

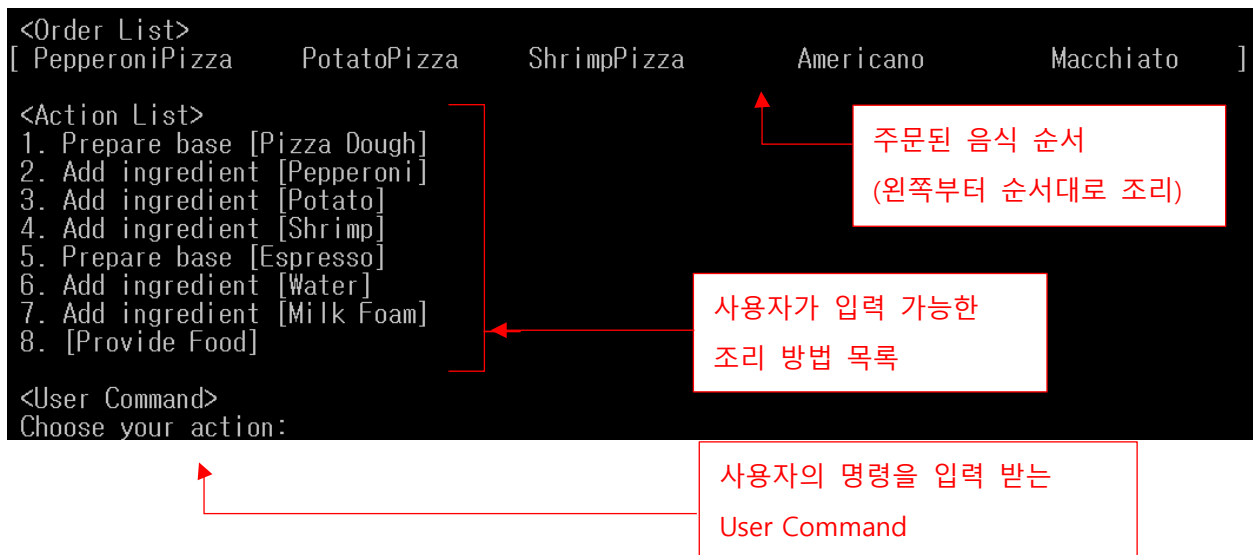
## Food Truck Simulation

Due date: 2018년 11월 17일 토요일 오후 11:59

본 과제에서는 **다형성(polymorphism)**을 사용하여 음식 준비 시뮬레이션을 구현한다.

### ● 음식 준비 시뮬레이션 설명

- 음식 준비 시뮬레이션은 사용자가 주문된 음식 순서에 맞춰 음식을 빠르고 정확하게 조리하여 제공하는 시뮬레이션을 말한다. 사용자는 미리 각 음식의 조리법을 숙지하여야 한다. 사용자가 얼마나 잘 숙지하였는지 시간과 정확도를 측정하는 것이 이 시뮬레이션의 목적이다.
- 음식 종류는 5가지(페페로니 피자, 포테이토 피자, 쉬림프 피자, 아메리카노, 마끼아또)로 고정되어 있다. 상세 조리법과 순서는 본문 내용에 자세히 설명되어 있다.
- 시뮬레이션 실행 화면에는, 주문된 음식이 차례대로 표시되어 있는 **Order List**, 사용자가 입력 가능한 조리방법 목록이 적혀 있는 **Action List**, 사용자의 명령을 입력받는 **User Command**가 표시된다.



- 조리법에 맞게 첫 음식(현재 Pepperoni Pizza)를 조리하는 모습은 아래와 같다.

```
<Order List>
[ PepperoniPizza    PotatoPizza    ShrimpPizza    Americano    Macchiato    ]

<Action List>
1. Prepare base [Pizza Dough]
2. Add ingredient [Pepperoni]
3. Add ingredient [Potato]
4. Add ingredient [Shrimp]
5. Prepare base [Espresso]
6. Add ingredient [Water]
7. Add ingredient [Milk Foam]
8. [Provide Food]

<User Command>
Choose your action: 1
: You added base [Pizza Dough]
Choose your action: 2
: You added ingredient [Pepperoni]
Choose your action: 3
: [Wrong] You should not add ingredient [Potato]
Choose your action: 8
```

현재 Pepperoni Pizza 조리

사용자가 번호로 Action List에서 명령을 선택하는 모습

- 사용자는 위와 같이 첫 음식을 제공한 후, 다음 음식을 조리한다. Order List가 왼쪽으로 Push 된다. User Command에는 이전 History가 사라진다. 예시 화면은 아래와 같다.

```
<Order List>
[ PotatoPizza    ShrimpPizza    Americano    Macchiato    ]

<Action List>
1. Prepare base [Pizza Dough]
2. Add ingredient [Pepperoni]
3. Add ingredient [Potato]
4. Add ingredient [Shrimp]
5. Prepare base [Espresso]
6. Add ingredient [Water]
7. Add ingredient [Milk Foam]
8. [Provide Food]

<User Command>
Choose your action:
```

현재 Potato Pizza 조리

User Command 초기화

- 마지막 음식이 조리된 이후 최종 결과 화면은 다음과 같다. 항목에 관한 자세한 설명은 본문에 설명되어 있다.

```
<Result>
# of correctly provided food: 4
# of wrongly provided food: 1
Minimum # of selection: 11
Your # of selection: 15
Time you spent: 25
```

- 프로그램 전체 흐름

본 프로그램은 모드 선택, 시뮬레이션, 최종 결과 세 단계로 나뉜다.



- 모드 선택

시뮬레이션을 진행하기 전, 사용자는 음식의 순서가 생성되는 방법을 선택할 수 있다. 사용자는 숫자 1, 2로 선택한다. 모드는 1.정해진 순서 모드(Static Order List), 2.랜덤순서 모드(Random Order List)로 구성된다. 모드가 반영된 결과는 '시뮬레이션'에 설명되어 있다.

```
<Food Truck Simulation>
Mode 1: Static Order List
Mode 2: Random Order List
Choose simulation mode:
```

- 시뮬레이션

음식 준비 시뮬레이션의 화면은 Order List, Action List, User Command로 구성된다.

```
<Order List>
[ PepperoniPizza    PotatoPizza    ShrimpPizza    Americano    Macchiato    ]

<Action List>
1. Prepare base [Pizza Dough]
2. Add ingredient [Pepperoni]
3. Add ingredient [Potato]
4. Add ingredient [Shrimp]
5. Prepare base [Espresso]
6. Add ingredient [Water]
7. Add ingredient [Milk Foam]
8. [Provide Food]

<User Command>
Choose your action: 1
```

1. Order List

Order List는 프로그램 가장 상단에 위치하며, 현재 주문된 음식을 나타낸다. 음식은 총 5종류로 고정되어 있고, 총 주문 목록 수는 5가지로 고정되어있다.

사용자가 선택한 '모드'에 따라 Order List가 다음과 같이 달라진다.

- **Static Order List 모드**인 경우 다음과 같은 순서의 Order List 생성

[ PepperoniPizza   PotatoPizza   ShrimpPizza   Americano   Macchiato]

- **Random Order List 모드**를 선택한 경우 5종류의 음식이 Random으로 생성(중복 가능)

Order List가 가져야 할 속성은 다음과 같다.

- 가장 먼저 준비해야 할 음식이 왼쪽부터 순서대로 위치한다.
- 음식 제공이 완료된 후, 왼쪽부터 순차적으로 사라진다.
- 마지막 음식 제공 후 '최종 결과'가 출력된다.

## 2. 음식 설명

음식은 Base와 Ingredient의 조합으로 이루어진다. Pizza와 Coffee에 해당하는 Base와 Ingredient는 다음과 같다.

Pizza	Base	Ingredient		
	Dough	Pepperoni	Potato	Shrimp
Pepperoni Pizza	○	○	×	×
Potato Pizza	○	×	○	×
Shrimp Pizza	○	○	×	○

Coffee	Base	Ingredient	
	Espresso	Water	Milk Foam
Americano	○	○	×
Macchiato	○	×	○

음식은 다음과 같은 속성을 지닌다.

- **Base는 Ingredient보다 먼저 추가되어야 한다.** 이를 지키지 않을 시, 오류 메시지가 출력된다.
- Base가 정해진 후, Ingredient의 추가 순서는 상관이 없다.

### 3. Action List

Action List는 함수 리스트다. 사용자는 User Command에서 선택할 수 있다.

```
<Action List>
1. Prepare base [Pizza Dough]
2. Add ingredient [Pepperoni]
3. Add ingredient [Potato]
4. Add ingredient [Shrimp]
5. Prepare base [Espresso]
6. Add ingredient [Water]
7. Add ingredient [Milk Foam]
8. [Provide Food]
```

함수 종류	함수 속성
Base 함수	<ul style="list-style-type: none"><li>- 1, 5가 해당</li><li>- 음식의 속성을 반영하여 출력 예) You added base [Pizza Dough]</li><li><b>[에러 경우]</b><ul style="list-style-type: none"><li>- 옳은 Base 함수 선택 이후, Base 함수 선택한 경우 예) [Wrong] You already added base [Pizza Dough]</li><li>- 잘못된 Base 함수 선택한 경우 예) [Wrong] Wrong Base</li></ul></li></ul>
Ingredient 함수	<ul style="list-style-type: none"><li>- 2, 3, 4, 6, 7 해당</li><li>- 음식의 속성을 반영하여 출력 예) You added ingredient [Pepperoni]</li><li><b>[에러 경우]</b><ul style="list-style-type: none"><li>- Base 함수보다 먼저 선택한 경우 예) [Wrong] Select Base First</li><li>- Ingredient 함수 중복으로 선택한 경우 예) [Wrong] You already added ingredient [Potato]</li><li>- 잘못된 Ingredient 함수 선택한 경우 예) [Wrong] You should not add ingredient [Potato]</li></ul></li></ul>
[Provide Food] 함수	<ul style="list-style-type: none"><li>- 8 해당</li><li>- 모든 순간 항상 실행 가능, 어떤 메시지도 출력하지 않음</li><li>- Order List에 다음 음식이 있을 경우 User Command의 이전 History는 없어짐</li><li>- Order List에 다음 음식이 없을 경우 결과 화면 전환</li><li><b>[에러 경우]</b><ul style="list-style-type: none"><li>- 없음</li></ul></li></ul>

#### 4. User Command

위에서 설명한 함수들을 User Command 란에서 선택 가능.

- 최종결과

주문 목록에 있던 모든 음식을 제공하고 난 후, 다음과 같은 최종 결과가 출력된다.

```
<Result>
# of correctly provided food: 4
# of wrongly provided food: 1
Minimum # of selection: 11
Your # of selection: 15
Time you spent: 25
```

항목	항목 설명
# of correctly provided food	[Provide Food]시 각 Base, Ingredient에 따라 <u>맞게</u> 제공된 음식의 수
# of wrongly provided food	[Provide Food]시 각 Base, Ingredient에 따라 <u>틀리게</u> 제공된 음식의 수
Minimum # of selection	모든 음식이 제공되기까지 선택해야 하는 <u>최소 선택의 개수</u> 를 나타낸다.
Your # of selection	모든 음식이 제공되기까지 <u>사용자의 선택 개수</u> 를 나타낸다.
Time you spent	시뮬레이션(모드 선택 이후) 실행 시 처음부터 마지막 음식이 제공되기까지 사용된 시간을 초(Second)단위로 나타낸다.

- 기타 사항

- 전체 class를 구성할 시, abstract class가 사용되어야 한다.
- 모든 선택 과정에서의 예러는 [Provide Food]함수의 경우를 제외하고 모두 출력되어야 한다.