

## # Vending Machine Challenge

Create a console program that imitates a Vending Machine, which receives a standard input and generate a standard output.

### ## Business rules

1. Valid coins are 10 JPY, 50 JPY, 100 JPY, 500 JPY. (Consider that other coins and paper money do not exist)
2. Use 10 JPY and 100 JPY for change.
3. Conditions to determine change possibility:  
Change is impossible if the coins contained in the Vending Machine's vault is in the conditions below:
  - 10 JPY coin: Less than 9 coins
  - 100 JPY coin: Less than 4 coinsIf vending machine runs out of 10 JPY coins, then 50 JPY, 100 JPY, 500 JPY coins cannot be used.  
If vending machine runs out of 100 JPY coins, then 500 JPY coins cannot be used.
4. State of whether or not the items can be purchased is as following
  - Item is out of stock and Item price > Inserted amount -> Cannot purchase [Sold out]
  - Item is out of stock and Item price  $\leq$  Inserted amount -> Cannot purchase [Sold out]
  - Item is in stock and Item price > Inserted amount -> Cannot purchase [Display nothing]
  - Item is in stock and Item price  $\leq$  Inserted amount -> Can purchase [Can purchase]
5. In case that there is any abnormal operation, then System does not response to this operation.  
For example: Return command before inserting coins, Purchase unavailable-for-purchase items, take out items from empty outlet, etc.  
(\* As alternative solution for above situations, output exception message to console is OK)

### ## Use cases

- Start Vending Machine

Pre-conditions: None

Post-conditions: Vending Machine shows current state and in position to accept new inputs.

Scenario:

1. The manager starts the Vending Machine.
2. System counts the money in vault and the goods in-stock.

System displays the state of the Vending Machine and turns into Input accept state.

- Purchase goods

Pre-conditions: Vending Machine shows current state and in position to accept new inputs.

Post-conditions: Vending Machine shows current state and in position to accept new inputs.

Scenario:

1. Customer inserts coins (standard input (Command (1)))
2. System stores the coins in provisional socket. System calculates the total inserted amount, and determines whether items can be purchased. (State of Vending Machine is displayed, being in Input accept state.)
3. Customer instructs System the items for purchase (Push the Purchase button) (Standard input (Command (2)))

- 4. System delivers specified items to the outlet. System deducts the amount of purchase from the total inserted amount.(State of Vending Machine is displayed, being in Input accept state.)
- 5. Customer get items from the outlet (Standard input (Command (3)))
- 6. System empties the outlet. (State of Vending Machine is displayed, being in Input accept state.)
- 7. Customer instructs System to return the change (Pull the return lever) (Standard input (Command (4)))
- 8. System stores the coins in provisional socket to the vault, and returns the change.(State of Vending Machine is displayed, being in Input accept state.)
- 9. Customer gets the coins from Return gate (Standard input (Command (5)))
- 10. System empties the Return gate.(State of Vending Machine is displayed, being in Input accept state.)

Additional Notes:

- s-2 In case change is not possible, just return the inserted coins as it is.
- s-3 If the total inserted amount is not enough to purchase items, then return to [1] and perform coin inserting.
- s-4 If unavailable-item is chosen for-purchase, then System does nothing.
- s-5,7 In case of continuous purchase of items, return to Coin inserting [1] or Purchase item instruction [3].
- s-1,3 If the purchase items process is interrupted(gave up), then go to Return command [7].
- s-8 In case that no item is purchased then just return the coins in the provisional socket as it is.

## Standard output image of Vending Machine state

1. Initial state, State after retrieving product and change

-----				
[Input amount]	0 JPY			
[Change]	100 JPY	No change		
	10 JPY	Change		
[Return gate]	Empty			
[Items for sale]	1. Canned coffee	120 JPY		
	2. Water PET bottle	100 JPY	Sold out	
	3. Sport drinks	150 JPY		
[Outlet]	Empty			
-----				

2. Goods purchasing state

-----				
[Input amount]	130 JPY			
[Change]	100 JPY	No change		
	10 JPY	Change		
[Return gate]	Empty			
[Items for sale]	1. Canned coffee	120 JPY	Available for purchase	
	2. Water PET bottle	100 JPY	Sold out	
	3. Sport drinks	150 JPY		
[Outlet]	Canned coffee			
-----				

3. After goods purchasing, returning lever was pulled state

```
-----
[Input amount]      0 JPY
[Change]            100 JPY      No change
                   10 JPY      Change
[Return gate]       100 JPY
                   10 JPY
                   10 JPY
                   10 JPY
[Items for sale]    1. Canned coffee      120 JPY
                   2. Water PET bottle    100 JPY      Sold out
                   3. Sport drinks        150 JPY
[Outlet]            Canned coffee
                   Canned coffee
                   Water PET bottle
-----
```

## ## Standard input method

Input method

Input [command number + space + arguments] into prompt

For example: "1 500" (Insert 500 JPY coin by Command number (1))

\*Console image. In fact, execute from Eclipse(Editor) is OK

Command (1)

```
Command name      : Insert coins
Command number    : 1
Argument          : int, coin types (any of 10, 50, 100, 500)
```

For example: "1 500" (Insert 500 JPY coin)

Command (2)

```
Command name      : Choose item to purchase
Command number    : 2
Argument          : int, item types (any of item numbers)
```

For example: "2 1" (1: Choose Canned coffee)

Command (3)

```
Command name      : Get items
Command number    : 3
Argument          : None
```

For example: "3" (Get items)

Command (4)

Command name : Return coins

Command number : 4

Argument : None

For example: "4" (Pull Return lever)

Command (5)

Command name : Get returned coins

Command number : 5

Argument : None

For example: "5" (Get returned coins)