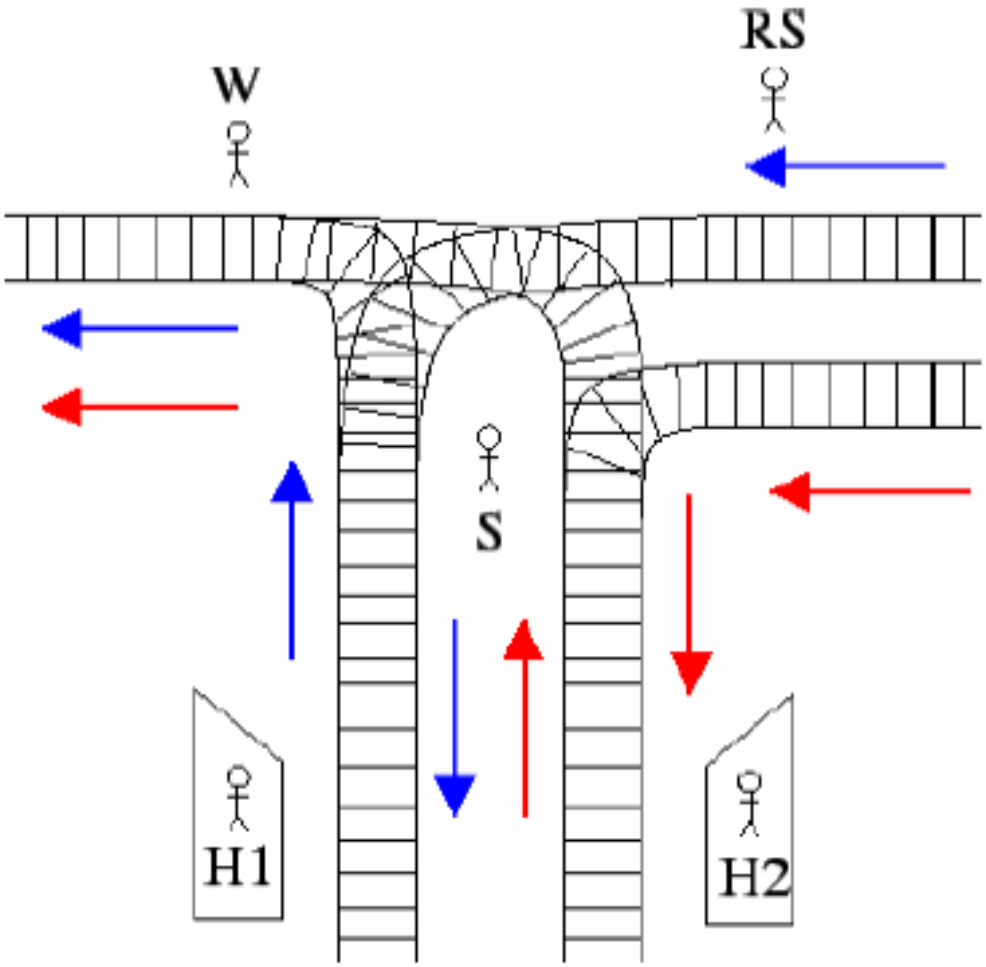
Railway Station Simulation

Object-oriented Design

Railway Station



- RS (rooftop signal man)
- S (station master)
- W (watchman)
- H (Helper): H1 or H2

기차 종류

- 승객용 기차
  - A칸 5명 정원
  - B칸 10명 정원
  - C칸 50명 정원
- 화물용 기차
  - C칸 10개 (covered)
  - O칸 10개 (open)

플랫폼과 트랙

- 승객용 플랫폼, 승객용 정차 트랙
- 화물용 플랫폼, 화물용 정차 트랙
  - 동시에 승객용 기차와 화물용 기차가 정차가능
- 출발 트랙은 공유
  - 동시에 두 기차가 출발할 수 없음
  - 동시에 출발 시 승객용 기차가 우선권이 있음

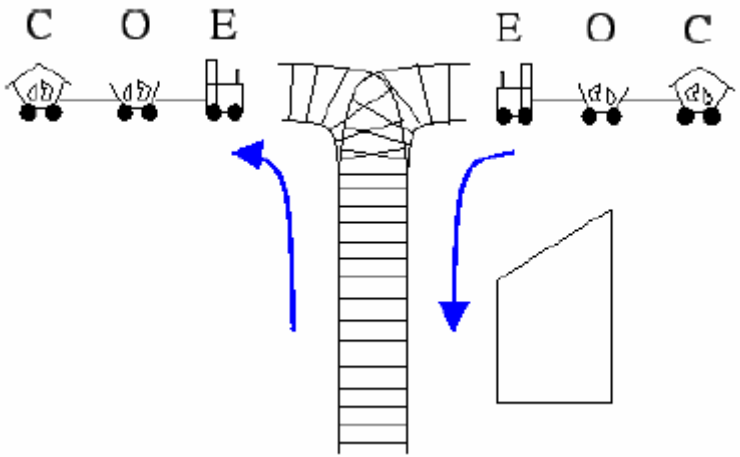
Message Rules 1/2

1. 진입하는 기차가 있을 때 RS는 S에게 허가를 요청한다
2. S는 기차의 종류에 따라 track이 비었는지 확인하여 RS의 요청을 수락한다
3. 허가가 나면 기차는 정차시키고 아니면 통과시킨다(무시)
4. S는 RS의 요청을 수락했을 경우 해당 기차에 해당하는 H에게 짐을 내릴 것을 요청한다
5. H는 Track의 짐을 모두 내렸을 경우 S에게 출발이 준비 되었음을 알린다 (짐을 내리는 데 필요한 시간은 기차에 따라 다르며 이는 input에서 출발 시간으로 주어진다).

Message Rules 2/2

6. S는 H로부터 출발 요청을 받은 후 W에게 출발 허가를 요청한다
7. W는 이전 기차가 출발한 후 5분이 지난 후부터 다음 기차의 출발을 허가한다
8. S가 W로부터 출발허가를 받으면 기차는 출발을 하며, 해당 트랙은 1분 후부터 다음 기차를 받아 들일 수 있다
9. S가 W로부터 출발을 거부 당하면 허가를 받을 때 까지 매 1분마다 재시도 한다

Train Ordering



기차는 출발할 때 그 순서가 뒤바뀐다

Input

- T 도착시간 출발예정시간 식별자 기차타입 엔진 [차량종류 승차인원 하차인원]*

- T 0915 0922 TRAIN_A P E A 3 2 A 2 0 B 6 6
- T 1115 1125 TRAIN_B G E O 8 5 C 8 8
- T 0910 0919 TRAIN_C G E C 3 0
- T 1112 1120 TRAIN_D G E 0 9 9

Outputs – Timing Information

- TRAIN_C : 0910 : arrived
- TRAIN_C : 0910 : 13 goods were downloaded.
- GOODS TRACK : 0910 : OCCUPIED
- TRAIN_A : 0915 : arrived
- TRAIN_A : 0915 : 28 passenger were downloaded.
- PASSANGER TRACK : 0915 : OCCUPIED
- TRAIN_C : 0919 : leaved
- PASSENGER TRACK : 0920 : EMPTY

Output – Statistics information

- ?? trains were serviced.
- ?? passenger trains were serviced.
- ?? goods trains were serviced.
- ?? passengers got off at this station.
- ?? units of goods were downloaded at this station.
- ?? passengers used this station as an intermediate point.
- ?? units of goods passed through this station for a future destination.

Output – Statistics information

- ?? trains were refused entry because of scheduling conflicts. Of these, there were ?? passenger trains and ?? goods trains. This means a potential loss of ?? people not coming into town, and ?? units of goods not coming into town.

  각 칸의 최대 정원으로 계산 해야 함!!

Potential classes

- Officer
  - rooftop signal man, station master, helpers, (H1/H2), watchman
- Train
- Railroad car
- Station, Platforms, Tracks
- Passenger, Goods

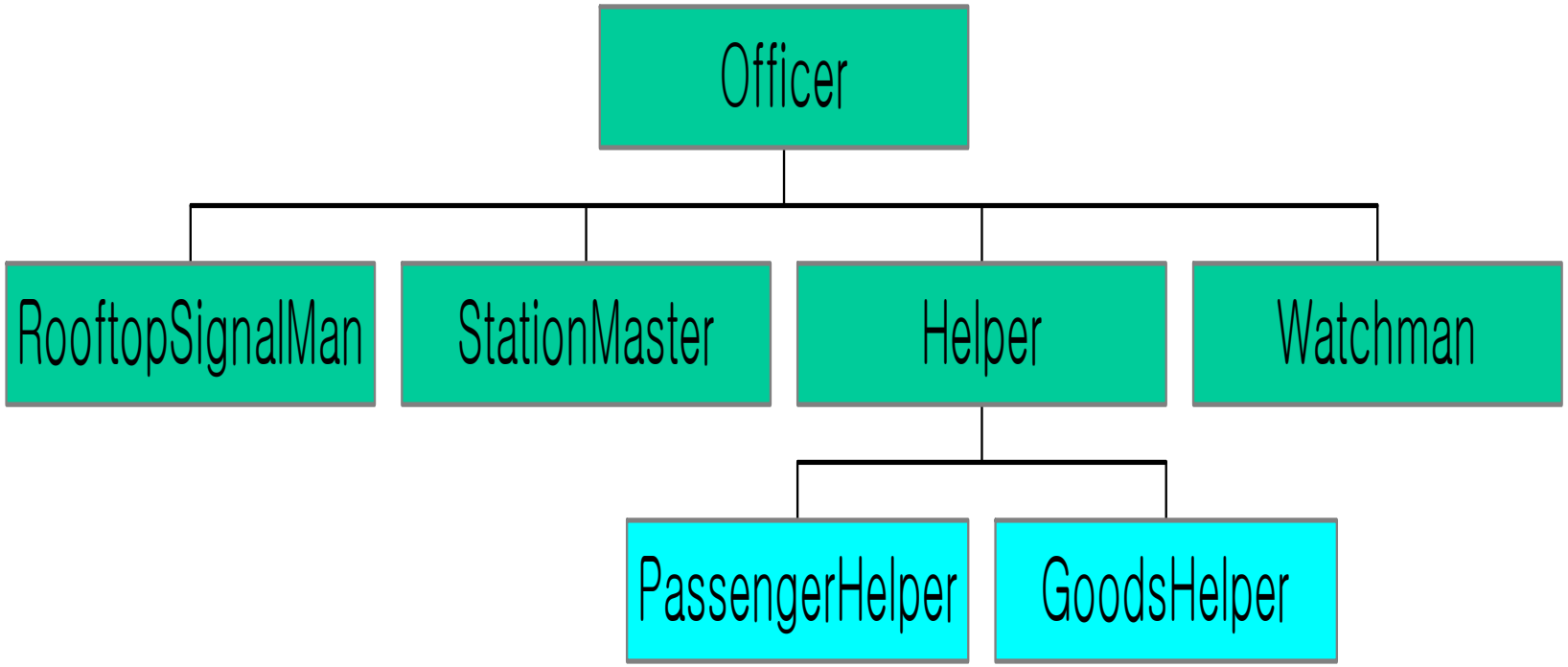
Actions on and from officers

- Rooftop signal man (RS)
  - Receive train
  - Send a message to SM
- Station master (SM)
  - Receive train and dispatch to platform
  - Grant permission for train leaving platform

Actions on officers (cont.)

- Helpers (H)
  - Download contents of train
  - Let out train
- Watchman (WM)
  - Let out train
- Common operations
  - Receive a message
  - Send a message

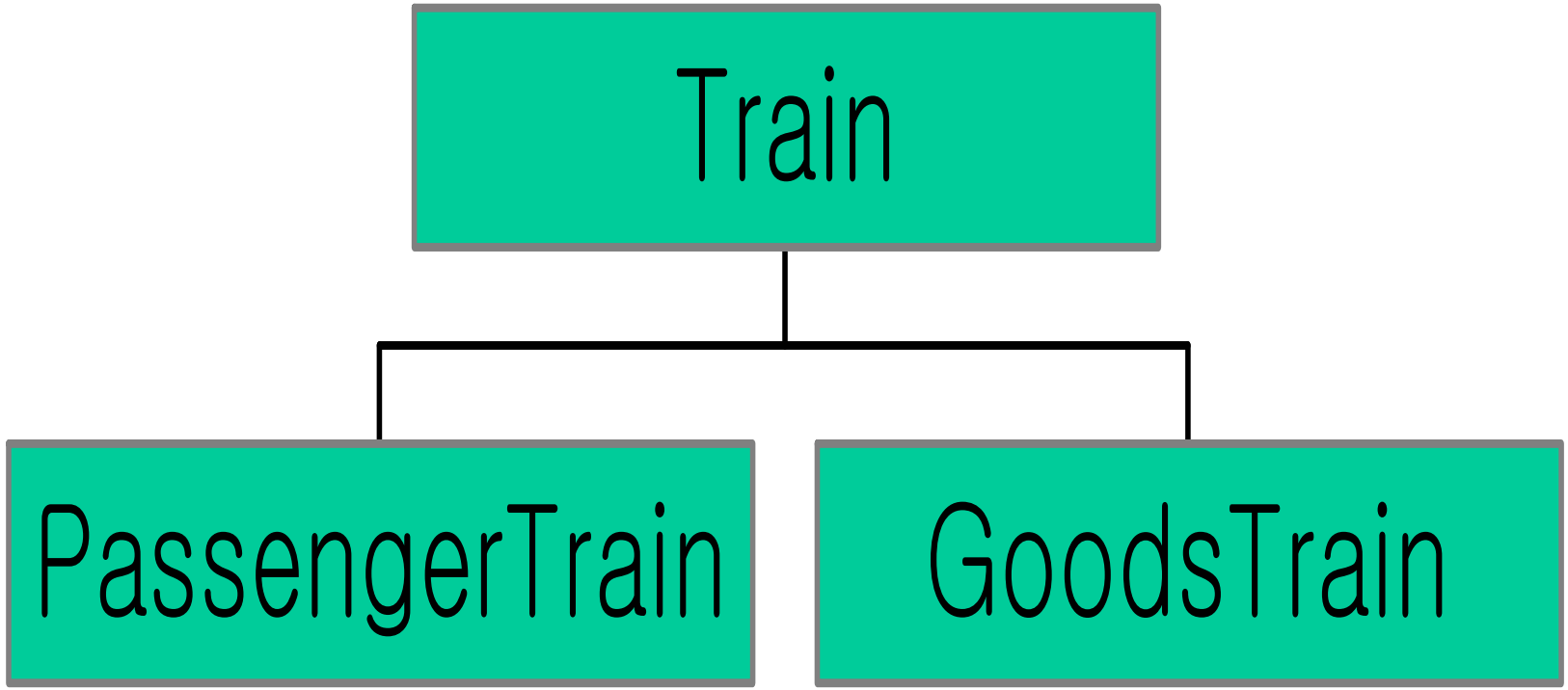
Officer classes



Actions on trains

- Move
- Reverse
- Enumerate railroad cars

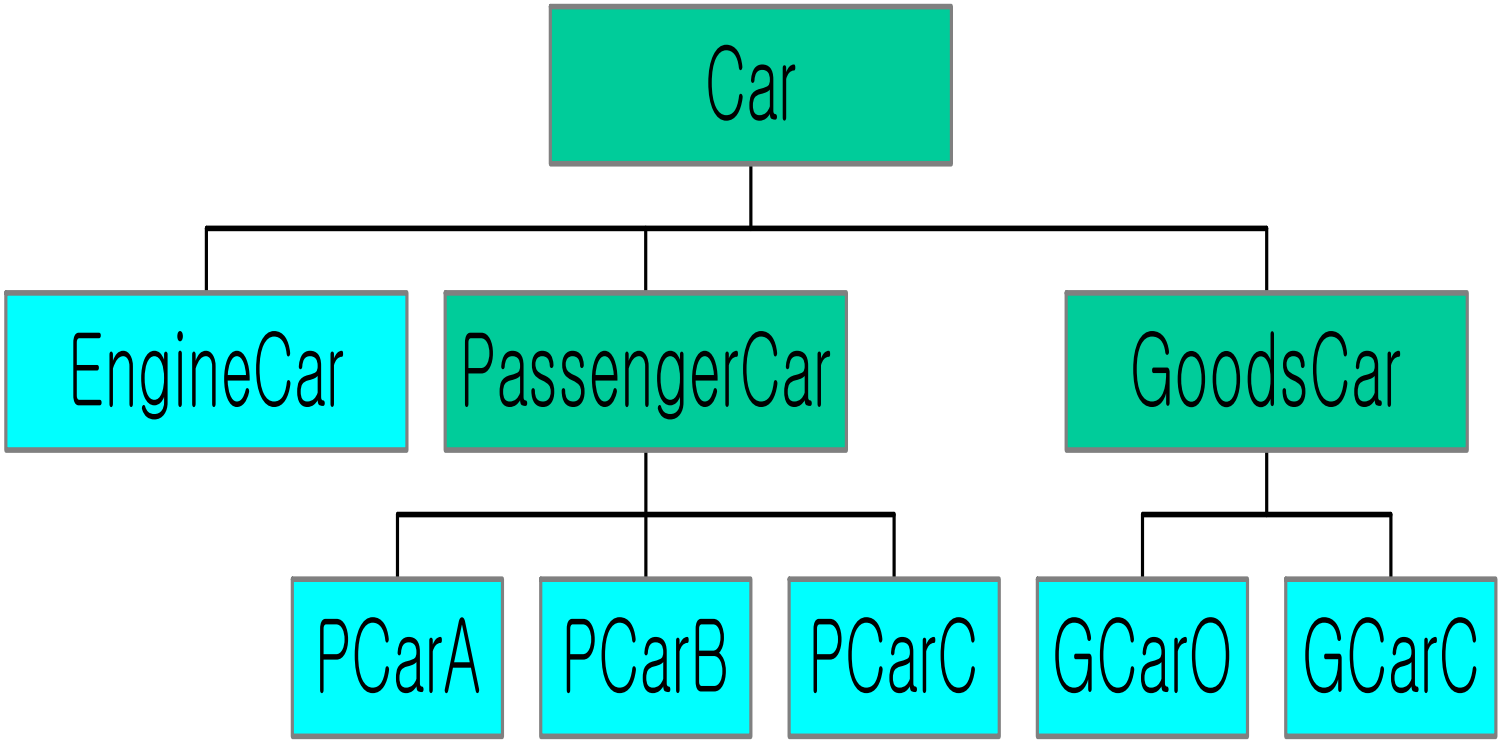
Train classes



Actions on Railroad Cars

- Download contents
- Compute amount of contents

Carriage classes



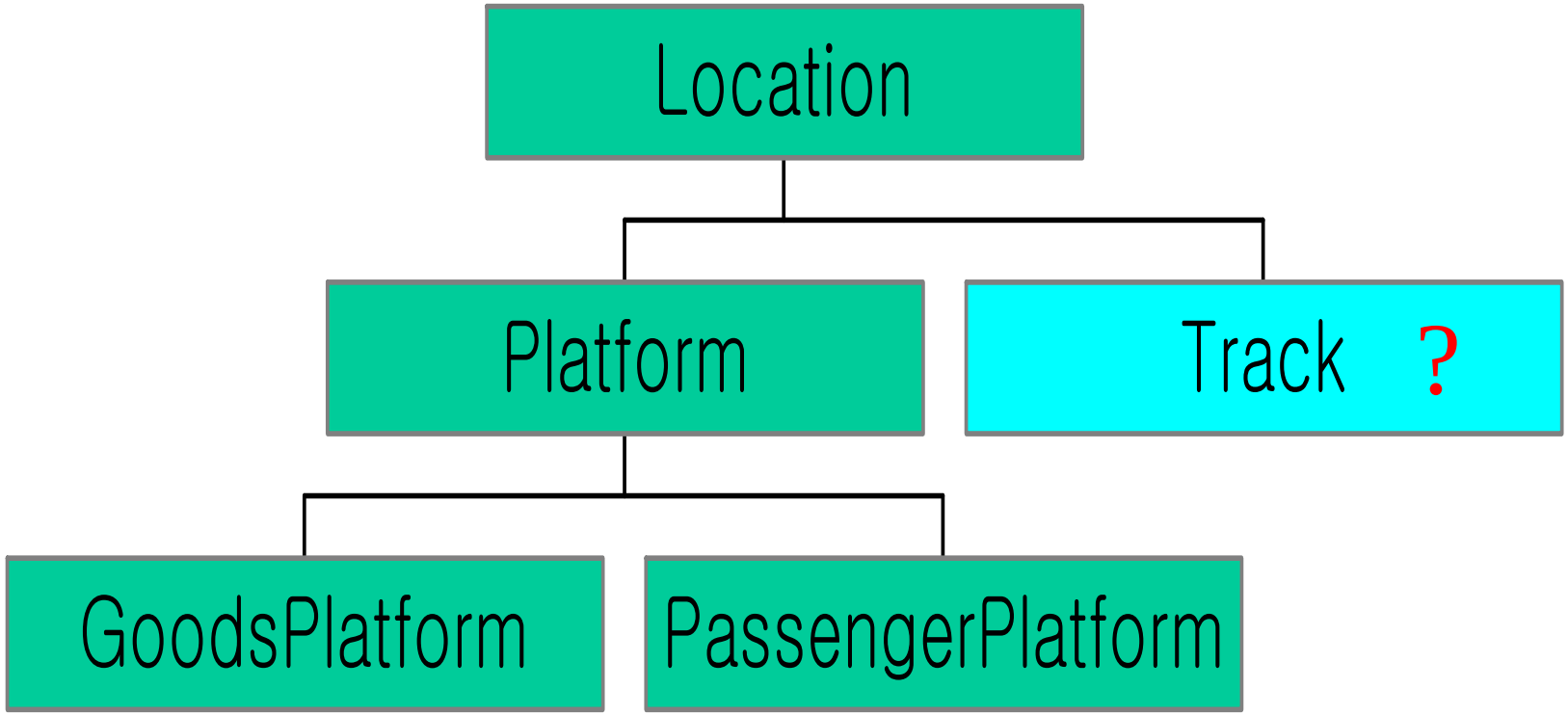
Actions on Station

- Components: platform, track
- Operations
  - Compute number of rejected trains and losses

Actions on station components

- Platforms
  - Check if occupied
  - Compute number of serviced trains
  - Compute amount of downloaded content
  - Compute amount of non-downloaded content

Station component classes



Message passing and events

- Many ways to model messages and events
  - Messages and events as method calls
    - `RS.receive() -> SM.check_track() -> (SM.)call_Helper -> GH.download()`
    - `SM.letout() -> WM.check() -> delay_DT`
    - Natural to work with some timing simulation
  - Explicit message-and-event queue
    - Able to model delayed events more nicely

Using event queues

- Events modeled as objects
  - event time and action recorded in object
- Operation
  - select event with minimum time
  - act on event
    - may add new events to the queue
  - repeat

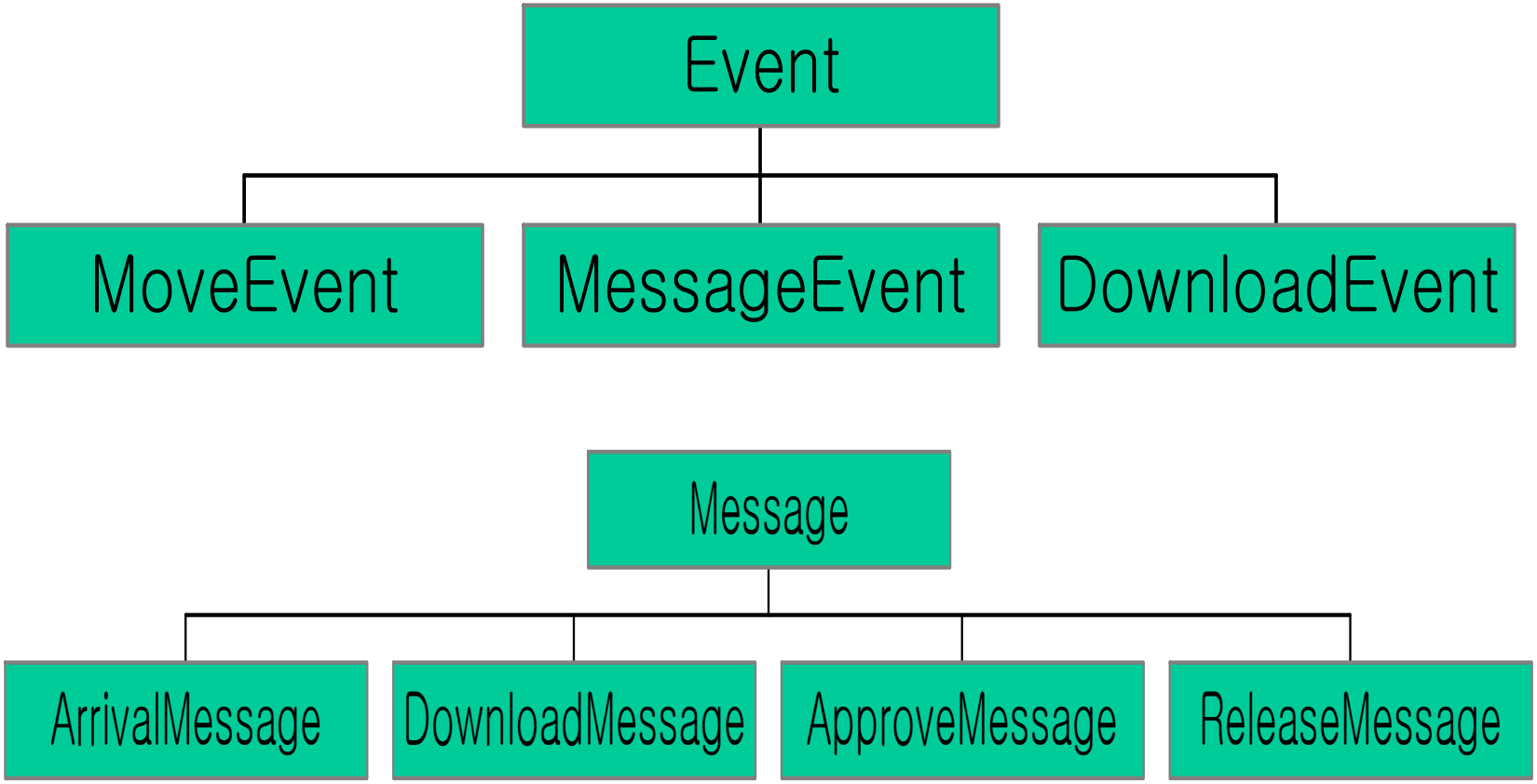
Events

- Actions on events
  - obtain event time
  - act on event
- Event types
  - train arrival events
  - download completion events
  - message events

Messages

- Means by which officers communicate
- Types
  - notify arrival
  - notify completion of download
  - approve
  - release train

Event and message classes



Hints

- Time accurate model
  - 상태 정보를 이용(state diagram)
  - 매 사이클(minute) 마다 수행하게
- Coding style & implementation
  - STL 사용
  - Header, Source 파일 분리
  - Header 파일 include는 꼭 필요한 경우만
  - 메모리 leak이 발생하지 않게 (new & delete)