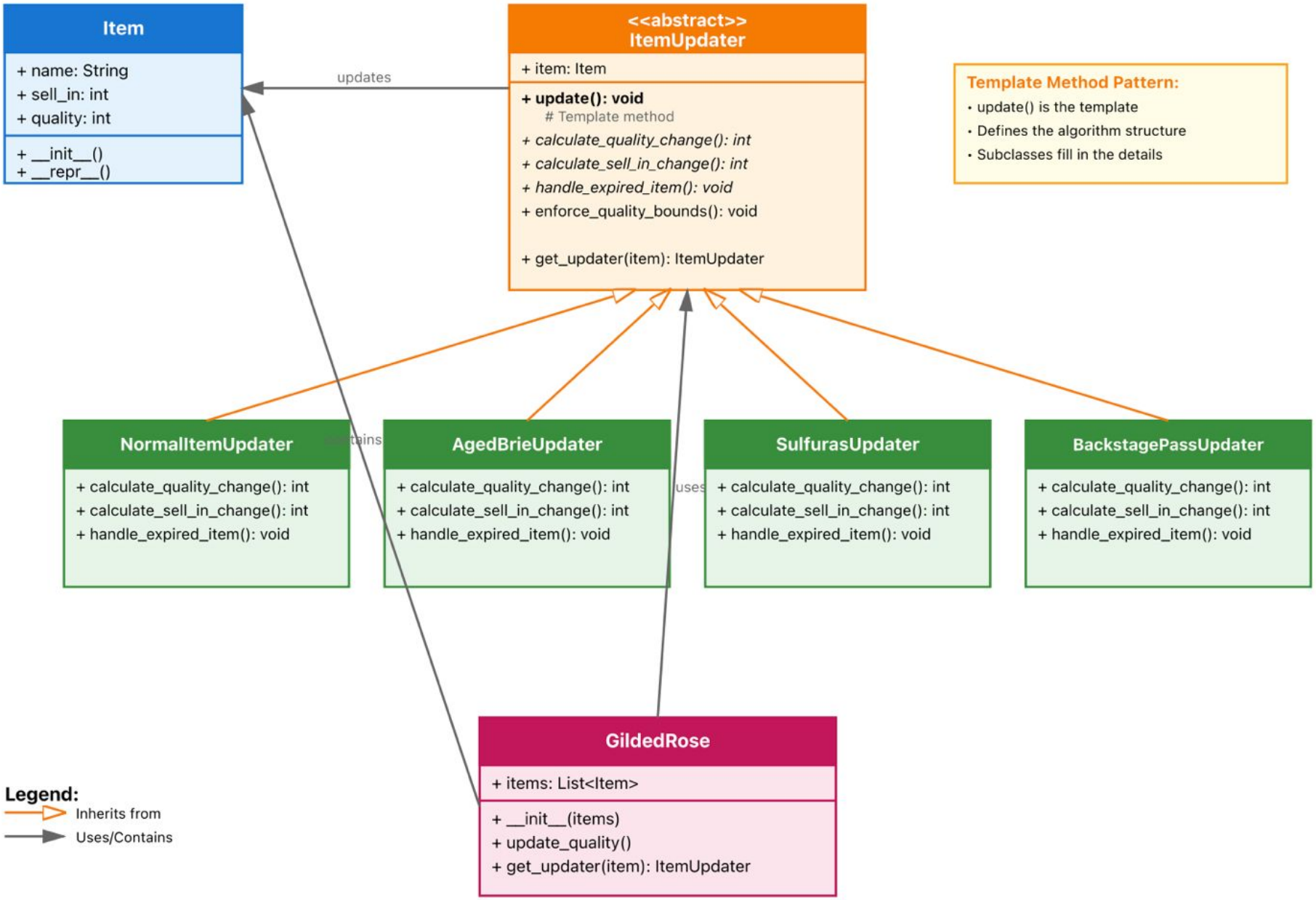
Item
+ name: String
+ sell_in: int
+ quality: int
+ __init__()
+ __repr__()
updates
<<abstract>>
ItemUpdater
+ item: Item
+ update(): void
# Template method
+ calculate_quality_change(): int
+ calculate_sell_in_change(): int
+ handle_expired_item(): void
+ enforce_quality_bounds(): void
+ get_updater(item): ItemUpdater
Template Method Pattern:
• update() is the template
• Defines the algorithm structure
• Subclasses fill in the details
NormalItemUpdater
+ calculate_quality_change(): int
+ calculate_sell_in_change(): int
+ handle_expired_item(): void
AgedBrieUpdater
+ calculate_quality_change(): int
+ calculate_sell_in_change(): int
+ handle_expired_item(): void
SulfurasUpdater
+ calculate_quality_change(): int
+ calculate_sell_in_change(): int
+ handle_expired_item(): void
BackstagePassUpdater
+ calculate_quality_change(): int
+ calculate_sell_in_change(): int
+ handle_expired_item(): void
uses
GildedRose
+ items: List<Item>
+ __init__(items)
+ update_quality()
+ get_updater(item): ItemUpdater
Legend:
Inherits from
Uses/Contains