Classes and Objects

我们定义类

```
public class Baby {
    String name;
    boolean isMale;
    double weight;
    double decibels;
    int numPoops = 0;
    void poop() {
        numPoops += 1;
        System.out.println(“Dear mother, ”+“I have pooped. Ready the diaper.”);
    }
}
```

我们声明一个新的对象，就可以直接使用类来进行声明

```
Baby mybaby=new Baby();
```

类的自定义初始化：就是在类中写一个同名的初始化函数

```
public class Baby {
    String name;
    boolean isMale;
    Baby(String myname, boolean maleBaby){
        name = myname;
        isMale = maleBaby;
    }
}
```

这样就可以将我们的baby的名字和性别都初始化了

还可以在其中增加函数

```
public class Baby {
    String weight = 5.0;
    void eat(double foodWeight) {
        if (foodWeight >= 0 && foodWeight < weight) {
            weight = weight + foodWeight;
        }
    }
}
```

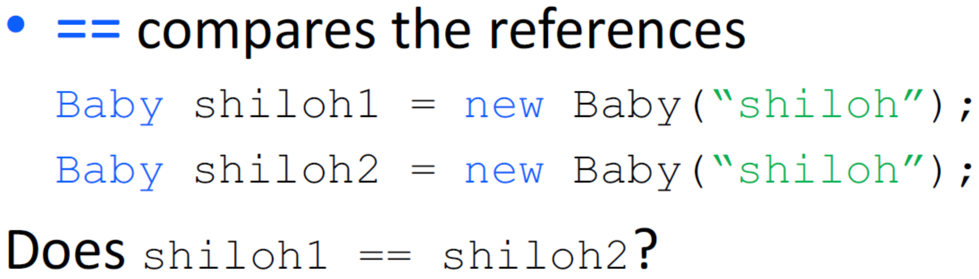
练习：设置我们的main函数和baby函数

```java
class Baby{
    String name;
    int age;
    boolean isMale;
    double weight;
    Baby(String my_name, int my_age, boolean my_isMale, double my_weight){
        name = my_name;
        age = my_age;
        isMale = my_isMale;
        weight = my_weight;
    }
    void eat(double add_weight){
        if(isMale){
            weight=weight+add_weight*0.5;
        }
        else{
            weight=weight+add_weight*0.4;
        }
    }
    void duanlian(double low_weight){
        if(isMale){
            weight=weight-low_weight*0.5;
        }
        else{
            weight=weight-low_weight*0.4;
        }
    }
    void show_weight(){
        System.out.println(weight);
    }

}
```

```java
public class main {
        public static void main(String[] args) {
        Baby lhz = new Baby("lhz", 19, true, 70.0);
        Baby abc = new Baby("abc", 18, false, 60.0);
        System.out.println(lhz.weight);
        System.out.println(abc.weight);
        System.out.println(lhz.isMale);
        System.out.println(abc.isMale);
        lhz.show_weight();
        lhz.eat(20.0);
        lhz.show_weight();
        lhz.duanlian(20);
        lhz.show_weight();
        abc.show_weight();
        abc.eat(20.0);
        abc.show_weight();
        abc.duanlian(20);
        abc.show_weight();
    }
}
```

这样就可以实现我们的类的功能了



- == compares the references

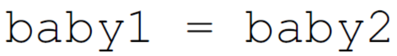
```
Baby shiloh1 = new Baby(“shiloh”);
Baby shiloh2 = new Baby(“shiloh”);
```

Does `shiloh1 == shiloh2`?

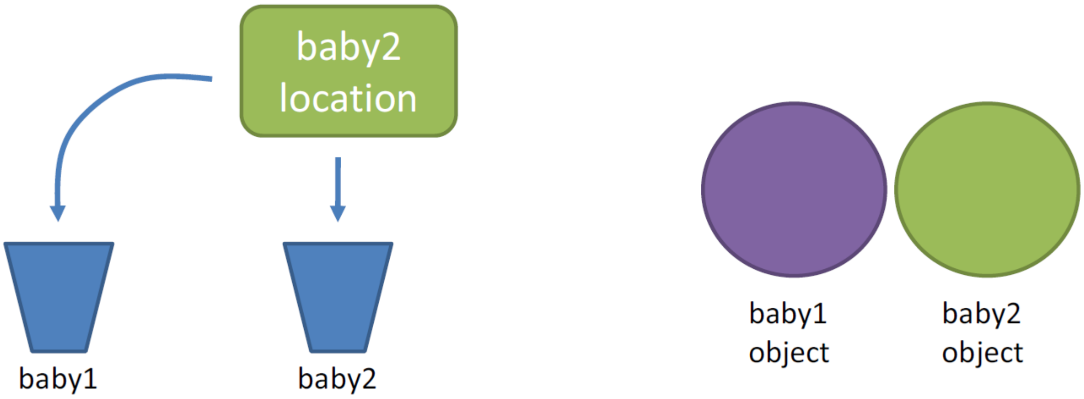
no

需要注意的是两个类就算所有的属性都是相同的，这两个类也是不一样的，不能划上等号

但是可以在类与类之间互相赋值



```
baby1 = baby2
```



静态变量可以修改值