프로그램 명세서

Programming Language in Java기말과제-티켓팅

2013-08-31

NHN NEXT

131083 최재은

1. Program description

50명의 고객의 여행 정보를 입력 받아 ticketing시간과 여행지 도착 시간을 계산하여 TravelReport.txt로 출력하는 프로그램

1. Class description

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **업무 영역** | | | | 고객 | | | | **업무 기능** | | 고객정보 | | |
| **클래스** | | | | Customer.java | | | | **패키지명** | | ticketing | | |
| **클래스 설명** | | | | 고객의 정보를 입력/초기화/조회하는 클래스 | | | | | | | | |
| **상속 Interface** | | | |  | | | | **상속 Class** | |  | | |
| **속성 (Attribute)** | | | | | | | | | | | | |
| **Visibility** | | **속성명** | | | **속성 Type** | | **설명** | | | | | **비고** |
| private | | id | | | int | | 고객 아이디 | | | | |  |
| private | | name | | | String | | 고객 이름 | | | | |  |
| private | | arrivalTimeAtTicketingOffice | | | Int | | 매표소에 도착하는 시각 | | | | |  |
| private | | timeForTicketing | | | Int | | 티켓팅 하는데 소요되는 시간 | | | | |  |
| private | | Departure | | | String | | 출발지 | | | | |  |
| private | | Destination | | | String | | 목적지 | | | | |  |
| private | | ticketingStartTime | | | int | | 티켓팅이 시작된 시각 | | | | |  |
| private | | ticketingEndTime | | | Int | | 티켓팅이 끝나는 시각 | | | | |  |
| private | | boardingTime | | | Int | | 기차에 탑승하는 시각 | | | | |  |
| private | | arrivalTimeAtDestination | | | Int | | 목적지에 도착하는 시각 | | | | |  |
| private | | totalWaitingTime | | | Int | | 매표소에 도착한 시점부터 기차 탑승 전까지 대기 시간 | | | | |  |
| private | | waitingTimeForTicketing | | | Int | | 티켓팅까지의 대기시간 | | | | |  |
| private | | waitingTimeForBoarding | | | Int | | 티켓팅 완료 후 기차탑승까지의 대기시간 | | | | |  |
| private | | departingTime | | | Int | | 탑승한 기차가 떠난 시각 | | | | |  |
| private | | arrivingTime | | | Int | | 탑승한 기차가 목적지에 도착한 시각 | | | | |  |
| private | | FILE\_PATH | | | String | | 고객정보를 가진 파일의 위치 | | | | |  |
| public | | COMPARE\_BY\_TICKETING \_TIME | | | Comparator<Customer> | | 고객의 리스트를 티켓팅 시간으로 sort할 Comparator | | | | |  |
| public | | COMPARE\_BY\_ID | | | Comparator<Customer> | | 고객의 리스트를 고객 id로 sort할 Comparator | | | | |  |
| **연산자 (Operation)** | | | | | | | | | | | | |
| **Visibility** | **연산명** | | | | | **파라메터** | | | **반환값** | | **비고** | |
| public | readCustomerListFromFile | | | | | (none) | | | ArrayList<Customer> | |  | |
| public | getName | | | | | (none) | | | String | |  | |
| public | getId | | | | | (none) | | | int | |  | |
| public | getArrivalTimeAtTicketingOffice | | | | | (none) | | | Int | |  | |
| public | getTimeForTicketing | | | | | (none) | | | Int | |  | |
| public | setTicketingStartEndTime | | | | | time:int | | | void | |  | |
| public | getTicketingEndTime | | | | | (none) | | | int | |  | |
| public | toString (override) | | | | | (none) | | | String | |  | |
| public | getTicketingStartTime | | | | | (none) | | | int | |  | |
| public | getBoardingTime | | | | | (none) | | | int | |  | |
| public | setBoardingTime | | | | | currentTime:int | | | void | |  | |
| public | getDeparture | | | | | (none) | | | String | |  | |
| public | getDestination | | | | | (none) | | | String | |  | |
| public | getTotalWaitingTime | | | | | (none) | | | Int | |  | |
| public | setTotalWaitingTime | | | | | boardingTime:int | | | void | |  | |
| public | setArrivalTimeAtDestination | | | | | arrivalTime:int | | | void | |  | |
| public | getArrivalTimeAtDestination | | | | | (none) | | | Int | |  | |
| public | getWaitingTimeForTicketing | | | | | (none) | | | int | |  | |
| public | getWaitingTimeForBoarding | | | | | (none) | | | Int | |  | |
| public | getDepartingTime | | | | | (none) | | | int | |  | |
| public | setDepartingTime | | | | | currentTime:int | | | void | |  | |
| **연산자 (Operation) 명세** | | | | | | | | | | | | |
| **연산자명** | | | readCustomerListFromFile (no parameter) : ArrayList<Customer> | | | | | | | | | |
| **설명** | | | 파일로부터 읽은 고객 정보를 ArrayList로 만들어 리턴한다. | | | | | | | | | |
| **처리 로직** | | | 1. FileReader로 파일을 읽어 BufferedReader로 넘긴다 2. BufferedReader가 파일을 한 줄씩 읽어서 Customer 생성자로 넘긴다. 3. Customer 생성자에서는 space를 기준으로 정보를 column별로 구분하여 고객의 id, 이름, 매표소 도착 시간, 티켓팅 소요시간, 출발지, 목적지 정보를 초기화하는 고객 인스턴스를 생성한다. 4. 파일에서 읽은 모든 고객의 리스트를 리턴한다. | | | | | | | | | |
| **외부호출** | | | Customer.readCustomerListFromFile() | | | | | | | | | |
| **연산자 (Operation) 명세** | | | | | | | | | | | | |
| **연산자명** | | | getName (no parameter) : String | | | | | | | | | |
| **설명** | | | 고객의 이름을 리턴한다. | | | | | | | | | |
| **처리 로직** | | | Customer 클래스의 멤버변수인 고객의 이름을 리턴한다. | | | | | | | | | |
| **연산자 (Operation) 명세** | | | | | | | | | | | | |
| **연산자명** | | | getId (no parameter) : int | | | | | | | | | |
| **설명** | | | 고객의 아이디를 리턴한다. | | | | | | | | | |
| **처리 로직** | | | Customer 클래스의 멤버변수인 고객의 아이디를 리턴한다. | | | | | | | | | |
| **연산자 (Operation) 명세** | | | | | | | | | | | | |
| **연산자명** | | | getArrivalTimeAtTicketingOffice (no parameter) : int | | | | | | | | | |
| **설명** | | | 고객이 매표소에 도착한 시간을 리턴한다. | | | | | | | | | |
| **처리 로직** | | | Customer 클래스의 멤버변수인 고객의 매표소 도착시간을 리턴한다. | | | | | | | | | |
| **연산자 (Operation) 명세** | | | | | | | | | | | | |
| **연산자명** | | | getTimeForTicketing (no parameter) : int | | | | | | | | | |
| **설명** | | | 티켓팅 소요시간을 리턴한다. | | | | | | | | | |
| **처리 로직** | | | Customer 클래스의 멤버변수인 고객의 티켓팅 소요시간을 리턴한다. | | | | | | | | | |
| **연산자 (Operation) 명세** | | | | | | | | | | | | |
| **연산자명** | | | setTicketingStartEndTime (time:int) : void | | | | | | | | | |
| **설명** | | | 티켓팅 시작과 종료 시간을 assign한다. | | | | | | | | | |
| **처리 로직** | | | Customer 클래스의 멤버변수인 고객의 티켓팅 시작시간을 입력한다.  Customer 클래스의 멤버변수인 고객의 티켓팅 종료시간을 입력한다. | | | | | | | | | |
| **연산자 (Operation) 명세** | | | | | | | | | | | | |
| **연산자명** | | | getTicketingEndTime (no parameter) : int | | | | | | | | | |
| **설명** | | | 티켓팅 종료시간을 리턴한다. | | | | | | | | | |
| **처리 로직** | | | Customer 클래스의 멤버변수인 티켓팅 종료시간을 리턴한다. | | | | | | | | | |
| **연산자 (Operation) 명세** | | | | | | | | | | | | |
| **연산자명** | | | toString (override) (no parameter) : String | | | | | | | | | |
| **설명** | | | 고객을 콘솔에 프린트아웃 할 때 고객의 이름을 출력한다 | | | | | | | | | |
| **처리 로직** | | | Customer 클래스의 멤버변수인 고객의 이름을 리턴한다. | | | | | | | | | |
| **연산자 (Operation) 명세** | | | | | | | | | | | | |
| **연산자명** | | | getTicketingStartTime (no parameter) : int | | | | | | | | | |
| **설명** | | | 티켓팅 시작 시간을 리턴한다. | | | | | | | | | |
| **처리 로직** | | | Customer 클래스의 멤버변수인 티켓팅 시작 시간 이름을 리턴한다. | | | | | | | | | |
| **연산자 (Operation) 명세** | | | | | | | | | | | | |
| **연산자명** | | | getBoardingTime (no parameter) : int | | | | | | | | | |
| **설명** | | | 기차 탑승 시간을 리턴한다. | | | | | | | | | |
| **처리 로직** | | | Customer 클래스의 멤버변수인 고객의 기차 탑승 시간을 리턴한다. | | | | | | | | | |
| **연산자 (Operation) 명세** | | | | | | | | | | | | |
| **연산자명** | | | setBoardingTime (currentTime:int) : void | | | | | | | | | |
| **설명** | | | 기차 탑승 시간을 설정한다. | | | | | | | | | |
| **처리 로직** | | | Parameter로 받은 시각을 고객의 기차 탑승시간으로 설정한다. | | | | | | | | | |
| **연산자 (Operation) 명세** | | | | | | | | | | | | |
| **연산자명** | | | getDeparture (no parameter) : String | | | | | | | | | |
| **설명** | | | 고객의 출발지를 리턴한다. | | | | | | | | | |
| **처리 로직** | | | Customer 클래스의 멤버변수인 고객의 출발지를 리턴한다. | | | | | | | | | |
| **연산자 (Operation) 명세** | | | | | | | | | | | | |
| **연산자명** | | | getDestination (no parameter) : String | | | | | | | | | |
| **설명** | | | 고객의 목적지를 리턴한다. | | | | | | | | | |
| **처리 로직** | | | Customer 클래스의 멤버변수인 고객의 목적지를 리턴한다. | | | | | | | | | |
| **연산자 (Operation) 명세** | | | | | | | | | | | | |
| **연산자명** | | | getTotalWaitingTime (no parameter) : String | | | | | | | | | |
| **설명** | | | 고객의 이름을 리턴한다. | | | | | | | | | |
| **처리 로직** | | | Customer 클래스의 멤버변수인 고객의 이름을 리턴한다. | | | | | | | | | |
| **연산자 (Operation) 명세** | | | | | | | | | | | | |
| **연산자명** | | | setTotalWaitingTime (arrivalTimeAtDestination : int) : void | | | | | | | | | |
| **설명** | | | 고객이 매표소에 도착했을 때부터 기차 탑승 시까지 걸린 시간을 리턴한다. | | | | | | | | | |
| **처리 로직** | | | Customer의 멤버변수인 기차 탑승 시각과 매표소에 도착한 시각의 차를 리턴한다. | | | | | | | | | |
| **연산자 (Operation) 명세** | | | | | | | | | | | | |
| **연산자명** | | | setArrivalTimeAtDestination (arrivalTimeAtDestination : int) : void | | | | | | | | | |
| **설명** | | | 고객이 목적지에 도착한 시각을 설정한다. | | | | | | | | | |
| **처리 로직** | | | Customer 클래스의 멤버변수인 목적지 도착 시각에 인자로 넘겨받은 시각을 설정한다. | | | | | | | | | |
| **연산자 (Operation) 명세** | | | | | | | | | | | | |
| **연산자명** | | | getArrivalTimeAtDestination (no parameter) : int | | | | | | | | | |
| **설명** | | | 고객이 목적지에 도착한 시각을 리턴한다. | | | | | | | | | |
| **처리 로직** | | | Customer 클래스의 멤버변수인 목적지 도착시각을 리턴한다. | | | | | | | | | |
| **연산자 (Operation) 명세** | | | | | | | | | | | | |
| **연산자명** | | | getWaitingTimeForTicketing (no parameter) : int | | | | | | | | | |
| **설명** | | | 고객의 티켓팅 대기시간을 리턴한다. | | | | | | | | | |
| **처리 로직** | | | Customer 클래스의 멤버변수인 티켓팅 시작 시각과 매표소 도착 시각의 차를 리턴한다. | | | | | | | | | |
| **연산자 (Operation) 명세** | | | | | | | | | | | | |
| **연산자명** | | | getWaitingTimeForBoarding (no parameter) : int | | | | | | | | | |
| **설명** | | | 기차 대기시간은 리턴한다. | | | | | | | | | |
| **처리 로직** | | | Customer 클래스의 멤버변수인 기차 탑승 시각과 티켓팅 종료 시각의 차를 리턴한다. | | | | | | | | | |
| **연산자 (Operation) 명세** | | | | | | | | | | | | |
| **연산자명** | | | getDepartingTime (no parameter) : int | | | | | | | | | |
| **설명** | | | 기차 출발 시각을 리턴한다. | | | | | | | | | |
| **처리 로직** | | | Customer 클래스의 멤버변수인 기차 출발 시각을 리턴한다. | | | | | | | | | |
| **연산자 (Operation) 명세** | | | | | | | | | | | | |
| **연산자명** | | | setDepartingTime (currentTime : int) : void | | | | | | | | | |
| **설명** | | | 기차 출발 시각을 설정한다. | | | | | | | | | |
| **처리 로직** | | | 인자로 받은 시각을 고객이 탑승한 기차 출발 시각으로 설정한다. | | | | | | | | | |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **업무 영역** | | | 티켓팅 메인 | | **업무 기능** | | 메인함수 | |
| **클래스** | | | TicketingMain.java | | **패키지명** | | ticketing | |
| **클래스 설명** | | | 이 프로그램의 메인 함수를 가지고 있다. | | | | | |
| **연산자 (Operation)** | | | | | | | | |
| **Visibility** | **연산명** | | | **파라메터** | | **반환값** | | **비고** |
| public | main | | | args | | String[] | |  |
| **연산자 (Operation) 명세** | | | | | | | | |
| **연산자명** | | main (args : String[]) void | | | | | | |
| **설명** | | 매표소 인스턴스를 만들고 고객정보를 전달한다. | | | | | | |
| **처리 로직** | | 1. Customer 클래스의 static 함수인 readCustomerList를 호출하여 고객 리스트를 받는다. 2. 티켓팅 오피스 인스턴스를 생성한다. 3. 티켓팅 오피스 인스턴스의 티켓팅 프로세스 함수인 ticketing에 위에서 받은 고객 리스트를 전달하여 티켓팅을 시작시킨다. | | | | | | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **업무 영역** | | | 매표소 직원 | | | | | **업무 기능** | 매표 | | | |
| **클래스** | | | Agent.java | | | | | **패키지명** | ticketing | | | |
| **클래스 설명** | | | 매표소 직원이 티켓팅하는 프로세스를 담당하는 클래스 | | | | | | | | | |
| **속성 (Attribute)** | | | | | | | | | | | | |
| **속성명** | | | | **속성 Type** | | **설명** | | | | **비고** | |
| allocatedCustomer | | | | ArrayList<Customer> | | 티켓팅 대기라인에서 할당받은 고객리스트 | | | |  | |
| **연산자 (Operation)** | | | | | | | | | | | | |
| **Visibility** | **연산명** | | | | **파라메터** | | **반환값** | | | | **비고** | |
| public | getCustomerList | | | | (none) | | ArrayList<Customer> | | | |  | |
| public | enqueueCustomer | | | | c : Customer | | void | | | |  | |
| public | dequeueCustomer | | | | (none) | | void | | | |  | |
| public | getTimeNeededToFinish | | | | (none) | | int | | | |  | |
| public | ticketing | | | | currentTime : int | | Customer | | | |  | |
| **연산자 (Operation) 명세** | | | | | | | | | | | | |
| **연산자명** | | getCustomerList (no parameter) : ArrayList<Customer> | | | | | | | | | | |
| **설명** | | Agent가 할당받은 고객 리스트를 리턴한다. | | | | | | | | | | |
| **처리 로직** | | Agent 클래스의 멤버 변수인 고객 리스트를 리턴한다. | | | | | | | | | | |
| **연산자 (Operation) 명세** | | | | | | | | | | | | |
| **연산자명** | | enqueueCustomer (c : Customer) : void | | | | | | | | | | |
| **설명** | | Agent가 할당받은 고객 리스트를 리턴한다. | | | | | | | | | | |
| **처리 로직** | | Agent 클래스의 멤버 변수인 고객 리스트를 리턴한다. | | | | | | | | | | |
| **연산자 (Operation) 명세** | | | | | | | | | | | | |
| **연산자명** | | dequeueCustomer (no parameter) : void | | | | | | | | | | |
| **설명** | | Agent의 멤버변수인 고객리스트에서 맨 앞의 고객을 dequeue한다. | | | | | | | | | | |
| **처리 로직** | | ArrayList인 allocatedCustomer의 remove 메소드를 사용하여 맨 앞의 고객을 dequeue한다. | | | | | | | | | | |
| **연산자 (Operation) 명세** | | | | | | | | | | | | |
| **연산자명** | | getTimeNeededToFinish (no parameter) : int | | | | | | | | | | |
| **설명** | | Agent가 할당받은 고객에게 티켓팅해주는데 걸리는 총 시간을 리턴한다. | | | | | | | | | | |
| **처리 로직** | | 1. 고객 리스트의 모든 고객을 한 명씩 for문을 돌려 티켓팅 시간을 누적하여 더한다. 2. 더한 시간을 리턴한다. | | | | | | | | | | |
| **연산자 (Operation) 명세** | | | | | | | | | | | | |
| **연산자명** | | ticketing (currentTime : int) : Customer | | | | | | | | | | |
| **설명** | | Agent가 인자로 받은 시각에 티켓팅을 진행하고, 그 시각에 끝난 고객이 있으면 리턴한다. | | | | | | | | | | |
| **처리 로직** | | 1. Agent가 가진 고객리스트에 아무도 없으면 null을 리턴한다. 2. 그렇지 않으면, 고객리스트의 맨 앞 고객을 dequeue하여 지역변수 customer에 저장한다. 3. 만약 customer가 아직 티켓팅을 시작하지 않았다면 현재 시각을 넘겨주어 티켓팅 시작/종료 시각을 설정한다. 4. 호출한 곳에서 넘겨 받은 인자인 currentTime이 customer의 티켓팅 종료시각이고, Agent 고객리스트에 한 명이라도 있다면, 고객 리스트의 맨 앞 고객을 nextCustomer에 저장하여 nextCustomer의 티켓팅 시작/종료 시각을 설정해준다. (이렇게 하는 이유는, 앞 고객의 티켓팅이 끝나는 시각에 쉬는 시간없이 다음 고객의 티켓팅을 시작하기 위해서 이다.) 5. 티켓팅이 종료된 customer를 리턴한다. 6. 아직 customer의 티켓팅이 종료되지 않았으면 null을 리턴한다. | | | | | | | | | | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **업무 영역** | | | 기차 | | | | **업무 기능** | | 기차 정보 | | | |
| **클래스** | | | Train.java | | | | **패키지명** | | ticketing | | | |
| **클래스 설명** | | | 고객들을 기차에 태우고 그들의 그 이후 시간 정보들을 설정한다. | | | | | | | | | |
| **속성 (Attribute)** | | | | | | | | | | | | |
| **속성명** | | | | **속성 Type** | | **설명** | | | | **비고** | |
| number | | | | int | | 출발하는 순서인 기차 번호 | | | |  | |
| customers | | | | ArrayList<Customer> | | 기차에 탑승한 고객 리스트 | | | |  | |
| **연산자 (Operation)** | | | | | | | | | | | | |
| **Visibility** | **연산명** | | | | **파라메터** | | | **반환값** | | | **비고** | |
| public | boardCustomer | | | | customer : Customer, currentTime : int | | | void | | |  | |
| private | setTimeForCustomer | | | | customer : Customer, currentTime : int | | | void | | |  | |
| public | getCustomers | | | | (none) | | | ArrayList<Customer> | | |  | |
| public | getNumber | | | | (none) | | | int | | |  | |
| **연산자 (Operation) 명세** | | | | | | | | | | | | |
| **연산자명** | | boardCustomer (customer : Customer, currentTime : int) : void | | | | | | | | | | |
| **설명** | | 고객을 기차에 태우고 그의 시간정보를 업데이트 한다.. | | | | | | | | | | |
| **처리 로직** | | 1. 넘겨받은 고객을 기차 탑승 고객 리스트에 추가한다. 2. 그 고객과 현재 시각을 setTimeForCustomer에 넘겨 시간 정보를 업데이트 한다. | | | | | | | | | | |
| **연산자 (Operation) 명세** | | | | | | | | | | | | |
| **연산자명** | | setTimeForCustomer (customer : Customer, currentTime : int) : void | | | | | | | | | | |
| **설명** | | 고객이 기차에 탑승한 이후의 시간정보를 설정한다. | | | | | | | | | | |
| **처리 로직** | | 1. 인자로 받은 currentTime으로 고객의 기차 탑승 시각, 기차를 타기까지의 총 대기시간을 설정한다. 2. 목적지 도착 시각 계산 클래스인 ArrivalTimeCalculator의 생성자에 customer를 인자로 주어 인스턴스를 생성한다. 3. ArrivalTimeCalculator의 함수인 calculate를 호출한다. | | | | | | | | | | |
| **연산자 (Operation) 명세** | | | | | | | | | | | | |
| **연산자명** | | getCustomers (no parameter) : ArrayList<Customer> | | | | | | | | | | |
| **설명** | | Train의 멤버변수인 고객리스트를 리턴한다. | | | | | | | | | | |
| **연산자 (Operation) 명세** | | | | | | | | | | | | |
| **연산자명** | | getNumber (no parameter) : int | | | | | | | | | | |
| **설명** | | Train의 멤버변수인 기차번호를 리턴한다. | | | | | | | | | | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **업무 영역** | | | 목적지 도착 시간 계산 | | | | **업무 기능** | | 도착 시간 계산 | | | |
| **클래스** | | | ArrivalTimeCalculator.java | | | | **패키지명** | | ticketing | | | |
| **클래스 설명** | | | 고객들을 기차에 태우고 그들의 그 이후 시간 정보들을 설정한다. | | | | | | | | | |
| **속성 (Attribute)** | | | | | | | | | | | | |
| **속성명** | | | | **속성 Type** | | **설명** | | | | **비고** | |
| customer | | | | Customer | | 도착 시간 계산할 고객 | | | |  | |
| maxNodeCount | | | | int | | 다른 멤버변수인distanceMatrix의 최대 인덱스 카운트 | | | |  | |
| distanceMatrix | | | | int[][] | | 각 지역간의 편도 여행에 걸리는 시간 | | | |  | |
| **연산자 (Operation)** | | | | | | | | | | | | |
| **Visibility** | **연산명** | | | | **파라메터** | | | **반환값** | | | **비고** | |
| public | calculate | | | | (none) | | | void | | |  | |
| private | getTravelTime | | | | departure : String, destination : String | | | int | | |  | |
| private | getIndex | | | | name : String | | | int | | |  | |
| **연산자 (Operation) 명세** | | | | | | | | | | | | |
| **연산자명** | | calculate (no parameter) : void | | | | | | | | | | |
| **설명** | | 고객을 기차에 태우고 그의 시간정보를 업데이트 한다.. | | | | | | | | | | |
| **처리 로직** | | 1. 이 클래스를 생성할 때 생성자에서 인자로 받은 고객의 기차 탑승시간을 받는다. 2. 이 클래스의 멤버 함수인 getTravelTime에 출발지와 목적지를 보내서 편도 여행 시간을 얻는다. 3. 기차 탑승시간과 편도 여행시간을 더해 arrivalTime에 저장한다. 4. arrivalTime을 고객의 멤버함수에 보내서 고객의 정보를 업데이트 한다. | | | | | | | | | | |
| **연산자 (Operation) 명세** | | | | | | | | | | | | |
| **연산자명** | | getTravelTime (departure : String, destination : String) : int | | | | | | | | | | |
| **설명** | | 고객의 출발지와 목적지를 받아서 두 지점 간의 최소 편도 여행 시간을 구한다. | | | | | | | | | | |
| **처리 로직** | | 1. 이 클래스 멤버함수인 getIndex에 출발지를 보내 distanceMatrix에서의 인덱스를 받는다. 2. 이 클래스 멤버함수인 getIndex에 목적지를 보내 distanceMatrix에서의 인덱스를 받는다. 3. 삼중for문으로 두 지점간의 최소 이동 시간을 구해서 distanceMatrix를 업데이트한다. 4. 두 지역간의 이동 시간인 변수 travelTime에 업데이트 된 최소 이동 시간을 저장한다. 5. travelTime을 리턴한다. | | | | | | | | | | |
| **연산자 (Operation) 명세** | | | | | | | | | | | | |
| **연산자명** | | getIndex (name : String) : int | | | | | | | | | | |
| **설명** | | 지역 이름을 받아서 distanceMatrix에서의 인덱스를 리턴한다. | | | | | | | | | | |
| **처리 로직** | | switch-case문으로 distanceMatrix에서 name의 인덱스를 리턴한다. | | | | | | | | | | |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **업무 영역** | | | 여행 리포트 | | **업무 기능** | | 리포트 파일 생성 | |
| **클래스** | | | TravelReport.java | | **패키지명** | | ticketing | |
| **클래스 설명** | | | 고객의 여행 목적지 도착시간 계산이 끝난 후 고객정보를 파일로 출력하는 클래스 | | | | | |
| **연산자 (Operation)** | | | | | | | | |
| **Visibility** | **연산명** | | | **파라메터** | | **반환값** | | **비고** |
| public | documentCustomer | | | trainList : ArrayList<Train> | | void | |  |
| private | makeReportHeader | | | (none) | | StringBuilder | |  |
| **연산자 (Operation) 명세** | | | | | | | | |
| **연산자명** | | documentCustomer (trainList : ArrayList<Train>) : void | | | | | | |
| **설명** | | 기차에 탄 고객들의 정보를 파일로 출력한다. | | | | | | |
| **처리 로직** | | 1. 처음 파일로부터 읽어들인 모든 고객들이 탄 기차들의 리스트와 그 안의 고객 리스트를 이용하여, 이 메소드의 지역 변수인 entireCustomerList에 차례로 추가한다. 2. 고객의 아이디 순으로 출력하기 위해, Customer 클래스의 COMPARE\_BY\_ID comparator로 entireCustomerList를 sort한다. 3. makeReportHeader 함수를 이용해 헤더를 만들어 파일에 출력한다. 4. for문으로 entireCustomerList의 고객들을 읽어 출력하고자 하는 고객 정보들을 출력한다. | | | | | | |
| **연산자 (Operation) 명세** | | | | | | | | |
| **연산자명** | | makeReportHeader (no parameter) : void | | | | | | |
| **설명** | | 여행 리포트의 헤더 부분에 들어갈 포맷을 만든다. | | | | | | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **업무 영역** | | | 매표소 | | | | **업무 기능** | | 매표소 프로세스 | | | |
| **클래스** | | | TicketingOffice.java | | | | **패키지명** | | ticketing | | | |
| **클래스 설명** | | | 매표소에서 일어나는 모든 프로세스를 관리하는 클래스 | | | | | | | | | |
| **속성 (Attribute)** | | | | | | | | | | | | |
| **속성명** | | | | **속성 Type** | | **설명** | | | | | **비고** |
| MAX\_TIME | | | | int | | ticketing 내 for문의 index max | | | | |  |
| agents | | | | ArrayList<Agent> | | 티켓팅을 하는 직원 리스트 | | | | |  |
| waitingAgentLine | | | | ArrayList<Customer> | | 매표소에 도착해 티켓팅을 기다리는 고객 리스트 | | | | |  |
| waitingTrainLine | | | | ArrayList<Customer> | | 티켓팅을 마치고 기차를 기다리는 고객 리스트 | | | | |  |
| trainList | | | | ArrayList<Train> | | 출발한 기차 리스트 | | | | |  |
| customerList | | | | ArrayList<Customer> | | 이 클래스를 열고 ticketing 함수를 호출한 곳에서 받은 고객 리스트를 저장할 곳 | | | | |  |
| remainingCustomer | | | | ArrayList<Customer> | | 티켓팅을 아직 마치지 못한 고객 리스트 | | | | |  |
| report | | | | TravelReport | | 티켓팅이 끝나고 결과를 출력할 리포트 인스턴스 | | | | |  |
| **연산자 (Operation)** | | | | | | | | | | | | |
| **Visibility** | **연산명** | | | | **파라메터** | | | **반환값** | | **비고** | | |
| private | setUpAgents | | | | (none) | | | void | |  | | |
| private | setCustomerList | | | | list : ArrayList<Customer> | | | void | |  | | |
| public | ticketing | | | | list : ArrayList<Customer> | | | void | |  | | |
| private | lineUpWaitingQueue | | | | currnetTime : int | | | void | |  | | |
| private | allocateCustomerToAgent | | | | time : int | | | void | |  | | |
| private | ticketingByAgent | | | | currentTime : int | | | void | |  | | |
| private | startTrain | | | | currentTime : int | | | void | |  | | |
| private | isEveryAgentDone | | | | (none) | | | boolean | |  | | |
| private | getLeastBusyAgent | | | | (none) | | | Agent | |  | | |
| **연산자 (Operation) 명세** | | | | | | | | | | | | |
| **연산자명** | | setUpAgents (no parameter) : void | | | | | | | | | | |
| **설명** | | 세 명의 직원을 직원리스트에 추가한다. | | | | | | | | | | |
| **연산자 (Operation) 명세** | | | | | | | | | | | | |
| **연산자명** | | setCustomerList (list : ArrayList<Customer>) : void | | | | | | | | | | |
| **설명** | | 이 클래스의 멤버 변수에 넘겨받은 고객리스트를 저장한다. | | | | | | | | | | |
| **연산자 (Operation) 명세** | | | | | | | | | | | | |
| **연산자명** | | ticketing (list : ArrayList<Customer>) : void | | | | | | | | | | |
| **설명** | | 고객이 도착하는 것부터 리포트 파일을 만드는 것 까지의 프로세스를 포함한다. | | | | | | | | | | |
| **처리 로직** | | 1. for문으로 시간이 가게 하면서, 각 직원이 가진 줄에 서지 못한 고객이 없고, 모든 직원이 티켓팅을 마치고, 기차를 기다리는 고객 리스트에 아무도 없으면 for문을 나간다. 2. 그렇지 않으면, 해당 시간에 매표소에 도착하는 고객들을 줄 세운다. 3. 고객들을 직원에게 할당하는 함수에 현재 시각을 보내어 호출한다. 4. 시간에 따라 직원들이 한 명씩 고객의 티켓팅을 진행하는 함수에 현재 시각을 보낸다. 5. 현재 시각이 5의 배수이면 기차가 출발하는 함수에 현재 시각을 보낸다. 6. 시간이 MAX\_TIME까지 갔다면, 출발한 기차의 리스트를 리포트를 만드는 함수로 보낸다.. | | | | | | | | | | |
| **연산자 (Operation) 명세** | | | | | | | | | | | | |
| **연산자명** | | lineUpWaitingQueue (currentTime : int) : void | | | | | | | | | | |
| **설명** | | 매표소에 도착한 고객들을 티켓팅 대기시간을 최소로 하는 방식으로 줄 세운다. | | | | | | | | | | |
| **처리 로직** | | 1. 모든 고객의 리스트에서 현재 시각에 도착하는 사람들을 대기라인에 입력한다. 2. 현재 시각에 대기라인에 있는 고객들을 티켓팅 시간이 짧은 사람을 앞으로 세우는 방법으로 sort한다. | | | | | | | | | | |
| **연산자 (Operation) 명세** | | | | | | | | | | | | |
| **연산자명** | | allocateCustomerToAgent (time : int) : void | | | | | | | | | | |
| **설명** | | 매표소 직원을 만나길 기다리는 대기라인에서 세 명에 직원에 고객들을 할당한다. | | | | | | | | | | |
| **처리 로직** | | 1. 대기라인 앞에 있는 고객을 부른다. 2. getLeastBusyAgent 함수로 직원 당 대기라인 큐에 있는 고객들의 티켓팅 소요시간 합이 가장 적은 직원을 리턴받는다. 3. 2)의 직원의 큐에 1)의 고객을 enqueue한다. 4. 이 과정을 직원을 기다리는 고객을 대상으로 모두 한다. 5. 직원을 기다리는 대기라인을 비운다. | | | | | | | | | | |
| **연산자 (Operation) 명세** | | | | | | | | | | | | |
| **연산자명** | | ticketingByAgent (currentTime : int) : void | | | | | | | | | | |
| **설명** | | 위의 함수에서 enqueue한 고객들을 가진 직원들의 티켓팅을 진행한다. | | | | | | | | | | |
| **처리 로직** | | 1. 전달받은 현재 시각에 각 직원마다 티켓팅을 진행한다. 2. 현재 시각에 티켓팅을 마친 직원에게서 고객을 받아서 기차 대기 라인에 추가한다. 3. 티켓팅을 마친 고객을 remainingCustomer 리스트에서 제거한다. 4. 발권받은 직원의 대기 리스트에서도 dequeue한다. | | | | | | | | | | |
| **연산자 (Operation) 명세** | | | | | | | | | | | | |
| **연산자명** | | startTrain (currentTime : int) : void | | | | | | | | | | |
| **설명** | | 열차를 출발시키고 열차의 정보를 다룬다. | | | | | | | | | | |
| **처리 로직** | | 1. 전달받은 현재시각을 5로 나누어 열차에 번호를 매기고, 매표소의 열차 리스트에 추가한다. 2. 열차 대기 라인에서 기다리고 있던 고객들을 하나씩 열차에 태운다. 3. 열차 대기 라인을 clear한다. | | | | | | | | | | |
| **연산자 (Operation) 명세** | | | | | | | | | | | | |
| **연산자명** | | isEveryAgentDone (no parameter) : boolean | | | | | | | | | | |
| **설명** | | 모든 직원이 일을 마쳤는지 여부를 확인한다. | | | | | | | | | | |
| **처리 로직** | | 각 직원의 업무 종료 시간을 계산하여 모두 끝났으면 true를 리턴한다. | | | | | | | | | | |
| **연산자 (Operation) 명세** | | | | | | | | | | | | |
| **연산자명** | | getLeastBusyAgent (no parameter) : Agent | | | | | | | | | | |
| **설명** | | 처리해야 할 일이 가장 빨리 끝날 직원을 리턴한다. | | | | | | | | | | |
| **처리 로직** | | 1. leastBusyAgent에 임의의 직원으로 초기화한다. 2. 모든 직원을 돌아가며 Agent 클래스의 getTimeNeededToFinish함수로 끝나려면 얼마나 시간이 필요한지 보고 더 적은 직원을 leastBusyAgent에 저장한다. 3. leastBusyAgent를 리턴한다. | | | | | | | | | | |