

第02课

背包里的变量盒子

给数据一个名字，追踪一次小小的冒险



欢迎屏里的玩家名一直写死成 Alex。如果来了另一位玩家，每处名字都要重新找一遍。变量给了我们更好的办法：把数据放在有名字的位置，读取它，也更新它。这一课要弄明白的，不只是“怎么写变量”，还有“它在这一刻究竟保存了什么”。

阅读方式：先猜结果，再运行核对；每次只改一个条件。

使用 Java 11 或更新版本。故事与插画借用方块世界，本系列练习运行在普通 Java 控制台。

01 标签帮我们找到当前的值



储藏室靠标签找物品；程序靠变量名读取数据。

史蒂夫把面包放进没有标签的箱子，晚饭时间却打开了圆石箱。变量名解决的是类似问题：让程序找到需要的数据。它不是数据本身，而是访问数据的名字。

先从两种类型开始：int 保存整数，String 保存文字。生命值、物品数量适合整数；玩家名适合文字。更多类型下一课再展开。

本课的生命恢复和物品变化都是练习规则。输出是控制台中的状态快照，不会自动变成游戏界面，也不会在程序结束后自动保存。

02 声明、初始化、读取

读懂 `int health = 20;`



完整程序 • `PlayerStatus.java`

```
public class PlayerStatus {  
    public static void main(String[] args) {  
        String playerName = "Alex";  
        int health = 20;  
        int bread = 3;  
        System.out.println(playerName + " 生命=" + health);  
        System.out.println("面包=" + bread);  
    }  
}
```

运行核对

Alex 生命=20
面包=3

`health` 不加引号时，读取变量的值；`"health"` 加上引号后，只是一段固定文字。把玩家名改成 `Steve`、面包改成 `5`，先猜两行输出，再运行。

03 赋值：先算右边，再写左边



赋值像一次搬运：先准备好结果，再放回指定位置。

bread = bread - 1 的三个动作

读取旧值

bread 是 3



计算右侧

3 - 1 得到 2



写回左侧

bread 变成 2

片段：放在 main 中，单独运行

```
int bread = 3;
System.out.println("吃之前=" + bread);
bread = bread - 1;
System.out.println("吃之后=" + bread);
int diamonds = 0;
diamonds = diamonds + 3;
System.out.println("钻石=" + diamonds);
```

运行核对

吃之前=3
吃之后=2
钻石=3

= 是一次更新，不是数学等式。第一次创建写 `int bread = 3;`，以后更新同一个变量写 `bread = 2;`，不要在同一作用域里再声明一次。

04 一次输出，就是一张快照

名字不等于内容：四项当前数据

变量名	类型	当前值
playerName	String	"Alex"
health	int	20
bread	int	3
diamondCount	int	0

先前输出的那一行文字不会跟着变量变动。想展示新状态，要再次输出。重新运行程序时，初始化语句也会重新执行。

片段：放在 main 中，单独运行

```
int chestA = 5;
int chestB = chestA;
chestA = 9;
System.out.println("A=" + chestA);
System.out.println("B=" + chestB);
```

运行核对

```
A=9
B=5
```

这里复制的是 int 数值：执行赋值时，把当时的5存进 chestB。之后只改 chestA，chestB 不会自动同步。对象与引用会在后面的课程专门学习。

05 沿着时间线更新生命值

同一变量，四个时刻



片段：放在 main 中，单独运行

```
int health = 20;
System.out.println("出生=" + health);
health = health - 4;
System.out.println("受伤=" + health);
health = health + 3;
System.out.println("进食=" + health);
health = 20;
System.out.println("恢复初值=" + health);
int arrows = 10;
arrows += 5;
arrows -= 2;
System.out.println("箭=" + arrows);
```

运行核对

出生=20
受伤=16
进食=19
恢复初值=20
箭=13

最后的20来自明确赋值，程序没有自动限制生命上限。对本课 int 变量，+= 与 -= 是相应加减后写回的简写；先学会完整写法，再使用简写。

06 好名字让代码自己解释自己

给命名路牌做一次检查

例子	判断
playerName	清楚，使用 camelCase
diamondCount	说明这是钻石数量
2player	错误：不能以数字开头
player health	错误：名字不能有空格
class	错误：这是 Java 关键字
x	可以运行，但用途不清楚

初学时，建议小写英文字母开头，多个单词使用 camelCase：第一个单词小写，后续单词首字母大写。Java 区分大小写，health 和 Health 不是同一个名字。

片段：放在 main 中，单独运行

```
int playerLevel = 8;
int torchCount = 24;
String petName = "Snow";
System.out.println(playerLevel + " / " + torchCount);
System.out.println(petName);
```

运行核对

```
8 / 24
Snow
```

把 x、n、thing 改成读者不用猜的名字。规则只是底线；能让人理解用途，才是命名的价值。

07 变量排错：先分清是哪一种问题



检查一个原因，再观察结果，比把整座建筑拆掉更有效。

故意写错：三个不同的问题

```
int 2player = 10;
int bread = 3;
int bread = 2;
int diamonds;
System.out.println(diamonds);
```

- 2player 以数字开头，改成 playerLevel。
- 同一作用域重复声明 bread：更新时去掉第二个 int。
- 局部变量 diamonds 还没有值：读取前明确赋值。

片段：放在 main 中，单独运行

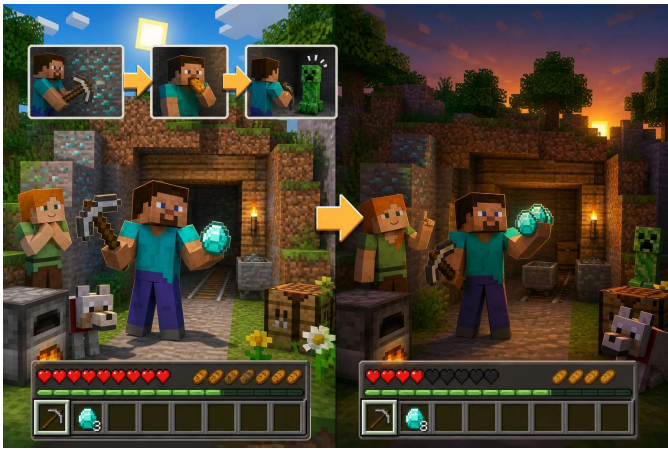
```
int playerLevel = 10;
int bread = 3;
bread = 2;
int diamonds = 0;
System.out.println(playerLevel + " / " + bread + " / " + diamonds);
```

运行核对

10 / 2 / 0

再核对类型：int health = "20"; 把文字交给整数变量，也会编译失败。第3课会把这个问题展开。

08 设计会变化的玩家状态栏



同一次冒险里，事件改变数据；新的输出记录改变后的状态。



09 完整作品与输出核对

完整程序 • AdventureStatus.java

```
public class AdventureStatus {
    public static void main(String[] args) {
        String playerName = "Alex";
        int health = 20;
        int bread = 3;
        int diamondCount = 0;
        System.out.println("出生 | " + playerName
            + " | 生命=" + health + " | 面包=" + bread
            + " | 钻石=" + diamondCount);
        health = health - 4;
        System.out.println("受伤 | " + playerName
            + " | 生命=" + health + " | 面包=" + bread
            + " | 钻石=" + diamondCount);
        health = health + 3;
        bread = bread - 1;
        diamondCount = diamondCount + 3;
        System.out.println("补给与挖矿 | " + playerName
            + " | 生命=" + health + " | 面包=" + bread
            + " | 钻石=" + diamondCount);
    }
}
```

运行核对

出生 | Alex | 生命=20 | 面包=3 | 钻石=0
受伤 | Alex | 生命=16 | 面包=3 | 钻石=0
补给与挖矿 | Alex | 生命=19 | 面包=2 | 钻石=3

只改玩家名，三行姓名都应改变。把伤害改为2，生命值依次为20、18、21；出现21并不是 Java 出错，而是你的练习规则没有设置上限。

10 带着预测继续练习



宠物狼也可以拥有自己的状态记录：名字、生命和物品数量。

第二课的能力地图



任选一个任务

- 基础：声明五种物品数量，输出一张背包清单。
- 进阶：铁锭从0开始，获得8个、用掉3个、再得到2个；三次输出应为8、5、7。
- 创造：设计宠物狼状态栏，模拟喂食和消耗骨头。

选学：两个箱子交换内容

片段：放在 main 中，单独运行

```
int leftChest = 12, rightChest = 30;
int temporary = leftChest;
leftChest = rightChest;
rightChest = temporary;
System.out.println(leftChest + " / " + rightChest);
```

运行核对

30 / 12

temporary 先保住原来的12。下一课会让我们认识更多类型，并解释为什么整数、小数、真假和文字不能随意混装。