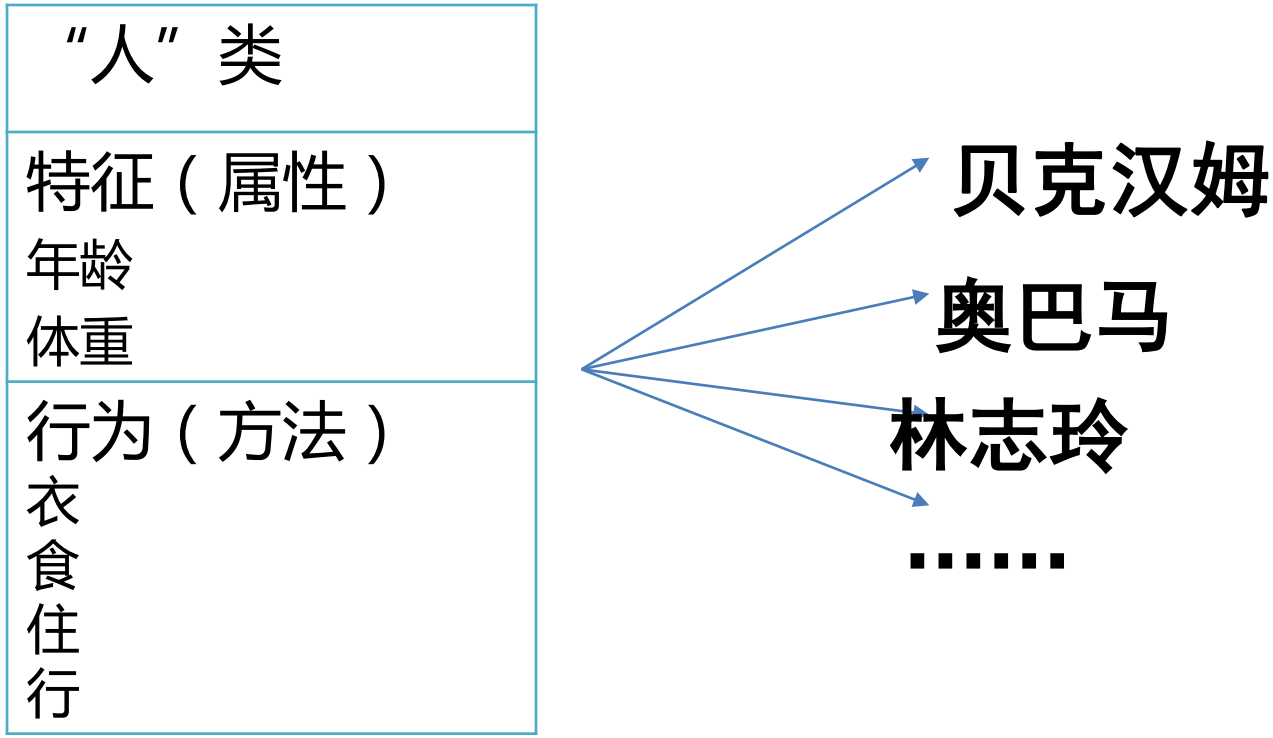
第五章 类的构建与对象的使用

类和对象



Java类语法

```
public class  类名 {
        //定义属性部分
        属性1的类型 属性1;
        属性2的类型 属性2;
            …
        属性n的类型 属性n;

        //定义方法部分
        方法1;
        方法2;
            …
        方法m;
}
```

定义属性

- 定义和使用属性

```
public class  类名 {

        //定义属性部分
        属性1的类型 属性1;
        属性2的类型 属性2;
            …
        属性n的类型 属性n;
}
```

属性

- 内存图
- 对象之间的独立性

定义方法

❑ 方法的定义格式

```
访问修饰符  返回值类型  方法名(参数1，参数2….，参数n){
    方法体；
}
```

带返回值方法

- 无参数，无返回值

  void 方法名( ){ 方法体；}

- 无参，有返回值

  int 方法名( ){ 方法体；}

- 有参，无返回值

  void 方法名(参数1，参数2….，参数n){ 方法体；}

- 有参，有返回值

  返回值类型 方法名(参数1，参数2….，参数n){ 方法体；}

this

- 方法调用中：
  - this
    - ✓作用

成员变量和局部变量区别

```
public class Person{

    变量1类型  变量1;
    变量2类型  变量2;
    变量3类型  变量3;

    public 返回类型 方法1(){
        变量4类型  变量4;
    }

    public 返回类型 方法2(){
        变量5类型   变量5;
    }
}
```

1. 作用域
2. 优先级
3. 初始值

如何使用带参数的方法

定义带参数的方法

```
public class Juicer{

    public String juicing ( String food ) {

            return “苹果”  +  "汁";

    }
}
```

调用带参数的方法

```
Juicer juicer_1 = new Juicer();

String c“苹果汁”r = juicer_1 .juici“苹果”果");

System.out.println(“苹果汁”er);
```

方法参数传递

- 值类型参数
  - 示例1
- 引用类型参数
  - 示例2
  - 示例3

可变参数方法

可变参数方法…