

C# 프로그래밍 과제 2

- 입력 파일명: "input.txt"
- 출력 파일명: "output.txt"
- 제출일: 5 월 28 일 (토) 23 시 59 분까지
- 제출물
 - 소스코드를 포함한 솔루션 전체를 압축한 파일 (예: 블랙보드에서 강의자료 예제코드 올린 것 참고)
 - 소스코드에 적절한 주석을 추가한다. 특히, 모든 메소드에 대하여 메소드의 역할 또는 기능을 간략히 기술한다.
 - 보고서
 1. 구현 정도 (과제에서 요구한 기능 중 얼마나 구현했는지 서술)
 2. 알고리즘 (의사 코드 또는 전체적인 구성 및 알고리즘 설명)
 3. 개발 환경 (닷넷 버전)

과제 2. 배달 자동차 관리 시스템

본 과제에서는 입력파일 “input.txt” 에서 명령어를 입력 받아 배달 서비스를 위한 자동차를 관리하는 프로그램을 작성한다. 이를 위해 C# 의 배열 자료구조를 이용한다 (단, 리스트, 큐와 같은 C#의 다른 컬렉션을 이용하면 안된다.) 배달 자동차 관리시스템(즉, DeliveryVehicleManager) 은 다수의 자동차 대기 장소(즉, waitPlace)를 관리한다. 각 대기장소 (waitPlace)에서는 다수의 자동차(즉, Vehicle)를 관리할 수 있다. 대기장소의 개수와 자동차의 개수는 제한을 두지 않는다.

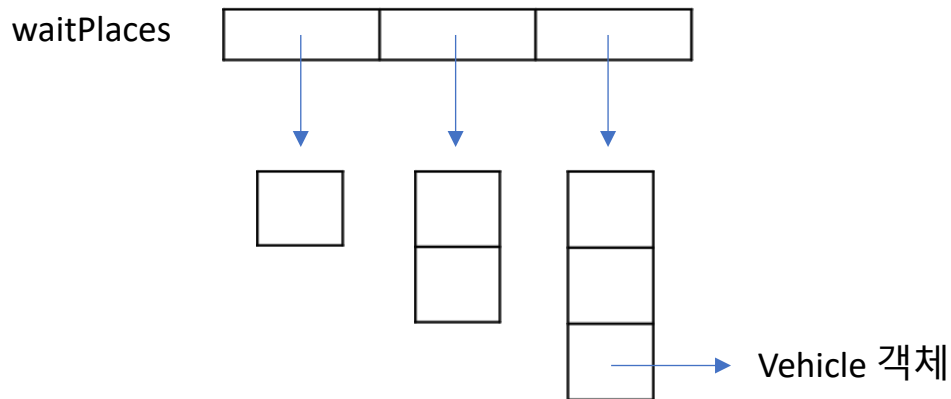
배달 자동차 관리시스템(DeliveryVehicleManager)에서 대기장소(waitPlace)는 배열 형태의 필드 멤버로 가지고 있다. 대기장소(waitPlace) 배열의 각 요소로 배달자동차(DeliveryVehicle)에 대한 배열 자료구조를 가지고 있다.

배달 자동차 관리시스템(DeliveryVehicleManager)과 배달자동차(DeliveryVehicle) 클래스는 다음과 같이 구성한다.

```
class DeliveryVehicle
{
    int vehicleId;           // 자동차 아이디
    string destination;      // 배달 목적지 문자열
    int priority;            // 우선순위
    // ...
}
```

[illegible]

예를 들어, 배달 자동차 관리시스템(DeliveryVehicleManager)에서 3 개의 대기장소(waitPlace)를 관리하고 있고, 각 대기장소에 1, 2, 3 대의 배달자동차(DeliveryVehicle)가 대기하고 있다면, DeliveryVehicleManager 클래스의 waitPlaces 멤버는 다음과 같은 구조를 가진다.



각 클래스 객체와 멤버 생성은 다음의 사항들을 주의한다. 먼저 입력파일로부터 대기장소(waitPlace)의 개수를 입력 받아 대기장소 배열(waitPlaces)을 생성한다. 이때 대기장소 배열의 크기는 프로그램이 종료될 때까지 변하지 않는다. 또한, 각 대기장소에 배달자동차를 관리하기 위해 대기장소 배열의 각 요소를 배달자동차에 대한 배열 자료구조로 구성한다.

각 대기장소에 대해 배달자동차 리스트를 생성 또는 확장할 것이다. 각 대기장소의 배달자동차에 대한 배열 자료구조는 배달자동차의 우선순위(priority)에 따라 정렬한다. 가장 우선순위(priority) 필드 값이 낮은 배달자동차가 우선순위가 가장 높다. (우선순위가 높을 수록 배열 앞쪽에 배치)

배달이 취소된 경우, 해당 배달자동차를 배열에서 삭제한다. 이때 주의할 점은 반드시 대기장소 배열과 각 요소의 배열은 항상 최소의 메모리를 사용하도록 보장해야 한다. 즉, 배열의 용량이 저장하고 있는 정보의 개수보다 큰 경우 평가점수에 큰 감점이 있다.

대기장소는 고유의 ID 가 있고 양의 정수 1 부터 순서대로 ID 가 부여된다. 배달자동차도 고유의 ID 를 가진다.

1. 명령어 처리

- A. Readyin : 배달자동차를 특정 대기장소(waitPlace)에 배정한다.

예) Readyin W2 1000 Address1 P7

➔ 목적지가 Address1 이고, 우선순위가 7 인 배달자동차 1000 을 ID 가 2 인 대기장소에 배정한다. 목적지 문자열의 최대 길이는 20 이다.

➔ 대기장소에 배달자동차를 새롭게 배정할 때마다 해당 배열의 배달자동차의 우선순위에 따라 정렬된다. (우선순위 필드 값이 낮을 수록 우선순위가 높고, 배열은 맨 앞이 가장 우선순위가 높도록 정렬한다.)

➔ 본 명령어의 처리 결과로써 다음의 메시지를 출력파일에 출력한다.

예) Vehicle 1000 assigned to waitPlace #2

- B. Ready : 명령어 Readyin 와 같이 배달자동차를 대기장소(waitPlace)에 배정한다. 단, 현재 대기장소에 가장 적은 배달자동차가 대기하고 있는 대기장소에 배정한다.

예) Ready 600 Address2 P5

➔ 본 명령어의 처리 결과로써 다음의 메시지를 출력파일에 출력한다.

예) Vehicle 600 assigned to waitPlace #1

- C. Status : 각 대기장소에 대기하고 있는 배달자동차의 정보를 출력한다.

예)

***** Delivery Vehicle Info *****

Number of WaitPlaces: 3

WaitPlace #1 Number Vehicles: 0

WaitPlace #2 Number Vehicles: 3

FNUM: 3000 DEST: Address3 PRIO: 1

FNUM: 2000 DEST: Address2 PRIO: 2

FNUM: 1000 DEST: Address1 PRIO: 7

WaitPlace #3 Number Vehicles: 0

***** End Delivery Vehicle Info *****

- D. Cancel : 해당 배달자동차를 대기장소에서 삭제한다.

예) Cancel 2000

➔ 본 명령어의 처리 결과로써 다음의 메시지를 출력파일에 출력한다.

예) Cancellation of Vehicle 2000 completed.

E. Deliver : 해당 대기장소에서 대기하고 있는 배달자동차 중에서 우선순위가 가장 높은 배달자동차를 배달 보낸다.

예) Deliver W1

➔ 본 명령어의 처리 결과로써 다음의 메시지를 출력파일에 출력한다.

예) Vehicle 800 used to deliver.

F. Clear : 해당 대기장소에서 대기하고 있는 배달자동차의 대기를 취소한다.

예) Clear W2

➔ 본 명령어의 처리 결과로써 다음의 메시지를 출력파일에 출력한다.

예) WaitPlace #2 cleared.

G. Quit : 프로그램을 종료한다.

[입력파일 예제]

3

ReadyIn W2 1000 Address1 P7

ReadyIn W2 2000 Address2 P2

ReadyIn W2 3000 Address3 P1

Status

Cancel 2000

Status

Ready 600 Address4 P5

Status

Ready 700 Address5 P4

Status

Ready 800 Address6 P1

Status

Clear W2

Status

Deliver W1

Status

Quit

[출력파일예제]

Vehicle 1000 assigned to WaitPlace #2.

Vehicle 2000 assigned to WaitPlace #2.

Vehicle 3000 assigned to WaitPlace #2.

***** Delivery Vehicle Info *****

Number of WaitPlaces: 3

WaitPlace #1 Number Vehicles: 0

WaitPlace #2 Number Vehicles: 3
FNUM: 3000 DEST: Address3 PRIO: 1
FNUM: 2000 DEST: Address2 PRIO: 2
FNUM: 1000 DEST: Address1 PRIO: 7

WaitPlace #3 Number Vehicles: 0

***** End Delivery Vehicle Info *****

Cancelation of Vehicle 2000 completed.

***** Delivery Vehicle Info *****

Number of WaitPlaces: 3
WaitPlace #1 Number Vehicles: 0

WaitPlace #2 Number Vehicles: 2
FNUM: 3000 DEST: Address3 PRIO: 1
FNUM: 1000 DEST: Address1 PRIO: 7

WaitPlace #3 Number Vehicles: 0

***** End Delivery Vehicle Info *****

Vehicle 600 assigned to WaitPlace #1.

***** Delivery Vehicle Info *****

Number of WaitPlaces: 3
WaitPlace #1 Number Vehicles: 1
FNUM: 600 DEST: Address4 PRIO: 5

WaitPlace #2 Number Vehicles: 2
FNUM: 3000 DEST: Address3 PRIO: 1
FNUM: 1000 DEST: Address1 PRIO: 7

WaitPlace #3 Number Vehicles: 0

***** End Delivery Vehicle Info *****

Vehicle 700 assigned to WaitPlace #3.

***** Delivery Vehicle Info *****

Number of WaitPlaces: 3
WaitPlace #1 Number Vehicles: 1
FNUM: 600 DEST: Address4 PRIO: 5

WaitPlace #2 Number Vehicles: 2
FNUM: 3000 DEST: Address3 PRIO: 1
FNUM: 1000 DEST: Address1 PRIO: 7

WaitPlace #3 Number Vehicles: 1
FNUM: 700 DEST: Address5 PRIO: 4

***** End Delivery Vehicle Info *****

Vehicle 800 assigned to WaitPlace #1.

***** Delivery Vehicle Info *****

Number of WaitPlaces: 3

WaitPlace #1 Number Vehicles: 2

FNUM: 800 DEST: Address6 PRIO: 1

FNUM: 600 DEST: Address4 PRIO: 5

WaitPlace #2 Number Vehicles: 2

FNUM: 3000 DEST: Address3 PRIO: 1

FNUM: 1000 DEST: Address1 PRIO: 7

WaitPlace #3 Number Vehicles: 1

FNUM: 700 DEST: Address5 PRIO: 4

***** End Delivery Vehicle Info *****

WaitPlace #2 cleared.

***** Delivery Vehicle Info *****

Number of WaitPlaces: 3

WaitPlace #1 Number Vehicles: 2

FNUM: 800 DEST: Address6 PRIO: 1

FNUM: 600 DEST: Address4 PRIO: 5

WaitPlace #2 Number Vehicles: 0

WaitPlace #3 Number Vehicles: 1

FNUM: 700 DEST: Address5 PRIO: 4

***** End Delivery Vehicle Info *****

Vehicle 800 used to deliver.

***** Delivery Vehicle Info *****

Number of WaitPlaces: 3

WaitPlace #1 Number Vehicles: 1

FNUM: 600 DEST: Address4 PRIO: 5

WaitPlace #2 Number Vehicles: 0

WaitPlace #3 Number Vehicles: 1

FNUM: 700 DEST: Address5 PRIO: 4

***** End Delivery Vehicle Info *****