

第37课

打造自定义物品

给矿工罗盘写上可靠的身份标签



名字给玩家看，身份数据给程序认。制作带说明的原版罗盘，用材料、命名空间、类型和值四项检查识别它，再让主手右键只显示一次任务。

前置背包：命令、事件、空值检查与List。新项目BlockAcademyL37沿用第31课构建脚本，继续锁定Java21和普通Paper1.21.11。

阅读方式：先猜结果，再运行核对；每次只改一个条件。

环境：JDK 21、普通 Paper 1.21.11、同版本客户端。服务器插件由 Paper 加载；只有明确标注的规则测试使用 main。

开始前：先认清运行位置

前30课打好了普通 Java 程序的基础。这一阶段把代码接到 Paper 服务器：Minecraft 客户端负责进入世界，服务器加载插件，事件和命令把玩家动作交给你的方法。

本书固定使用 JDK 21 与 Paper 1.21.11，客户端版本也保持一致。下载页面上的最新版本可能已经变了，按版本号选择。本书采用普通 plugin.yml 与 Bukkit API；本阶段没有声称支持 Folia。

图框中的短例是局部片段，先读本节的位置说明，不要各自保存成插件。完整文件会明确给出文件名；同一文件分成多段时，按编号连接，包名、import 与外层类只出现一次。

- 预测：先写成功条件和预期消息。
- 运行：构建成功后，在专用本机测试服安装并重启。
- 核对：普通玩家、管理身份与控制台分别走对应路径。

场景插画帮助理解故事；图解解释真实规则，不作为 IDE 或游戏实测截图。做 Java 修改后重新构建、停服替换新 JAR，再启动；不要把旧 JAR 的表现当作新代码结果。

01 物品本体和说明书分开改

普通罗盘在铁砧里改名“超级矿工罗盘”，就来申请特殊能力。门卫看了看名字差点放行。显示名能换，可靠识别要查看插件写入的身份标记。

ItemStack 是物品，ItemMeta 是说明和附加数据。制作时写入 NamespacedKey 对应的 PDC 标记；识别时检查材料、数据类型和值。名字负责让玩家读懂，标记负责让程序认对。

物品本体和说明书分开改

ItemStack

物品本体

ItemMeta

名称说明与附加数据

setItemMeta

把修改写回本体

本课并不换贴图：这是带自定义元数据的原版罗盘，外观仍使用游戏的罗盘纹理；不会自动增加材质包，也不会改变指针目标。

先制作可读的罗盘 • 位置见本节说明

```
ItemStack item = new ItemStack(Material.COMPASS);
var meta = item.getItemMeta();
meta.displayName(Component.text("矿工罗盘"));
meta.lore(List.of(Component.text("主手右键查看课堂任务")));
item.setItemMeta(meta);
```

故障实验：下面故意保留错误，仅在练习副本中观察。不要把它合并进可用插件。

故意漏掉最后一步 • 故意错误，先预测再修复

```
// 若忘了setItemMeta，检查名称和说明是否真的保存。
```

02 铁砧能改名 不能证明身份

把普通罗盘改名为“矿工罗盘”，看起来像真的，但名字只是标签纸。程序要检查藏在物品数据里的PDC标记。显示名负责让玩家认得，数据键负责让程序认得。

铁砧能改名 不能证明身份

显示名称

方便玩家理解

NamespacedKey

稳定的数据位置

BYTE值1

本章身份标记

不是防伪保险箱：PDC不是秘密，也不是防作弊加密。管理员或其他插件仍可能复制和修改物品数据；它解决的是可靠识别，不是所有安全问题。

为本插件创建一个键 • 位置见本节说明

```
compassKey = new NamespacedKey(this, "miner_compass");
```

写入类型明确的标记 • 位置见本节说明

```
meta.getPersistentDataContainer().set(
    compassKey, PersistentDataType.BYTE, (byte) 1);
```

核对表

信息	用途
显示名称与lore	让玩家看懂
NamespacedKey	命名空间加键名，减少冲突
BYTE类型和值1	本课约定的罗盘身份

03 把制作和识别放进同一个类

新建CompassItems.java，放进de.teacherma.lesson37包。制作和识别使用同一个键与同一种类型，避免一边写miner_compass，另一边却找mining_compass，像把钥匙送错了门。

把制作和识别放进同一个类



名字可以变：铁砧改名后，只要材料和标记仍满足本课条件，程序仍会识别它。普通罗盘仅改成同名不会获得PDC标记。

完整CompassItems.java • 完整文件

```

package de.teacherma.lesson37;

import java.util.List;
import net.kyori.adventure.text.Component;
import org.bukkit.Material;
import org.bukkit.NamespacedKey;
import org.bukkit.inventory.ItemStack;
import org.bukkit.persistence.PersistentDataType;

public final class CompassItems {
    private CompassItems() { }
    public static ItemStack create(NamespacedKey key) {
        ItemStack item = new ItemStack(Material.COMPASS);
        var meta = item.getItemMeta();
        meta.displayName(Component.text("矿工罗盘"));
        meta.lore(List.of(Component.text("主手右键查看课堂任务")));
        meta.getPersistentDataContainer().set(key, PersistentDataType.BYTE, (byte) 1);
        item.setItemMeta(meta);
        return item;
    }
    public static boolean isOurs(ItemStack item, NamespacedKey key) {
        if (item == null || item.getType() != Material.COMPASS || !item.hasItemMeta())
            return false;
        Byte marker = item.getItemMeta().getPersistentDataContainer()
            .get(key, PersistentDataType.BYTE);
        return Byte.valueOf((byte) 1).equals(marker);
    }
}
  
```

04 检查材料 类型和值

空手时item可能为null。普通罗盘可能没有meta；有meta也未必有我们的标记。识别函数按顺序排除不匹配的情况，最后比较BYTE值是不是1。只检查“键存在”，会把别的标记值也收进来。

检查材料 类型和值

材料COMPASS

先排除不相关物品

键和类型

读取对应BYTE标记

值等于1

才认作本章罗盘

键也要稳定：插件名参与默认命名空间。发布后随意改plugin.yml的name，旧物品可能不再匹配；改名时需要设计数据迁移。

安全比较可能为空的值 • 位置见本节说明

```
Byte marker = meta.getPersistentDataContainer()
    .get(compassKey, PersistentDataType.BYTE);
boolean ours = Byte.valueOf((byte) 1).equals(marker);
```

删除自定义身份的片段 • 位置见本节说明

```
meta.getPersistentDataContainer().remove(compassKey);
item.setItemMeta(meta);
```

核对表

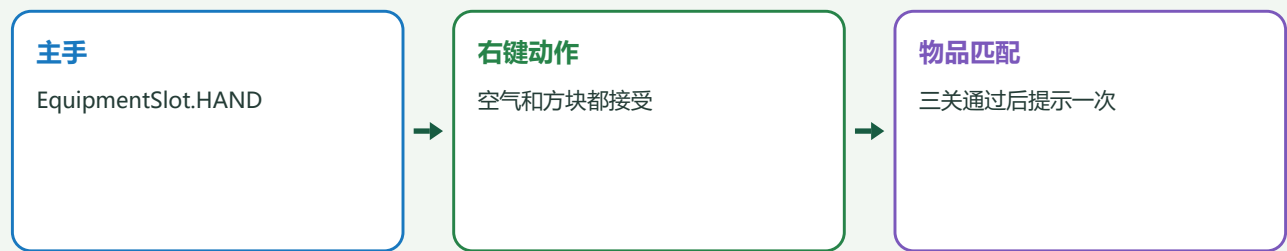
物品数据	识别结果
普通罗盘或无标记	false
同键STRING文字"1"	false
同键BYTE值2	false
COMPASS加同键BYTE值1	true

05 两只手不要抢着汇报

交互事件可能分别为两只手触发。若副手也拿着罗盘，同一动作就可能出现两条任务提示。我们只接受 EquipmentSlot.HAND，也就是主手；不是靠“发消息后删一条”补救。

停下来想一想：真罗盘被玩家改名后还该识别吗？普通罗盘改成同名后应该获得能力吗？

两只手不要抢着汇报



沿箭头进入下一步前，必须满足前一步条件；被拒绝的分支提前返回。

想通这一步：只要材料和PDC标记仍符合规则，真罗盘仍能识别；只有同名、没有标记的普通罗盘不能获得能力。

先筛选主手 · 位置见本节说明

```
if (event.getHand() != EquipmentSlot.HAND) return;
```

两种右键都允许 · 位置见本节说明

```
Action action = event.getAction();
if (action != Action.RIGHT_CLICK_AIR
    && action != Action.RIGHT_CLICK_BLOCK) return;
```

- 主手持持有本课罗盘右键空气：显示一次任务。
- 主手右键方块：显示任务，方块原有交互仍可继续。
- 副手持持有罗盘或左键挥动：本课不显示任务。

06 读任务不代表批准交互

对空气右键时，原版可能预测“没有动作”而把事件标为取消。因此这个只读提示监听器没有设置 `ignoreCancelled=true`。它只看信息并发提示，不恢复取消，也不授权方块操作。

读任务不代表批准交互

只读观察

只发送任务文字

原有交互

不被本章取消或恢复

未来有副作用

须重新设计许可策略

边界说清楚：即使其他插件禁止交互，本课也可能显示只读任务文字。如果未来右键会传送、扣费或改方块，需要另行检查权限、保护和对应交互结果，不能直接照搬这个策略。

只读观察方法的入口 • 位置见本节说明

```
@EventHandler(priority = EventPriority.MONITOR)
public void onUse(PlayerInteractEvent event) {
    // 主手、动作、物品都匹配后，只发送任务文字。
}
```

故障实验：下面故意保留错误，仅在练习副本中观察。不要把它合并进可用插件。

不要为提示解封保护 • 故意错误，先预测再修复

```
event.setCancelled(false); // 本课不要这样做
```

07 给罗盘项目登记命令

项目名设为BlockAcademyL37。resources目录只需要plugin.yml，本课没有config.yml。主类和物品工具类都在lesson37包，构建脚本沿用第31课。

给罗盘项目登记命令

CompassItems

制作与识别共用规则

CompassPlugin

登记命令与监听器

plugin.yml

声明领取命令和权限

领取与使用分开：默认仅op可领取，但拿到罗盘的玩家都能查看只读任务。若需要绑定所有者，可在进阶任务中另存UUID并验证。

完整plugin.yml • 完整文件

```
name: BlockAcademyL37
version: 1.0.0
main: de.teacherma.lesson37.CompassPlugin
api-version: '1.21.11'
author: TeacherMa
website: https://www.teacher-ma.de
commands:
  minercompass:
    description: 领取课堂矿工罗盘
    usage: /minercompass
permissions:
  blockacademy.compass.give:
    default: op
```

完整项目中的Java文件 • 完整文件

```
src/main/java/de/teacherma/lesson37/
  CompassItems.java
  CompassPlugin.java
```

08 罗盘主类上半部

新建CompassPlugin.java。本页先声明依赖、创建一次键、注册命令和监听器；下一页继续命令与交互逻辑。不要在两页中间加一个新的class。

罗盘主类上半部

启用时

创建稳定键

命令登记

领取入口可找到

监听登记

主手右键能收到通知

少做重复工作：NamespacedKey在启用时创建后重复使用。真正交互发生时，只需要读取并比较，不必重新发明一套身份规则。

主类第1部分 • 顺序接在同一源文件

```
package de.teacherma.lesson37;

import java.util.Objects;
import net.kyori.adventure.text.Component;
import org.bukkit.NamespacedKey;
import org.bukkit.command.Command;
import org.bukkit.command.CommandSender;
import org.bukkit.entity.Player;
import org.bukkit.event.EventHandler;
import org.bukkit.event.EventPriority;
import org.bukkit.event.Listener;
import org.bukkit.event.block.Action;
import org.bukkit.event.player.PlayerInteractEvent;
import org.bukkit.inventory.EquipmentSlot;
import org.bukkit.plugin.java.JavaPlugin;

public final class CompassPlugin extends JavaPlugin implements Listener {
    private NamespacedKey compassKey;
    @Override
    public void onEnable() {
        compassKey = new NamespacedKey(this, "miner_compass");
        Objects.requireNonNull(getCommand("minercompass")).setExecutor(this);
        getServer().getPluginManager().registerEvents(this, this);
    }
}
```

09 罗盘主类下半部

紧接上一页。领取前检查发送者、权限与参数；背包装不下的罗盘放到脚边。交互方法只做三次筛选和一次提示，短得足够让你逐行解释。

罗盘主类下半部

领取

先检查身份、权限与参数

剩余

满背包如实处理

观察

不改方块，只发提示

安装验证：运行.\gradlew.bat clean build。停服后把本课JAR放入plugins，重启测试服；用管理员授权的玩家执行/minercompass。

主类第2部分 • 顺序接在同一源文件

```
@Override
public boolean onCommand(CommandSender sender, Command command,
                        String label, String[] args) {
    if (!(sender instanceof Player p)) {
        sender.sendMessage("请在游戏中领取罗盘"); return true;
    }
    if (!p.hasPermission("blockacademy.compass.give")) {
        p.sendMessage("请老师授权后领取"); return true;
    }
    if (args.length != 0) { p.sendMessage("用法: /minercompass"); return true; }
    var left = p.getInventory().addItem(CompassItems.create(compassKey));
    left.values().forEach(item -> p.getWorld().dropItemNaturally(p.getLocation(), item));
    p.sendMessage(left.isEmpty() ? "罗盘已入包，主手右键试试" : "罗盘在脚边");
    return true;
}

@EventHandler(priority = EventPriority.MONITOR)
public void onUse(PlayerInteractEvent event) {
    if (event.getHand() != EquipmentSlot.HAND) return;
    Action action = event.getAction();
    if (action != Action.RIGHT_CLICK_AIR && action != Action.RIGHT_CLICK_BLOCK) return;
    if (!CompassItems.isOurs(event.getItem(), compassKey)) return;
    event.getPlayer().sendMessage(Component.text("课堂任务：准备工具，参加矿石冲刺赛！"));
}
}
```

10 真假罗盘一起参加考试

只测试真罗盘会漏掉大半问题。准备一个普通罗盘、一个改名罗盘和本课罗盘，让“假学生”也来考试。记录预期和实际，不要只在聊天栏看到一句话就宣布毕业。

真假罗盘一起参加考试

真物品

改名后仍可识别

同名假物品

没有PDC就不识别

动作边界

空手、副手和左键不提示

保存的是什么：标签随物品数据保存，不等于插件已保存任务进度。本课显示固定任务文字，没有自动记住你挖了多少矿。

核对表

测试	预期
真罗盘主手右键空气	显示一次任务
真罗盘主手右键箱子	提示出现，箱子交互不被本课改变
同名普通罗盘	不显示任务
空手或左键	不显示任务
真罗盘只放副手	不显示任务
满背包领取	剩余罗盘放脚边并提示
保存物品后正常重启	同名同键插件仍识别标签

11 拓展挑战与自己的验收记录

留下源代码、实际运行结果和一条问题记录。先用自己的话解释，再运行验证；结果与预测不同的地方，往往最值得讨论。

拓展任务与通关复盘

木镐

增加两行lore

铁镐

测试键、类型和值

钻石镐

设计三件不同标签的工具

下一站：第38课把命令、事件、玩家数据和计时器组合成矿石冲刺赛。
官方延伸阅读：docs.papermc.io/paper/dev/pdc/

- 木镐：给罗盘增加两行lore，解释为什么要setItemMeta。
- 铁镐：制作材料、键、类型和值四项识别测试，普通物品不能误触发。
- 钻石镐：设计矿工三件套，用不同标签区分功能；写明领取权限与重启后的行为。
- 验收要求：测试真物品、同名假物品、空手、副手以及改名真物品；一次操作不能因两只手重复响应。

核对表

我能独立完成	已掌握	再练一次
能创建并保存ItemMeta	☐	☐
会读写PDC身份标记	☐	☐
能处理主手与右键动作	☐	☐

把证据留给下一次冒险

把“能编译”“数字规则通过”和“游戏实际表现正确”分成三栏。没有运行的场景写待测；发现 Bug 时记录触发条件、预期、实际、最小修复与复测结果。

你的验证记录

类别	你选择的操作	留下的证据
正常	本课的一次成功路径	版本与实际输出
边界	结束前一刻、最大值或满背包	与合同逐项比较
拒绝	错误参数、无权限或不相关事件	反馈清楚，状态未误改

- 解释本课每条箭头为什么从这里指向那里；并列卡片没有先后含义。
- 先改一个规则，再同时修改说明和预期，用同一输入复测。

技术核对：本课以 Paper 1.21.11 API 和 Java SE 21 为准。参考文档可能切换到较新版本，查阅时注意版本号。

<https://jd.papermc.io/paper/1.21.11/>
<https://docs.papermc.io/paper/dev/>
<https://docs.oracle.com/en/java/javase/21/docs/api/>