Design Rationale (REQ 5)

<<interface>> TradableItem implemented by SuperMushroom, PowerStar and Wrench. To follow *open-closed principle*, these three items share the same method but different implementations. By using an interface as an abstraction, it can open for extension of the same method but different implementation.

TradeAction class is created and *Toad---<<creates>>--->TradeAction*. In this action, it simply checks the player's wallet balance whether sufficient to buy the corresponding item. There are only 3 items that can be sold to the player which are wrench, powerstar and supermushroom. Once the player buys a powerstar or supermushroom from Toad, these items cannot be dropped by the player. To achieve this requirement, whenever the player buys one of these two items from Toad, a new instance on these items' class which has the attribute portable=false will be created and added to the player's inventory. The reason for doing this is because any item that is not portable will not allow pick up and drop action.

WalletManager has a list of tradable items that were bought by the player. It is a static class with a private constructor. The reason for doing this is because we only want an instance of WalletManager and the only way to get the instance is using the getInstance() method. WalletManager allows us to keep track with item bought by player and deduct the price of item from its wallet balance