

# CSED101. Programming & Problem solving

## Fall 2023

### Programming Assignment #3 (75 points)

김민수 (km19809@postech.ac.kr)

■ 제출 마감일: 2023.11.30 23:59

■ 파이썬 버전: Python 3.x

#### ■ 제출물

- .py 소스 코드 (assn3.py)
  - 프로그램의 소스 코드에 채점자의 이해를 돕기 위한 주석을 반드시 붙여주세요.
- 보고서 파일 (.docx, .hwp 또는 .pdf; assn3.docx, assn3.hwp 또는 assn3.pdf)
  - 보고서는 AssnReadMe.pdf를 참조하여 작성하시면 됩니다.
  - **명예 서약 (Honor code)**: 표지에 다음의 서약을 기입하여 제출해 주세요: “나는 이 프로그래밍 과제를 다른 사람의 부적절한 도움 없이 완수하였습니다.” 보고서 표지에 명예 서약이 기입되어 있지 않은 과제는 제출되지 않은 것으로 처리됩니다.
  - 작성한 소스 코드와 보고서 파일은 PLMS를 통해 제출해 주세요.

#### ■ 주의 사항

- 구문 오류(Syntax Error)가 발생하거나 실행이 되지 않는 과제는 0점으로 채점됩니다.
- 제출 기한보다 하루 늦게 제출된 과제는 최종 20%, 이틀 늦게 제출된 과제는 최종 40% 감점됩니다. 제출 기한보다 사흘 이상 늦으면 제출 받지 않습니다 (0점 처리).  
늦은 제출시 PLMS에 기록된 최종 수정일시를 기준으로 감점합니다.
- 각 문제의 제한 조건과 요구 사항을 반드시 지켜 주시기 바랍니다.
- 모든 문제의 출력 형식은 채점을 위해 아래에 제시된 예시들과 최대한 비슷하게 작성해 주세요.
- 이번 과제에서는 추가 기능 구현에 대한 추가 점수는 없습니다.
- 부정행위에 관한 규정은 POSTECH 전자컴퓨터공학부 학부위원회의 “POSTECH 전자컴퓨터공학부 부정행위 정의”를 따릅니다 (PLMS의 본 과목 공지사항에 등록된 글 중, 제목이 [document about cheating]인 글에 첨부되어 있는 disciplinary.pdf를 참조하세요).

## ■ Problem: 포스몬 배틀(Posmon Battle)

### [문제]

포스몬 배틀은 유명한 수집형 RPG게임, 포켓몬스터의 배틀 요소를 간단히 한 게임입니다. 클래스와 OOP를 활용해 배틀을 구현해봅니다.

### [목적]

이번 과제에서는 객체지향 프로그래밍(Object Oriented Programming)의 핵심 요소를 익힙니다.

- 클래스 정의 및 인스턴스 생성을 익힙니다.
- 클래스 상속 및 메서드 오버라이딩을 익힙니다.

### [주의사항]

- (1) 문서에 반드시 구현해야 하는 함수와 클래스가 존재합니다. 잘 확인하고 설명된 요구사항에 맞게 작성해 주세요.
- (2) 구현 시, 표준 라이브러리만 사용하여 구현해주세요. (추가 설치한 모듈 이용 불가능)
- (3) 명시된 예외 처리 외에는 고려하지 않아도 됩니다.

### [설명]

#### 0. 규칙 설명

- (1) 포스몬 배틀은 최대 3마리의 포스몬(Posmon)을 사용하여 상대의 포스몬을 전부 쓰러뜨리면 이기는 게임입니다. 본 과제에서는 두 명의 플레이어(컴퓨터, 사용자)의 대결로 구현합니다.
- (2) 배틀은 1:1배틀을 기본 규칙으로 합니다. 각 플레이어는 배틀 시, 소유한 포스몬 중 한 마리를 선택하여 배틀에 내보내어 대결을 하게 합니다.
- (3) 플레이어는 포스몬 보유기술과 포스몬 교대를 적절히 활용하여 상대의 포스몬을 모두 쓰러뜨려야 합니다.
- (4) 각 포스몬은 타입(posmon\_type), 체력(health), 공격력(attack), 방어력(defence) 수치를 가지며, 체력이 0이 되면 쓰러집니다.
- (5) 포스몬의 타입에는 가위("Scissors"), 바위("Rock"), 보("Paper"), 없음("Nothing")이 존재합니다. 상대보다 강한 속성일 경우 주는 피해는 2배가 됩니다.
- (6) 각 포스몬은 최대 3개의 기술(Move)을 가질 수 있으며, 기술은 상대의 체력을 깎는 공격 기술(PhysicalMove)과 자신 또는 상대의 공격력, 방어력을 변화시키는 변화 기술(StatusMove)로 구성됩니다.
- (7) 기술에는 속도(speed)가 존재하며, 속도가 빠른 기술이 먼저 사용됩니다. 즉, 배틀 시 속도가 빠른 기술을 선택한 포스몬이 먼저 기술을 사용합니다. 동점일 경우 사용자의 기술이 먼저 사용됩니다.
- (8) 공격 기술에는 추가로 위력(power)이 존재해서, 그 위력만큼 상대에게 피해를 줍니다. 아래 피해 계산식을 참고하세요.

포스몬의 체력 피해 계산식은 다음과 같습니다.

$$\text{상대의 체력 피해} = \max(0, \text{기술의 위력} + \text{자신의 공격력} - \text{상대 방어력}) * \text{상성 배율}$$

상성 배율은 다음과 같이 계산됩니다.

공격자 : 방어자	가위	바위	보	없음
가위	1	1	2	1
바위	2	1	1	1
보	1	2	1	1
없음	1	1	1	1

즉, 가위 바위 보에서 이기는 경우에만 2배의 피해를 입히고, 그 외에는 항상 같은 피해를 줍니다.

## 1. 게임 준비

포스몬 배틀을 하려면 조건에 맞게 포스몬을 준비해야 합니다. 각 클래스를 구현할 때엔 '**5.필수 구현 항목**'을 참고하고, 아래의 조건에 맞도록 포스몬을 준비합니다.

### (1) 포스몬의 종류

- 포스몬의 종류는 구현의 편의상 4종으로 합니다.
- 각 포스몬의 타입과 기본 능력치(체력, 공격력, 방어력), 보유 기술은 다음과 같습니다.

포스몬(클래스 이름)	타입	체력	공격력	방어력	보유 기술
Ponix	Paper	86	20	23	Tackle, Growl, SwordDance
Normie	Nothing	80	20	20	Tackle, Swift, TailWhip
Rocky	Rock	80	15	25	Tackle, Growl
Swania	Scissors	80	30	10	ScissorsCross, SwordDance

- 각 포스몬은 "Rock", "Paper", "Scissors", "Nothing" 중 하나의 타입을 가집니다.
- 포스몬마다 사용할 수 있는 기술이 2~3 가지씩 있습니다.

### (2) 포스몬의 기술

- 포스몬의 기술은 공격 기술과 변화 기술로 나눌 수 있습니다. 구현의 편의상 6종으로 합니다.
- 모든 기술의 속도는 -3 이상 3 이하입니다. (숫자가 클수록 빠르며, 동물이면 사용자가 우선합니다.)

#### A. 공격 기술: 상대의 체력을 감소시킵니다.

기술명(클래스 이름)	위력	속도
몸통박치기(Tackle)	25	0
시저크로스(ScissorsCross)	30	0
스피드스타(Swift)	0	3

#### B. 변화 기술: 자신/상대의 공격력/방어력을 변화시킵니다.

기술명(클래스 이름)	use 효과	속도
울음소리(Growl)	상대 공격력 5 감소	1
칼춤(SwordDance)	자신 공격력 10 증가	0
꼬리 흔들기(TailWhip)	상대 방어력 5 감소	1

## 2. 프로그램 시작 화면

프로그램을 실행하면, 다음과 같은 타이틀 화면과 함께 3개의 선택사항이 있는 메뉴를 출력하고, 사용자로부터 3가지 선택사항 중 하나를 입력 받도록 합니다. 예시의 빨간색 밑줄은 사용자 입력에 해당합니다.

```
| | | \ / \ / \ / \ / | T T / \ | Y Y | \ / |
| o ) Y   Y( \ _ | - \ - | Y Y | _ | Y
| _ / | 0 | \ _ | T | \ / | 0 | _ | | | | | | | | |
| | | | | | | / \ | | | | | | | | | 
| | | 1 | ! \ | | | | | 1 | ! | 
1_j    \_ /    \_ j 1_j_j \_ / 1_j_j
```

=====

0. 포스몬 선택  
1. 배틀하기  
2. 종료하기

=====

입력: 4  
잘못된 입력입니다. 다시 입력하세요.  
입력:

**예외처리)** 0, 1, 2 외의 입력에 대해서는 "잘못된 입력입니다. 다시 입력하세요." 메시지 출력 후 위의 예시처럼 다시 입력 받습니다.

### 3. 포스몬 선택(초기 메뉴에서 0을 선택시)

사용자가 0을 선택하면 아래와 같이 배틀에서 사용할 사용자의 포스몬을 선택할 수 있는 화면으로 이동합니다. 먼저 선택 가능한 포스몬 목록이 출력되고, 사용자의 선택을 기다립니다.

```
0. 포스몬 선택
1. 배틀하기
2. 종료하기
=====
입력: 0

=====
당신이 사용할 포스몬을 선택하세요. 현재 0 마리/최대 3 마리
0. Ponix
1. Normie
2. Swania
3. Rocky
=====
입력:
```

사용자는 포스몬을 최소 1 마리, 최대 3 마리 선택할 수 있으며, 동일한 종류의 포스몬 선택도 가능합니다. (동일한 종류의 포스몬 선택 시, 같은 객체가 되지 않도록 주의합니다. 생성자를 사용할 필요가 있습니다.)

1 마리를 고르고 나면, "-1. 그만두기" 항목이 나타나 포스몬 선택을 끝낼 수 있습니다. 아래 예시는 포스몬 2마리 선택 후 -1을 입력하여 포스몬 선택을 끝낸 예시이며, 선택이 끝난 경우 선택한 포스몬의 목록을 출력하고 초기 메뉴 화면으로 돌아갑니다.

포스몬 3마리를 모두 선택한 경우, 선택한 포스몬 목록 출력 후 초기 메뉴 화면으로 돌아갑니다.

예외처리) - 0~4 외의 입력은 에러 메시지 출력 후 다시 입력 받습니다.  
- 아무 포스문도 고르지 않은 경우, -1은 잘못된 입력입니다.

0. 포스몬 선택

1. 배틀하기

2. 종료하기

입력: 0

당신이 사용할 포스몬을 선택하세요. 현재 0 마리/최대 3 마리

0. Ponix

1. Normie

2. Swania

3. Rocky

입력: 0

당신이 사용할 포스몬을 선택하세요. 현재 1 마리/최대 3 마리

0. Ponix

1. Normie

2. Swania

3. Rocky

-1. 그만두기

입력: 3

당신이 사용할 포스몬을 선택하세요. 현재 2 마리/최대 3 마리

0. Ponix

1. Normie

2. Swania

3. Rocky

-1. 그만두기

입력: 5

잘못된 입력입니다. 다시 입력하세요.

입력: -1

당신의 포스몬 목록: Ponix Rocky

```

|   \ /   \ /   |   T   T /   \ |   \
|   o )Y   Y(   \ |   -   |Y   Y|   - Y
|   _/ |   0   |   \   T   \ /   ||   0   ||   | | | |
|   |   |   |   |   /   \   ||   |   ||   |
|   |   1   !   \   |   |   |   ||   1   !   |
|   l_j   \_/_   \_/_j_l_j_j   \_/_   l_j_j

```

0. 포스몬 선택

1. 배틀하기

2. 종료하기

입력:



Ponix	< Paper 86 / 86	# 현재 배틀에 나온 사용자 포스몬 정보
당신의 포스몬: [00] 2 / 2		# 사용자가 보유한 포스몬들의 상태 정보
+++++		
기술: (0) Tackle (1) Growl (2) SwordDance		# 현재 배틀에 나온 사용자 포스몬의 기술
#####		
입력:		

### (1) 배틀 화면 출력

배틀하기를 선택하면, 위의 예시와 같이 컴퓨터, 사용자 모두 목록의 첫 번째 포스몬이 선택되어 나옵니다.

- ① 컴퓨터와 사용자가 보유한 포스몬들의 상태 정보는 아래와 같이 표현됩니다.

[{매 포스몬마다 포스몬이 쓰러졌으면 x, 아니면 0}] {남은 포스몬 수} / {총 포스몬 수}

예를 들어, 컴퓨터의 세 포스몬 중 두 번째 포스몬이 쓰러졌다면 "컴퓨터 포스몬: [0X0] 2 / 3"으로 출력됩니다.

- ② 현재 배틀에 나온 각 포스몬 정보는 아래와 같이 표현됩니다. 포스몬 이름과 <| 사이 공백의 수는 하나 이상이면 됩니다.

{포스몬 이름} <|{포스몬 타입} {현재 체력}/{최대 체력}|

### (2) 사용자 순서

사용자는 매 차례, 해당 포스몬의 보유 기술과 포스몬 교대 중 하나를 선택합니다. 그때 사용하는 명령어는 다음과 같습니다.

- ① 포스몬 확인 (entry)

**e**: 내 포스몬의 숫자, 이름 및 체력을 확인합니다. 차례를 끝내지 않습니다.

- ② 기술 사용 (order)

**o 숫자**: 숫자 번째 (0 부터 시작) 기술을 사용합니다. 차례를 끝냅니다. 예) 'o 1'

- ③ 포스몬 교대 (switch out)

**s 숫자**: 숫자 번째 (0 부터 시작) 포스몬과 교대합니다. 차례를 끝냅니다. 예) 's 2'

- 포스몬 교대는 반드시 상대의 기술보다 먼저 사용됩니다.
- 포스몬 교대를 했으므로 해당 순서에는 교대한 포스몬의 기술을 사용할 수 없습니다. 다음 차례부터 기술 선택 및 사용이 가능합니다.
- 자기 자신 또는 이미 쓰러진 포스몬과는 교대할 수 없습니다.

### (3) 컴퓨터 순서

- ① 컴퓨터 포스몬의 기술 선택

사용자가 기술 선택(또는 교대 선택)을 한 경우, 현재 대결 중인 컴퓨터의 포스몬도 보유 기술 중 하나를 랜덤하게 선택하도록 한다.

- ② 컴퓨터의 교대

컴퓨터 포스몬의 교대는 해당 포스몬이 쓰러진 경우에만 가능하며, 이때 다음 순서의 포스몬이 선택되어 나옵니다. (목록의 순서대로 나옴)

#### (4) 1:1 배틀(선택 기술 적용)

두 포스몬의 기술 선택이 끝나면 1:1 배틀이 진행됩니다.

- 기술 속도를 비교하여, 기술의 속도가 더 빠른 포스몬이 먼저 공격을 합니다. 동률이면 사용자의 포스몬이 먼저 공격합니다.
- 사용자 입력으로 선택한 '포스몬 교대' 시, 상대(컴퓨터) 기술만 교대한 포스몬에게 적용됩니다.

한 플레이어의 모든 포스몬이 쓰러질 때까지 (1) ~ (4)가 반복됩니다.

#### ● 기술 사용 예시 1)

아래 예시는 사용자가 변화 기술(Growl)을 사용한 예시로 차례의 진행결과를 출력합니다.

```
배틀이 시작됩니다.
#####
컴퓨터 포스몬: [000] 3 / 3
Normie          <|Nothing 80 / 80|
                VS
Ponix           <|Paper 86 / 86|
당신의 포스몬: [00] 2 / 2
+++++
기술: (0) Tackle (1) Growl (2) Idle
#####
입력: 0.1 # Ponix 기술 중 (1) Growl 선택
#####
컴퓨터 Normie: Swift 기술 사용
- 당신 포스몬의 [체력] 3 감소 (86 -> 83)
당신의 Ponix: Growl 기술 사용
- 컴퓨터 포스몬의 [공격력] 5 감소 (20 -> 15)

#####
컴퓨터 포스몬: [000] 3 / 3
Normie          <|Nothing 80 / 80|
                VS
Ponix           <|Paper 83 / 86|
당신의 포스몬: [00] 2 / 2
+++++
기술: (0) Tackle (1) Growl (2) Idle
#####
입력:
```

##### ① 선공 선택 및 기술 사용

첫 번째 차례에서, 컴퓨터 포스몬(Normie)은 보유 기술 중 "스피드스타(Swift)"를 선택했고, 사용자는 사용자 포스몬(Ponix)의 보유 기술 중 "울음소리(Growl)"를 선택했습니다. 이때 기술의 속도가 빠른 컴퓨터의 포스몬이 공격권을 가지고 먼저 기술이 사용되고, 이후, 사용자의 기술이 사용된 것을 볼 수 있습니다.

##### ② 사용된 기술 이름과 효과 출력

A. 기술의 이름을 출력할 때는 다음 양식을 따릅니다.

{컴퓨터|당신의} {포스몬 이름}: {기술 이름} 기술 사용

B. 기술의 효과를 출력할 때는 다음 양식을 따릅니다.



- {컴퓨터|당신} 포스몬의 [{체력|공격력|방어력}] {변화량} {증가|감소} ({변화 이전 값} -> {변화 이후 값})

사용된 기술의 처리가 끝나면, 프로그램은 다시 "입력:"이라는 프롬프트와 함께 다음 명령어를 선택하기를 기다립니다.

## ● 교대 실행 예시

다음 예시는 사용자가 보유한 포스몬을 확인하고, 교대시킨 경우입니다.

```
입력: e # 사용자가 보유한 포스몬 확인을 위해 e 입력
#####
(0) Ponix <|Paper 83 / 86|
(1) Rocky <|Rock 80 / 80|

입력: s 1 # 사용자의 포스몬을 (1) Rocky 로 교대
#####
당신의 포스몬 Rocky 로 교대
컴퓨터 Normie: Swift 기술 사용
- 당신 포스몬의 [체력] 0 감소 (80 -> 80)

#####
컴퓨터 포스몬: [000] 3 / 3
Normie <|Nothing 80 / 80|
VS
Rocky <|Rock 80 / 80|
당신의 포스몬: [00] 2 / 2
+++++
기술: (0) Growl (1) Tackle
#####
입력:
```

### ① 포스몬 확인

사용자가 'e 명령어'를 입력하여 자신이 보유한 포스몬을 확인합니다. 이때는 대결이 진행되지는 않습니다.

### ② 교대

- 사용자가 교대 명령어 's 1'을 입력하여 대결 포스몬을 Ponix 에서 Rocky 로 교대합니다. 포스몬 교대를 선택한 경우에는 포스몬 교대만 가능하고 해당 순서에서는 기술을 선택하여 사용할 수 없습니다.
- 컴퓨터가 선택한 기술이 교대한 포스몬 Rocky 에게 적중합니다.
- 교대 상황을 출력할 때는 다음 양식을 따릅니다.

당신의 포스몬 {교대한 포스몬 이름}로 교대

- 교대할 경우, 돌아온 포스몬의 공격력과 방어력은 생성 시의 능력치로 되돌아갑니다. 체력은 유지됩니다. 예시: 상대의 "울음소리"에 당해 공격력이 떨어진 "Ponix"는 교대될 때 공격력이 다시 20 이 됨.

## ● 기술 사용 예시 2)

다음 예시는 사용자가 공격 기술로 상대를 쓰러뜨린 예시입니다.

```
Rocky                <|Rock    80 / 80|
당신의 포스몬: [00] 2 / 2
+++++
기술: (0) Growl (1) Tackle
#####
입력: 0 1
#####
당신의 Rocky: Tackle 기술 사용
- 컴퓨터 포스몬의 [체력] 20 감소 (5 -> -15)
컴퓨터 Normie: 쓰러짐
컴퓨터 포스몬 Swania 로 교대

#####
컴퓨터 포스몬: [X00] 2 / 3          # 첫번째 포스몬 쓰러진 것 반영
Swania                <|Scissors 80 / 80|
VS
Rocky                <|Rock    80 / 80|
당신의 포스몬: [00] 2 / 2
+++++
기술: (0) Growl (1) Tackle
#####
입력:
```

### ③ 선공 선택 및 기술 사용

- A. 배틀에 나온 두 포스몬은 공격을 위해 각각 하나의 기술을 선택하게 됩니다. 이때 기술의 속도가 더 빠른 포스몬이 먼저 공격을 하며, 동률이면 사용자의 포스몬이 먼저 공격합니다.

위 예시에서는 컴퓨터 Normie와 사용자 Rocky의 대결에서 사용자는 Tackle 기술을, 컴퓨터도 Normie 보유 기술 중 Tackle 기술을 선택했습니다. 기술 속도가 같아 사용자 Rocky가 공격권을 가집니다.

- B. 공격권을 먼저 가진 포스몬이 자신이 선택한 기술을 상대방에게 사용합니다. 위 예시는 공격기술 Tackle이 선택되어, 2 쪽 상대 체력 피해 계산식에 따라 계산하여 상대방을 공격합니다.

위 예시에서는 사용자 Rocky는 계산식  $\max(0, 25(\text{Tackle 위력}) + 15(\text{Rocky의 공격력}) - 20(\text{Normie의 방어력})) \times 1(\text{상성 배율})$ 의 결과로 컴퓨터 Normie의 체력을 20 감소 시킵니다.

- C. 다음 상대 포스몬이 선택한 기술에 따라 상대방에게 사용합니다.

- 포스몬이 기술을 사용하기 전 쓰러진 경우, 기술을 사용하지 않고 차례가 끝납니다. 포스몬이 쓰러질 때 출력 양식은 다음과 같습니다.

{컴퓨터|당신의} {포스몬 이름}: 쓰러짐

- 포스몬이 쓰러지면 남은 포스몬 중 가장 처음의 포스몬으로 자동으로 교대합니다. 이때 출력되는 양식은 다음과 같습니다.

{컴퓨터|당신의} {교대한 포스몬}로 교대

- 피해 공식을 디버깅하기 쉽도록, 체력은 음수가 될 수 있습니다. 하지만 그 외의 값, 피해량, 공격력, 방어력은 음수가 될 수 없습니다.

## ● 배틀 종료 예시

포스몬이 모두 쓰러진 경우 배틀이 종료됩니다. 아래와 같이 메시지를 출력하고 초기 메뉴로 돌아갑니다.

컴퓨터 Ponix: 쓰러짐

```
#####
컴퓨터 포스몬: [XXX] 0 / 3
Ponix                <|Paper 0 / 86|
                    VS
Ponix                <|Paper 79 / 86|
당신의 포스몬: [00] 2 / 2
+++++
기술: (0) Tackle (1) Growl (2) Idle
#####
```

[배틀 결과] 당신이 이겼습니다!

```

|   \   /   \   /   |   T   T   /   \   |   \
| o )Y    Y(   \   |   -   |Y    Y|   -   Y
|   /   | 0   |   \   T   |   /   |   0   |   | | | | |
|   |   |   |   /   \   |   |   |   |   |   |
|   |   |   |   |   |   |   |   |   |   |
|   |   |   |   |   |   |   |   |   |   |
|_j   \_/_   \_/_j_l_j_j   \_/_   l_j_j
=====
```

```
0. 포스몬 선택
1. 배틀하기
2. 종료하기
=====
```

입력:

승리/패배 메시지는 포스몬 배틀이 끝난 후 타이틀 메뉴로 돌아가기 전 출력됩니다.

승리 시에는 "[배틀 결과] 당신이 이겼습니다."를, 패배 시에는 "[배틀 결과] 컴퓨터가 이겼습니다."를 출력합니다.

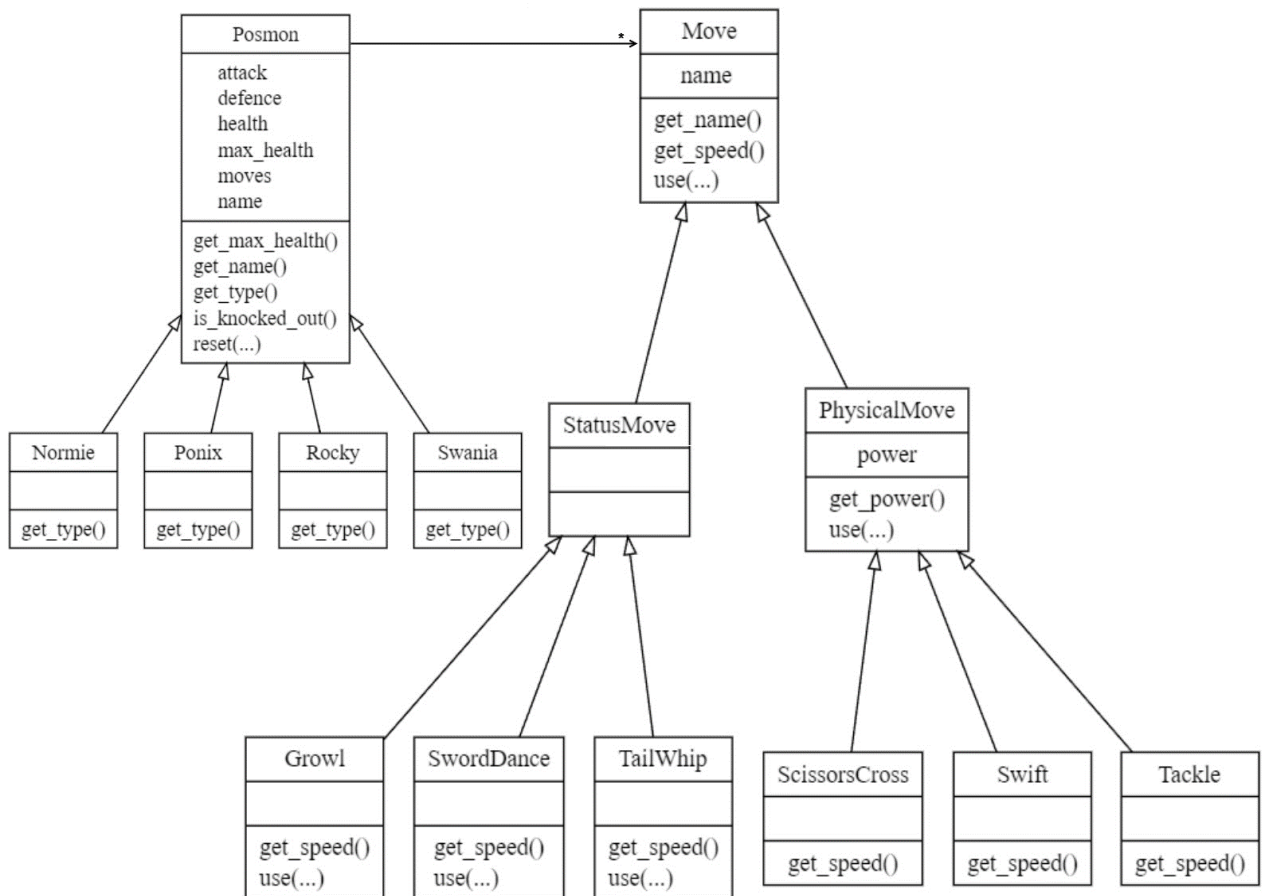
배틀 종료와 동시에 플레이어의 보유 포스몬은 모두 사라집니다. 즉, 배틀하기를 위해서는 포스몬 선택부터 다시 시작해야 합니다.

**예외처리)** 배틀에서 사용자 입력 시, 범위를 벗어난 숫자를 입력하거나, 현재 포스몬 또는 쓰러진 포스몬을 교대하면 아래 예시와 같은 에러와 함께 다시 행동을 선택하게 됩니다. (이 때 차례는 끝나지 않습니다.) 아래 예시는 당신의 3 번째 포스몬이 현재 포스몬이라고 가정합니다.

```
입력: s 2
포스몬을 교대시킬 수 없습니다!
입력: o 4
선택할 수 없는 기술입니다!
입력: m 1
잘못된 명령어: m 1
입력:
```

## 5. 필수 구현 사항

### [클래스 구현 시 요구사항]



본 과제에서 사용되는 상속 구조 및 연관 관계도는 위 그림과 같습니다.

### [클래스 변수 및 메서드]

다음 클래스 변수 및 메서드는 반드시 작성하여 프로그램을 구현해야 합니다. 아래 명시된 변수 및 메서드 이름은 변경하지 말아주세요. 메서드의 매개변수의 개수 및 리턴 값 등은 자유롭게 변경 가능합니다. 그러나 동일한 기능을 하는 메서드는 반드시 있어야 하며, 변경 시 무엇을 어떻게 변경했는지 보고서에 기록하도록 합니다. 이 외에 필요한 메서드는 정의해서 사용할 수 있습니다.

**참고)** "매개변수:자료형" 또는 "-> 반환 자료형" 형태로 작성된 것은 이해를 돕기 위한 것으로, 구현할 때는 적지 않아도 됩니다.

## (1) 포스몬(Posmon) 클래스 정의

```
class Posmon
    # 인스턴스 변수

    self.health      # 포스몬의 체력 (int)
    self.max_health  # 포스몬의 최대(초기) 체력 (int)
    self.attack      # 포스몬의 공격력 (int)
    self.defence     # 포스몬의 방어력 (int)
    self.moves       # 포스몬이 보유한 기술(Move) 리스트 (list of Move)
    self.name        # 포스몬의 이름(str)

    # 메서드
    def get_name(self) -> str :
        # 포스몬의 이름을 반환하는 메서드 (예. 'Ponix' 반환)

    def get_max_health(self) -> int :
        # 포스몬의 최대 체력을 반환하는 메서드

    def get_type(self) -> str :
        # 포스몬의 타입을 반환하는 메서드로,
        # 실제 기능은 서브클래스에서 오버라이딩을 통해 구현합니다.
        # 구현 시, 포스몬의 타입에 따라 "Rock", "Paper", "Scissors", "Nothing" 중
        # 하나를 반환합니다. (대소문자 구분할 것)
        pass

    def reset_status(self, reset_health:bool = False):
        # 포스몬의 공격력과 방어력을 생성될 때 값으로 초기화하는 메서드
        # reset_health가 True 면, 체력도 최대 체력으로 초기화합니다.
```

### (1-2) 포스몬 캐릭터 클래스 정의

3쪽의 '포스몬의 종류' 표를 참고하여 Ponix, Normie, Rocky, Swania 클래스를 생성합니다. 해당 클래스는 Posmon 을 상속받습니다.

예시)

```
class Ponix(Posmon):

    def __init__():
        # super() 함수를 사용하여 Ponix 의 체력을 86, 공격력을 20, 방어력을 23,
        # 이름을 'Ponix', 보유 기술 리스트(표 참고)를 설정합니다.
        # 예) super().__init__(health=86, ...)

    def get_type(self) -> str : # 포닉스의 타입은 "Paper"로 해당 타입 반환
```

## (2) 포스몬 기술(Move) 클래스 정의

Move 클래스는 포스몬 배틀에서 모든 기술의 기본이 되는 베이스 클래스입니다.

Move 클래스는 get\_name 메서드의 기능만 추가 작성 후, 그대로 사용하면 됩니다.

```
class Move:
    def __init__(self, name):
        self.name = name      # 기술 이름(str) (예. 'Tackle')

    def get_name(self)->str : # 기술의 이름을 반환하는 메서드

    def get_speed(self)->int:
        # 기술의 속도를 반환하는 메서드로,
        # 실제 기능은 서브클래스에서 오버라이딩을 통해 구현합니다.
        pass

    def use(self, our_posmon:Posmon, opponent_posmon:Posmon, is_player_move=True):
        # 기술을 사용하는 메서드로,
        # 실제 기능은 서브클래스에서 오버라이딩을 통해 구체적으로 구현합니다.
        pass
```

### (2-1) 공격 기술(PhysicalMove) 클래스 정의

```
class PhysicalMove(Move):
    # 인스턴스 변수
    self.power # 해당 기술의 위력

    # 메서드
    def __init__(self, power, name):
        super().__init__(name)

    def get_power(self) -> int: # 이 기술의 위력을 반환하는 메서드

    def use(self, our_posmon:Posmon, opponent_posmon:Posmon, is_player_move=True):
        # (1) 현재 배틀 중인 포스몬(our_posmon)이 기술을 사용하여
        # 상대편 포스몬(opponent_posmon)에게 피해를 주는 메서드입니다.
        # 예시: 컴퓨터 Normie 가 당신의 Ponix 를 공격하면
        # our_posmon 은 컴퓨터 Normie, is_player_move 는 False.
        # (2) 피해를 줄 때 계산 공식은 '0.규칙 설명'을 사용합니다.
        # (3) 체력 변화에 따른 메시지를 출력합니다.
        print("- %s 포스몬의 [체력] %d 감소 (%d->%d)" ...)
```

#### (2-1-1) 실제 공격 기술 클래스 정의

PhysicalMove 를 상속할 때는 생성자, get\_speed 를 오버라이드하여 구현합니다.

포스몬 캐릭터(Ponix, Normie 등)들이 실제 보유할 수 있는 공격 기술인 몸통박치기(Tackle), 시저크로스(ScissorsCross), 스피드스타(Swift) 클래스를 아래의 표를 참고하여 정의합니다.

기술명(클래스 이름)	위력	속도
몸통박치기(Tackle)	25	0
시저크로스(ScissorsCross)	30	0
스피드스타(Swift)	0	3

예시)

```
class Tackle(PhysicalMove):
    # 메서드
    def __init__(self):
        super().__init__(power=25, name="Tackle")
    def get_speed(self):
        return 0
```

## (2-2) 변화 기술(StatusMove) 클래스 정의

Move 를 상속합니다. 개념적인 클래스이므로, 추가적으로 선언할 속성이나 오버라이드할 메소드는 없습니다.

```
class StatusMove(Move):
    pass
```

### (2-2-1) 실제 변화 기술 클래스 정의

StatusMove 를 상속하는 변화 기술은 공격 기술과 달리 생성자, get\_speed, use 를 오버라이드 해야합니다. 아래 표를 참고하여 3 가지 기술 클래스를 정의합니다.

기술명(클래스 이름)	효과	속도
울음소리(Growl)	상대 공격력 5 감소	1
칼춤(SwordDance)	자신 공격력 10 증가	0
꼬리 흔들기(TailWhip)	상대 방어력 5 감소	1

※ 기술 적용으로 공격력/방어력 감소 시 0 보다 낮아지는 경우는 없습니다.

예시)

```
class Growl(StatusMove):
    # 메서드
    def __init__(self, name = "Growl"): # Growl 기술의 인스턴스 초기화
        # super() 함수 사용하여 코드 채우기
        self.amount = -5

    def get_speed(self) -> int: # Growl 기술의 속도를 반환하는 메서드

    def use(self, our_posmon:Posmon, opponent_posmon:Posmon, is_player_move=True):
        # (1) 현재 배틀 중인 포스몬(our_posmon)이 기술을 사용하여
        #     상대편 포스몬(opponent_posmon)의 공격력을 감소시키는 메서드입니다.
        # (2) 상대 포스몬의 공격력을 감소시키는 로직 구현 (Growl 효과에 맞춰서 구현)
        # (3) 값을 변화시킨 내용을 print를 사용하여 출력
        print("- %s 포스몬의 [공격력] %d 감소 (%d->%d)" ...)
```