

Python程序设计

案例：库存管理系统



张 华

库存管理系统

目的

- 通过一个多模块的库存管理系统案例，深入了解基于模块的Python应用程序的开发流程。

库存管理系统

系统设计

本节实现一个简单的基于模块的库存管理系统。系统采用 JSON 文件来保存数据。产品信息设计为字典，键为 `sku_id` (产品 ID)，值为 `sku_name` (产品名称)，使用 `products.json` 实现其持续化。货架位置信息也设计为字典，键为 `loc_id` (货架 ID)，值为 `loc_name` (货架名称)，使用 `location.json` 实现其持续化。商品库存信息设计为列表 `[sku_id, loc_id]` 的列表，使用 `items.json` 实现其持续化。

库存管理系统

系统设计

库存管理系统设计为三个模块文件：data.py、ui.py 和 main.py。

库存管理系统 data.py 负责数据的管理，其 API 设计如表 CS-1 所示。

表 CS-1 库存管理系统 data.py 模块的 API

全局变量/函数	功能描述
_products	保存产品信息的字典：sku_id:sku_name
_locations	保存货架位置的字典：loc_id:loc_name
_items	保存商品库存的列表，元素为列表[sku_id, loc_id]
init()	从磁盘 JSON 格式文件中读取数据
_save_products()	把产品信息数据_products 以 JSON 格式保存到磁盘文件
_save_locations()	把货架位置数据_locations 以 JSON 格式保存到磁盘文件
_save_items()	把商品库存数据_items 以 JSON 格式保存到磁盘文件
get_products()	返回产品信息
get_locations()	返回货架位置信息
get_items()	返回货架商品信息
add_product(sku_id, sku_name)	增加一个产品 sku_id、sku_name
add_location(loc_id, loc_name)	增加一个货架位置 loc_id、loc_name
add_item(sku_id, loc_id)	入库一件商品：商品 sku_id、货架 sku_id
remove_item(sku_id, loc_id)	出库一件商品：商品 sku_id、货架 sku_id, 返回 True; 如果不存在，返回 False

库存管理系统

系统设计

库存管理系统 ui.py 负责用户界面交互，其 API 设计如表 CS-2 所示。

表 CS-2 库存管理系统 ui.py 模块的 API

全局变量/函数	功能描述
<code>prompt_for_action()</code>	提示功能菜单，返回用户输入选择
<code>prompt_for_old_sku_id()</code>	提示用户输入有效的产品 <code>sku_id</code> 并返回有效的产品 ID。如果用户输入为空，则返回 <code>None</code>
<code>prompt_for_new_sku_id()</code>	提示用户输入新的产品 <code>sku_id</code> 并返回新的产品 ID。如果用户输入为空，则返回 <code>None</code>
<code>prompt_for_sku_name()</code>	提示用户输入产品名称 <code>sku_name</code> 并返回产品名称。如果用户输入为空，则返回 <code>None</code>
<code>prompt_for_old_loc_id()</code>	提示用户输入有效的货架位置 <code>loc_id</code> 并返回有效的货架位置 ID。如果用户输入为空，则返回 <code>None</code>
<code>prompt_for_new_loc_id()</code>	提示用户输入新的货架位置 <code>loc_id</code> 并返回新的货架位置 ID。如果用户输入为空，则返回 <code>None</code>
<code>prompt_for_loc_name()</code>	提示用户输入货架位置名称 <code>loc_name</code> 并返回货架位置名称。如果用户输入为空，则返回 <code>None</code>
<code>report_products()</code>	产品信息报表
<code>report_locations()</code>	货架位置报表
<code>report_items()</code>	库存信息报表

库存管理系统

功能设计

库存信息管理系统主要包括如下功能：

(1) 增加产品信息

调用 `ui.prompt_for_new_sku_id()`，提示用户输入新的产品 ID，调用 `ui.prompt_for_sku_name()`，提示用户输入产品名称。调用 `data.add_product(sku_id, sku_name)` 增加新的产品。如果用户输入为空，则返回 `None`，即什么也不做。

(2) 产品信息报表

调用 `ui.report_products()`，显示产品信息列表。

(3) 增加货架位置

调用 `ui.prompt_for_new_loc_id()`，提示用户输入新的货架位置 ID，调用 `ui.prompt_for_loc_name()`，提示用户输入货架名称。调用 `data.add_location(sku_id, sku_name)` 增加新的货架。如果用户输入为空，则返回 `None`，即什么也不做。

(4) 货架位置报表

调用 `ui.report_location()`，显示货架位置信息列表。

库存管理系统

功能设计

(5) 商品入库管理

调用 `ui.prompt_for_old_sku_id()`, 提示用户输入产品 ID, 调用 `ui.prompt_for_old_loc_id()`, 提示用户输入货架 ID。调用 `data.add_item(sku_id, loc_id)`, 实现商品入库。如果用户输入为空, 则返回 `None`, 即什么也不做。

(6) 商品出库管理

调用 `ui.prompt_for_old_sku_id()`, 提示用户输入产品 ID, 调用 `ui.prompt_for_old_loc_id()`, 提示用户输入货架 ID。调用 `data.remove_item(sku_id, loc_id)`, 实现商品出库。如果库存不存在, 则报错。如果用户输入为空, 则返回 `None`, 即什么也不做。

(7) 商品库存信息报表

调用 `ui.report_items()`, 显示库存信息列表。

库存管理系统

实现

✿ 主模块: **main.py**

主模块导入 data 和 ui 模块。

在 main.py 中, 定义 main() 函数, 首先调用 data.init(), 从磁盘 JSON 格式文件中读取数据。然后在无限循环中, 调用 ui.prompt_for_action() 显示功能菜单, 接受用户输入, 并根据用户的功能选择, 实现各模块相应功能。

✿ 用户界面交互模块: **ui.py**

用户界面交互模块导入 data 模块。实现表 CS-2 所示的 API。

✿ 数据处理模块: **data.py**

数据处理模块实现表 CS-1 所示的 API。通过 Python 标准库模块 json 中的 loads() 函数和 dumps() 函数, 可以实现从 JSON 文件读取数据和转储数据到 JSON 文件的功能。