Public class Item {

Private String itemName;

Private int quantity;

Private double price;

Public Item(String itemName, int quantity, double price) {

This.itemName = itemName;

This.quantity = quantity;

This.price = price;

}

Public String getItemName() {

Return itemName;

}

Public void setItemName(String itemName) {

This.itemName = itemName;

}

Public int getQuantity() {

Return quantity;

}

Public void setQuantity(int quantity) {

This.quantity = quantity;

}

Public double getPrice() {

Return price;

}

Public void setPrice(double price) {

This.price = price;

}

}

Public enum Category {

BEVERAGES,

BREAD\_BAKERY,

CANNED\_JARRED\_GOODS,

DAIRY\_PRODUCTS,

DRY\_BAKING\_GOODS,

FROZEN\_PRODUCTS,

MEAT,

FARM\_PRODUCE,

HOME\_CLEANERS,

PAPER\_GOODS,

HOME\_CARE

}

Import java.util.Stack;

Public class CategoryStack {

Private Stack<Item> itemStack;

Public CategoryStack() {

itemStack = new Stack<>();

}

Public void addItem(Item item) {

itemStack.push(item);

}

Public Item removeItem() {

If (!itemStack.isEmpty()) {

Return itemStack.pop();

}

Return null;

}

Public boolean isEmpty() {

Return itemStack.isEmpty();

}

}

Import java.util.LinkedList;

Import java.util.Queue;

Public class CategoryQueue {

Private Queue<Item> itemQueue;

Public CategoryQueue() {

itemQueue = new LinkedList<>();

}

Public void addItem(Item item) {

itemQueue.add(item);

}

Public Item removeItem() {

Return itemQueue.poll();

}

Public boolean isEmpty() {

Return itemQueue.isEmpty();

}

}

Import java.util.ArrayList;

Import java.util.List;

Public class CategoryList {

Private List<Item> itemList;

Public CategoryList() {

itemList = new ArrayList<>();

}

Public void addItem(Item item) {

itemList.add(item);

}

Public boolean removeItem(Item item) {

Return itemList.remove(item);

}

Public boolean isEmpty() {

Return itemList.isEmpty();

}

}

Import java.util.HashMap;

Import java.util.Map;

Public class VendorManager {

Private Map<String, Vendor> vendorMap;

Public VendorManager() {

vendorMap = new HashMap<>();

}

Public void addVendor(Vendor vendor) {

vendorMap.put(vendor.getName(), vendor);

}

Public Vendor getVendor(String name) {

Return vendorMap.get(name);

}

Public boolean removeVendor(String name) {

Return vendorMap.remove(name) != null;

}

Public boolean isEmpty() {

Return vendorMap.isEmpty();

}

}