



AFS

Agile Full Stack
Developer Bootcamp
Thoughtworks @

Parking Lot

/thoughtworks

© 2022 Thoughtworks Commercial in Confidence

ooCL
We take it personally



AFS

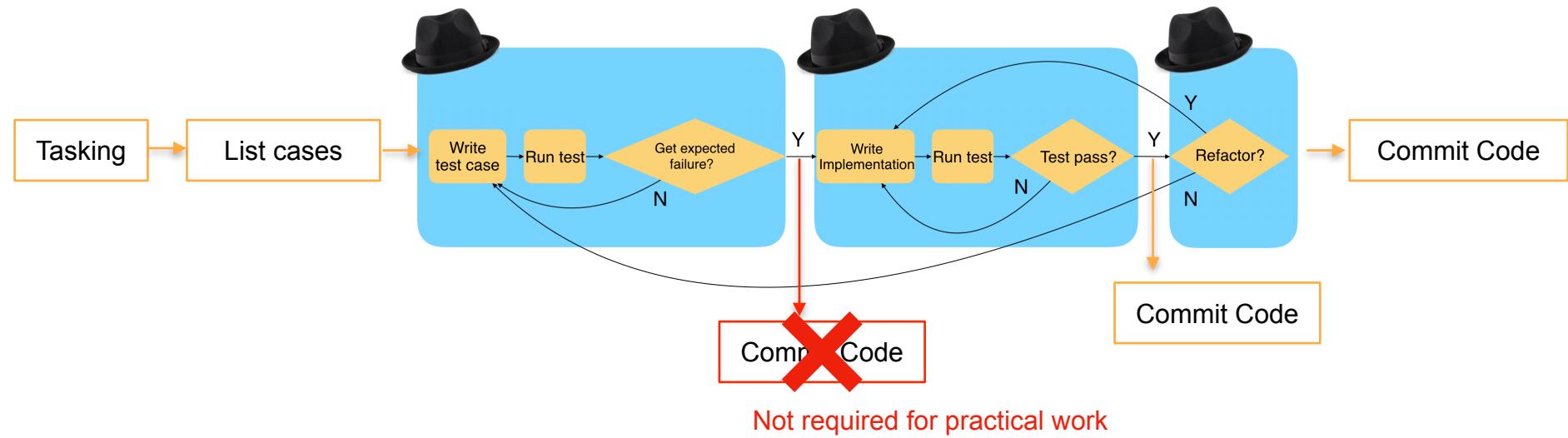
Agile Full Stack
Developer Bootcamp
Thoughtworks @



TDD Review

/thoughtworks

TDD Review





AFS

Agile Full Stack
Developer Bootcamp
Thoughtworks @



Parking Lot Story 1

/thoughtworks

Story 1: Parking Lot

As a parking lot manager, I would like to provide a parking lot, so that the customer can park his/her car and fetch it.

AC1: A customer can park a car into the parking lot and receive a parking ticket. The customer can fetch his/her car with the ticket from the parking lot.

AC2: Every customer can park his/her car into the parking lot. And can fetch right car using correspond ticket.

AC3: If the customer fetches car with a wrong ticket (the parking lot did not provide the ticket) or without a ticket. Then no car should be fetched.

AC4: If the customer fetches car with a ticket that has already been used. Then no car should be fetched.

AC5: The parking lot has a capacity (the default capacity of a parking lot is 10). If there is no position, then the customer cannot park the car into it. Thus (s)he will not get any ticket.

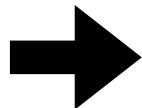
There are some cases which are not a requirement but may happen technically

- Park a parked car to a parking lot.
- Park a null car to a parking lot.
- Fetch a car with a null ticket from a parking lot.

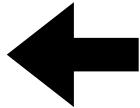
Story 1 - Park a car



Car



ParkingTicket



ParkingLot

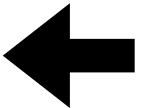
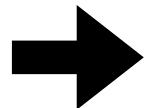
Story 1 - Fetch a car



ParkingTicket



Car



ParkingLot

Story 1 - List Cases

- Case 1 - **Given** a parking lot, and a car, **When** park the car, **Then** return a parking ticket.
- ...

Demo

Story 1 - List Cases

Practice

List all cases for the ACs

Story 1 - List Cases

- Case 1 - **Given** a parking lot, and a car, **When** park the car, **Then** return a parking ticket.
- Case 2 - **Given** a parking lot with a parked car, and a parking ticket, **When** fetch the car, **Then** return the parked car.
- Case 3 - **Given** a parking lot with two parked cars, and two parking tickets, **When** fetch the car twice, **Then** return the right car with each ticket
- Case 4 - **Given** a parking lot, and a wrong parking ticket, **When** fetch the car, **Then** return nothing.
- Case 5 - **Given** a parking lot, and a used parking ticket, **When** fetch the car, **Then** return nothing.
- Case 6 - **Given** a parking lot without any position, and a car, **When** park the car, **Then** return nothing

Story 1 - TDD

Practice

Implement all cases using TDD



AFS

Agile Full Stack
Developer Bootcamp
Thoughtworks @



Parking Lot Story 2

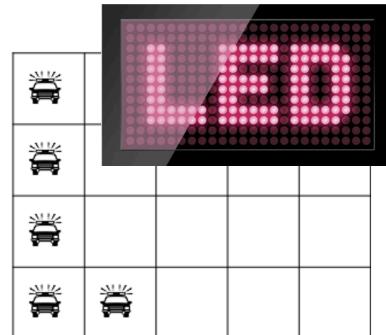
/thoughtworks

Story 2: Reminder Message

As a parking lot manager, I would like to give some friendly reminder message when a customer cannot park/fetch a car. So that the customer can know what happens.

AC1: When the customer fetches car with a wrong ticket (unrecognized ticket / the ticket has been used). Then no car should be fetched, with error message "**Unrecognized parking ticket.**".

AC2: When the customer attempt to park a car into a parking lot without any position. The error message should be "**No available position.**"



Story 2 - List Cases

Practice

List all cases for the ACs

Story 2 - List Cases

- Case 1 - **Given** a parking lot, and an unrecognized ticket, **When** fetch the car, **Then** return nothing with error message "Unrecognized parking ticket."
- Case 2 - **Given** a parking lot, and a used ticket, **When** fetch the car, **Then** return nothing with error message "Unrecognized parking ticket."
- Case 3 - **Given** a parking lot without any position, and a car, **When** park the car, **Then** return nothing with error message "No available position."

Story 2 - TDD

Practice

Implement all cases using TDD



AFS

Agile Full Stack
Developer Bootcamp
Thoughtworks @



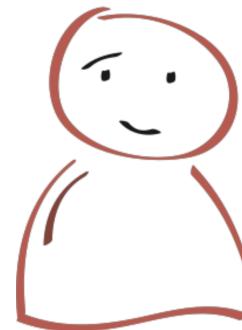
Parking Lot Story 3

/thoughtworks

Story 3: Parking Boy

As a parking lots manager, I would like to provide a standard parking boy, so that he can help the customer park and fetch the car.

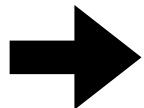
AC1: All the requirement in *Story 1 and Story 2* **MUST** be satisfied by the standard parking boy.



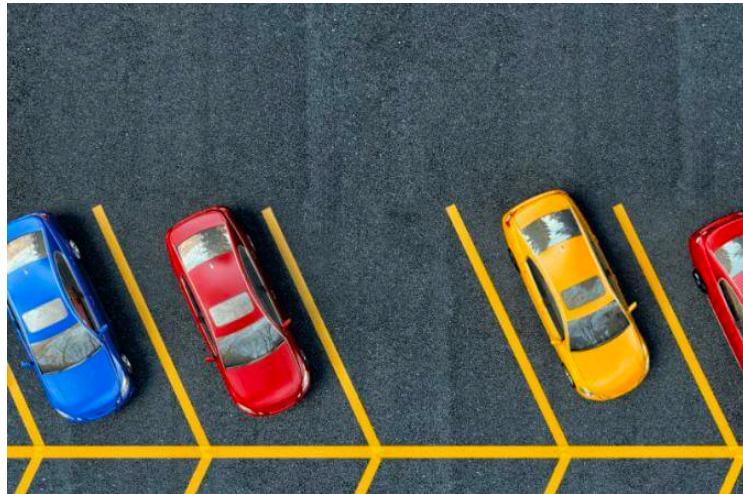
Story 3 - Park a car



Car



ParkingBoy



ParkingLot



ParkingTicket

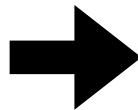
Story 3 - Fetch a car



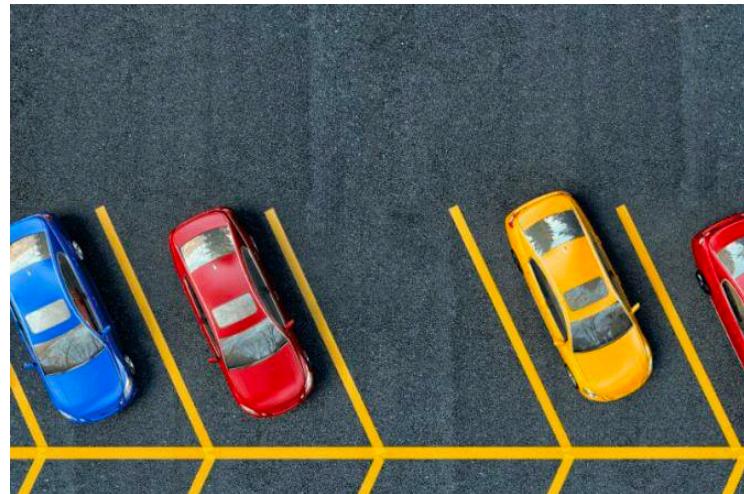
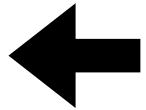
ParkingTicket



Car



ParkingBoy



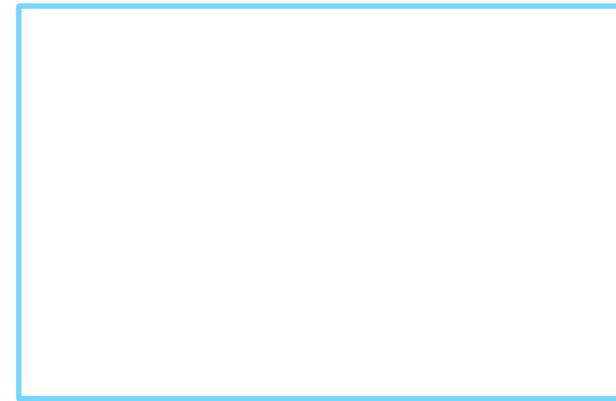
ParkingLot

Story 3: OOP

What is the relationship between ParkingBoy and ParkingLot?

IS-A

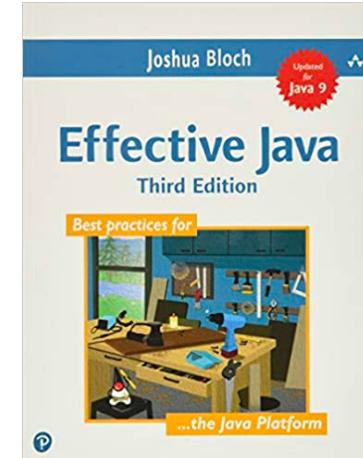
HAS-A



Story 3: OOP

Favor composition over inheritance

From *Effective Java*



Story 3 - List Cases

Practice

List all cases for the ACs

Story 3 - List Cases

- Case 1 - **Given** a parking lot, a standard parking boy, and a car, **When** park the car, **Then** return a parking ticket.
- Case 2 - **Given** a parking lot with a parked car, a standard parking boy, and a parking ticket, **When** fetch the car, **Then** return the parked car.
- Case 3 - **Given** a parking lot with two parked cars, a standard parking boy, and two parking tickets, **When** fetch the car twice, **Then** return the right car with each ticket
- Case 4 - **Given** a parking lot, a standard parking boy, and a wrong parking ticket, **When** fetch the car, **Then** return nothing with error message "Unrecognized parking ticket."
- Case 5 - **Given** a parking lot, a standard parking boy, and a used parking ticket, **When** fetch the car, **Then** return nothing with error message "Unrecognized parking ticket."
- Case 6 - **Given** a parking lot without any position, a standard parking boy, and a car, **When** park the car, **Then** return nothing with error message "No available position."

Story 3 - TDD

Practice

Implement all cases using TDD



AFS

Agile Full Stack
Developer Bootcamp
Thoughtworks @



Parking Lot Story 4

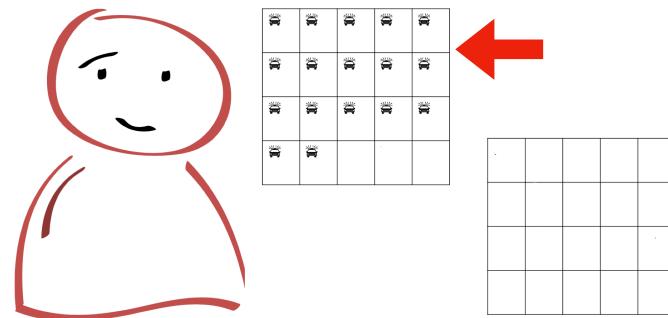
/thoughtworks

Story 4: Multiple Parking Lots

As a parking lots service manager, I would like to have a standard parking boy parking cars to multiple parking lots. So that I can provide more parking positions.

AC1. The standard parking boy is not that clever, and he will always park cars sequentially (suppose that there are two parking lots managed by the standard parking boy. The standard parking boy will park cars to the second parking lot when the first parking lot is full).

AC2: All the requirement in *Story 1-3* **MUST** be satisfied.



Story 4 - List Cases

- Case 1 - **Given** a standard parking boy, who manage two parking lots, both with available position, and a car, **When** park the car, **Then** the car will be parked to the first parking lot
- Case 2 - **Given** a standard parking boy, who manage two parking lots, first is full and second with available position, and a car, **When** park the car, **Then** the car will be parked to the second parking lot
- Case 3 - **Given** a standard parking boy, who manage two parking lots, both with a parked car, and two parking ticket, **When** fetch the car twice, **Then** return the right car with each ticket
- Case 4 - **Given** a standard parking boy, who manage two parking lots, and an unrecognized ticket, **When** fetch the car, **Then** return nothing with error message "Unrecognized parking ticket."
- Case 5 - **Given** a standard parking boy, who manage two parking lots, and a used ticket, **When** fetch the car, **Then** return nothing with error message "Unrecognized parking ticket."
- Case 6 - **Given** a standard parking boy, who manage two parking lots, both without any position, and a car, **When** park the car, **Then** return nothing with error message "No available position."



AFS

Agile Full Stack
Developer Bootcamp
Thoughtworks @



Parking Lot Story 5

/thoughtworks

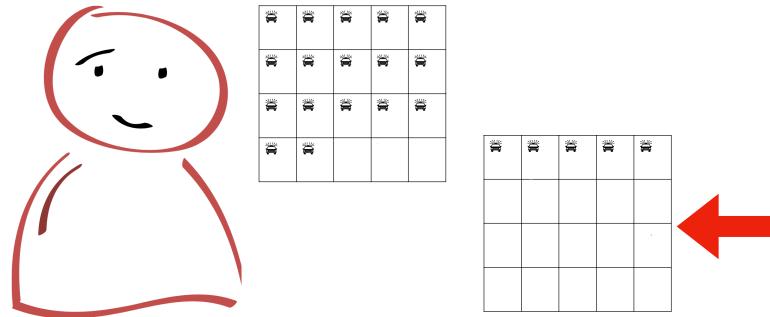
Story 5: Smart Parking Boy

As a parking lots service manager. I would like to have another kind of parking boy to help me parking cars to multiple parking lots. So that the parking positions can be better used.

The new kind of parking boy is called **smart parking boy**.

AC1. The smart parking boy will always park cars to the parking lot which contains more empty positions.

AC2: All the requirement in *Story 1-3* **MUST** be satisfied.





AFS

Agile Full Stack
Developer Bootcamp
Thoughtworks @

ooCL
We take it personally

Parking Lot Story 6(optional)

/thoughtworks

Story 6 - Strategy Pattern

The number of parking boy classes continues to expand.

Please try to apply strategy pattern to reduce the number of parking boy classes



AFS

Agile Full Stack
Developer Bootcamp
Thoughtworks @



Homework

/thoughtworks

Homework

1. Diary using ORID(finish on Friday)

2. Parking Lot

- ❖ Complete Story 1-5 using TDD
- ❖ Complete Story 6(Optional)
- ❖ Baby step commit, commit message format is **type: detail message**
- ❖ Submit your homework with your GitHub repository address



Dec 4 Sunday
16:00pm

Extra points:

- No obvious access modifier usage errors, e.g., no incorrect use of private, public
- No obvious duplicate code
- Use Java stream instead of for loop
- Use meaningful names for variables, methods



AFS

Agile Full Stack
Developer Bootcamp
Thoughtworks @



Q & A

/thoughtworks