

2023년 2학기

프로그래밍과 문제해결

Assignment #3

담당 교수: 윤은영

학번: 20230499

학과: 무은재학부

이름: 김재환

POVIS ID: carotinoid

명에서약(Honor code)

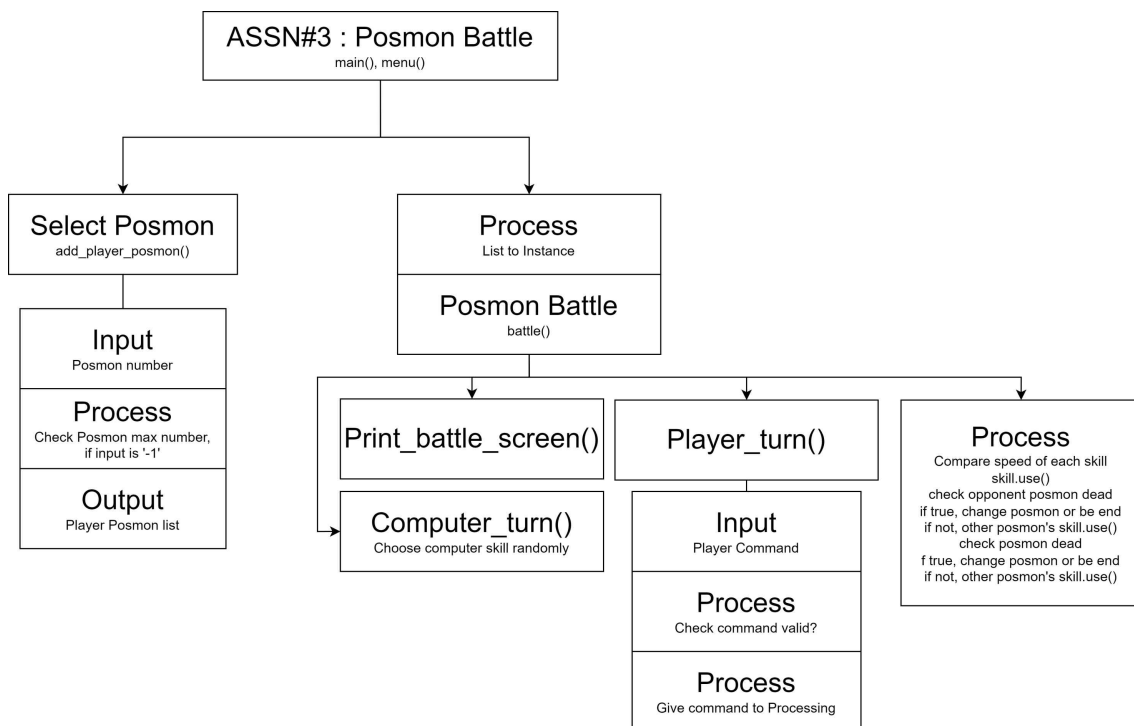
“나는 이 프로그래밍 과제를 다른 사람의 부적절한 도움 없이 완수하였습니다.”

1. 문제의 개요

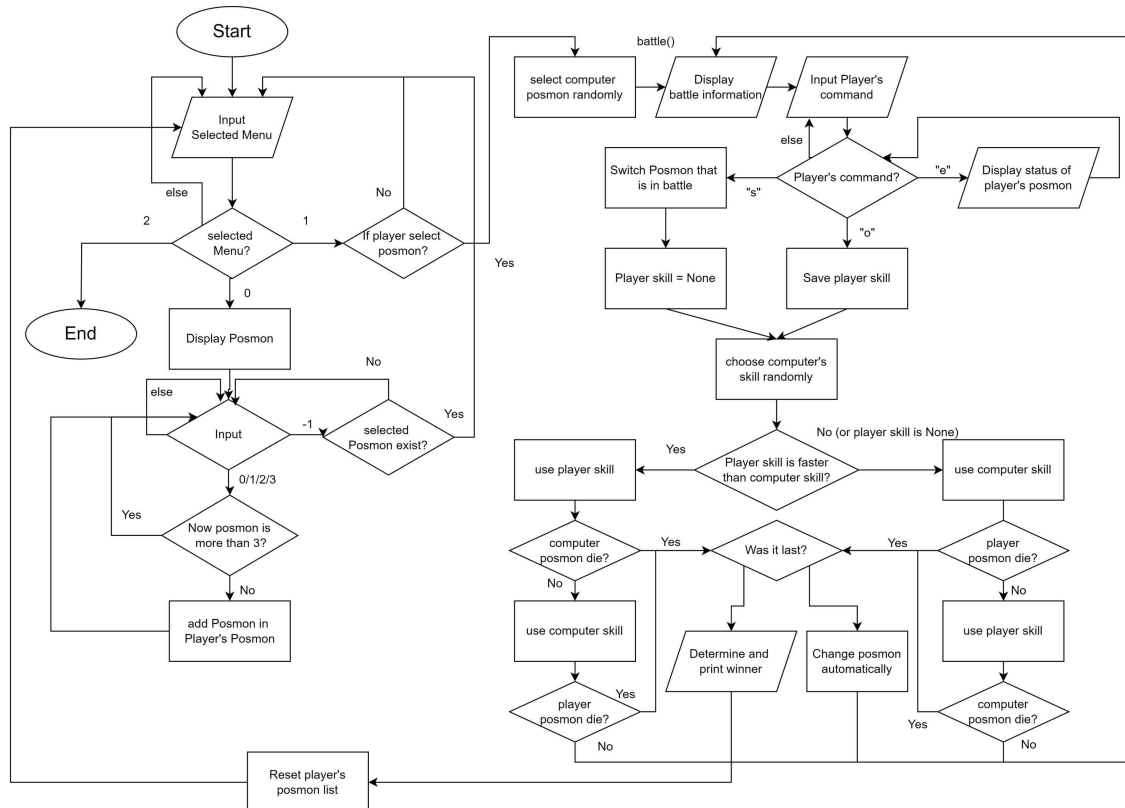
본 프로그램을 간략히 설명하면 다음과 같다.

- 메뉴에서 0을 선택하면, 플레이어가 사용할 포스몬을 선택한다.
- 메뉴에서 2를 선택하면, 프로그램을 종료한다.
- 메뉴에서 1을 선택하면, 플레이어가 선택한 포스몬과 자동으로 선택된 3마리의 컴퓨터 포스몬과 전투한다. 이때 플레이어는 명령을 통해 현재 포스몬 확인 / 전투 중인 포스몬 교체 / 전투 중인 포스몬 기술 사용 3가지 행동을 할 수 있다.
- 포스몬의 기술은 공격 기술과 변화 기술로 분류되며, 공격 기술은 정해진 수식에 따라 상대 포스몬의 체력을 감소시키며, 변화 기술은 상대나 자신의 공격력 또는 방어력을 변화시킨다.
- 포스몬이 교체될 때, 변화 기술로 인해 변화된 공격력과 방어력은 원래 상태로 초기화된다.
- 포스몬이 쓰러지면 자동으로 다음 포스몬으로 교체되며, 남은 포스몬이 없다면 패배한다. 승패가 결정된 이후에는 메뉴로 돌아오며 포스몬 선택을 다시 해야한다.

본 프로그램의 구조 차트(structure chart)는 다음과 같다.



본 프로그램의 흐름도(workflow chart)는 다음과 같다. 다만, 흐름도를 간략하게 작성하기 위해서 대부분의 출력은 생략하였다.



본 프로그램 작성은 클래스 활용법을 익히는 것이 주된 목표이므로, 크게 복잡한 알고리즘은 존재하지 않는다. 그래도 본 프로그램에서 주의해야 할 점을 꼽아보자면, 포스몬을 선택할 때 서로 같은 포스몬이라도 다른 객체를 저장해야 한다는 점과 서로의 행동을 진행할 때 공격으로 인해 한 포스몬이 쓰러질 때의 경우를 잘 고려해야 한다. 공격을 맞고 쓰러진다면 이후 행동은 실행해선 안 되며, 마지막 포스몬일 때에는 게임을 종료해야 한다. 서로의 기술을 진행하면서 가능한 경우의 수를 빠뜨리지 않도록 주의해야 한다.

3. 프로그램 구조 및 설명

a) 메인메뉴

- 포스몬 선택, 배틀 시작, 프로그램 종료 3가지 중 하나를 선택할 수 있다. 입력이 유효하지 않으면 재입력받는다. 0을 받으면 포스몬을 선택하는 단계로 넘어가며, 선택 후 다시 메뉴로 돌아온다. 포스몬을 선택한 후 1을 선택하면 배틀을 시작한다. 2를 선택할 시 프로그램을 종료한다.

b) 포스몬 선택

- 포스몬 선택 화면에서는 0, 1, 2, 3을 선택하여 각 포스몬을 선택할 수 있다. 순서대로 Ponix, Normie, Rocky, Swania를 선택하며, 중복해서 선택할 수 있다. 최대 3마리까지 선택할 수 있다. 선택 후에는 -1을 입력하여 메뉴로 돌아갈 수 있으며, 아무 포스몬도 선택하지 않고 -1을 입력할 시에는 잘못된 입력으로 처리한다.

c) 배틀

- 포스몬을 선택한 후 메뉴에서 1을 입력할 시 배틀을 진행한다.
- 컴퓨터는 랜덤으로 포스몬 3마리를 선택하며, 컴퓨터 또한 중복해서 뽑을 수 있다.
- 처음에는 가장 먼저 뽑은 포스몬이 자동으로 싸우게 된다.
- 각 차례마다 자신이 할 명령을 입력한다. 'e'를 입력할 시 자신이 가지고 있는 전체 포스몬과 그것들의 상태를 출력하며, 차례를 소모하지 않는다.
- s (number) 입력 시 해당 번호의 포스몬과 교체한다. 숫자는 0부터 시작해 2까지 입력할 수 있으며, 현재 싸우고 있는 포스몬의 번호이거나 이미 쓰러진 포스몬의 번호일 경우, 0, 1, 2 이외의 번호일 경우 잘못된 입력으로 처리하여 다시 명령을 입력받는다. 성공적으로 교체하면 차례를 소모하며, 속도와 관계없이 무조건 컴퓨터의 기술보다 먼저 진행한다.
- o (number) 입력 시 해당 번호의 기술을 사용한다. 숫자는 Ponix, Normie, Swania일 경우 0, 1, 2, Rocky의 경우 0, 1 이외의 수는 잘못된 입력을 처리한다. 유효한 기술 사용 명령을 입력받았을 경우 차례가 소비되며, 임의로 선택된 컴퓨터의 기술과 속도를 비교하여 빠른 기술을 먼저 사용한다.
- 속도가 빠른 기술을 사용한 후, 상대 포스몬이 쓰러졌을 경우, 쓰러진 포스몬이 선택한 기술은 사용되지 않으며, 자동으로 다음 번호의 포스몬으로 교체된다. 만약 남은 포스몬이 없는 상황에서 마지막 포스몬이 쓰러졌을 경우, 승자를 결정하고 메뉴로 돌아오면서 플레이어가 선택한 포스몬을 초기화한다.

d) 클래스

- 클래스는 크게 포스몬 클래스와 기술 클래스로 나뉜다.

- 포스몬 클래스를 상속하여 각 포스몬의 세부 능력치를 구현하며, 포스몬 클래스는 인스턴스 변수로 체력, 공격력, 방어력, 이름, 기술 리스트, 타입을 가지며, 클래스 외부에서 이들에게 접근하기 위한 get_으로 시작하는 메서드를 제공한다. 또한 해당 포스몬이 쓰러졌는지를 확인하기 위한 메서드와 공격력과 방어력을 초기화하는 메서드가 있다. (사실 포스몬의 타입은 변수에 할당되는 것이 아닌, 호출했을 때 타입이 적힌 문자열을 반환하는 메서드로 구성된다)
- 포스몬의 세부 능력치는 다음과 같다.

포스몬	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

- 상성이 좋은 경우 공격 데미지가 2배지만, 상성이 좋지 않더라도 가하는 데미지가 줄어들지는 않는다.
- 기술 클래스는 기술 클래스를 상속하는 공격 기술 클래스와 변화 기술 클래스로 나뉘며, 공격 기술 클래스는 Tackle, ScissorsCross, Swift, 변화 기술 클래스는 Growl, SwordDance, TailWhip이 있다.
- 기술 클래스의 인스턴스 변수에는 이름, 위력(공격 기술의 경우), 속도가 있으며, 기술을 사용하기 위한 use 메서드를 갖는다.
- 기술의 세부 사항은 다음과 같다.

분류	기술명	위력 / 효과	속도
공격 기술	Tackle	25	0
	ScissorsCross	30	0
	Swift	0	3
변화 기술	Growl	상대 공격력 5 감소	1
	SwordDance	자신 공격력 10 증가	0
	TailWhip	상대 방어력 5 감소	1

4. 프로그래밍 실행 방법 및 예시

● 모든 사진이 이어지는 장면은 아닙니다. (1~4 / 6 / 5, 7~)

- (1) 프로그램을 맨 처음 실행하면 나오는 화면으로, 0, 1, 2중 하나를 입력받는다. 그 이외의 입력은 잘못된 입력으로 처리하며 다시 입력받는다.
- (2) 메뉴에서 0을 선택하면 포스몬을 선택하는 화면으로 넘어간다. 0, 1, 2, 3중 하나를 입력해 포스몬을 선택하며, 그 이외의 입력은 잘못된 입력으로 처리한다. 포스몬을 선택하지 않았을 때 -1을 입력하면 잘못된 입력으로 처리한다.

- (3) 차례대로 0, 3, 1을 선택해 Ponix, Rocky, Normie를 선택한 장면이다.
- (4) 3마리 포스몬을 선택한 후 다시 Ponix를 고르려고 하자 더 이상 추가할 수 없다는 문구가 출력된다. 최소 1마리 이상 선택한 이후 -1을 입력하면 선택한 포스몬 목록이 출력되며 메뉴로 돌아온다. (3마리를 다 고르지 않은 상태에서 메뉴로 나간 후, 메뉴에서 0을 입력해 이어서 추가할 수 있다)
- (5) 메뉴에서 1을 선택하면 현재 선택한 포스몬 목록과 임의로 선택된 컴퓨터 포스몬 목록을 출력하며 배틀을 시작한다. 이때 포스몬 선택은 중복이 가능하며, 컴퓨터는 Swania 둘을 중복으로 선택한 모습을 볼 수 있다. 배틀 과정의 맨 처음에는 현재 상황을 담은 화면을 출력하며, 현재 포스몬이 사용할 수 있는 포스몬의 기술을 포함한다.
- (6) 명령은 "e", "o (num)", "s (num)"이 가능하다. 'e'를 입력했을 경우 현재 소유한 포스몬의 목록과 상태를 출력한다. 이때 장면은 선두였던 Ponix가 쓰러진 이후 Swania로 교체된 상태이다. 's' 명령을 통해 전투중인 포스몬을 교체할 수 있다. 이 상태에서 0, 이미 쓰러진 포스몬을 선택하거나, 1, 현재 전투 중인 포스몬으로 교체를 시도할 경우 불가능하다는 메시지를 출력하며, 현재 비전투 중이며 살아있는 Normie로 교체한 모습이다. 교체는 자신의 차례를 소모하므로 교대한 후 컴퓨터 포스몬의 기술이 사용된 모습이다.
- (7) (5)를 보면 현재 역상성이므로, 교체를 통해 상성이 없는 Normie로 교체한 모습이다.
- (8) 'o' 명령을 통해 기술을 사용할 수 있다. tailwhip의 속도는 1, swordDance의 속도는 0이므로 tailWhip이 먼저 사용되었다.
- (9) 공격 기술을 사용한 장면이다. Tackle과 ScissorsCross 모두 속도는 0이므로, 속도가 같을 때는 플레이어의 기술이 먼저 사용된다. 데미지 공식은 다음과 같다.
- $$\text{데미지} = \max(0, \text{위력} + \text{사용자의 공격력} - \text{피해자의 방어력}) * \text{배율}$$
- Normie는 무상성이므로 배율은 1이며, 컴퓨터의 Swania의 방어력은 5가 감소된 상태이므로 5, Normie의 공격력은 20, Tackle의 위력은 25이므로 $25+20-5 = 40$ 이다. 해당 사진에서 컴퓨터 포스몬이 공격을 맞아 체력이 80에서 40으로 감소한 모습을 볼 수 있다. 마찬가지로 Swania의 공격력은 SwordDance로 10 증가해 50이며, Normie의 방어력은 20, ScissorCross의 위력은 30, 배율은 1이므로 $30+50-20 = 60$, 사진이 맞음을 확인할 수 있다.
- 변화된 공격력과 방어력은 (8)번 사진을 확인하라.
- 한 번 더 공격하면 동일한 계산에 따라 상태 포스몬의 체력이 0이 되므로 컴퓨터의 포스몬이 교대되는 것을 확인할 수 있다. 또한 포스몬이 쓰러지면 현재 상황을 출력할 때 O가 X로 바뀌며 남은 수가 줄어든다.
- (10) Tackle을 남용하여 컴퓨터의 포스몬을 모두 쓰러뜨리면, 최종 화면을 출력하며 [배틀 결과] 문구와 함께 승리자를 소개하고 메뉴로 돌아간다.
- (11) 메뉴에서 3을 입력하면 프로그램이 종료된다.
- (12) 메뉴에서 포스몬을 선택하지 않은 상태로 1을 입력해 배틀하려고 하면 먼저 포스몬을 선택하라는 안내가 출력된다.

<p>(1)</p>  <pre>===== 0. 포스몬 선택 1. 배틀하기 2. 종료하기 ===== 입력: 4 잘못된 입력입니다. 다시 입력하세요. 입력: s 잘못된 입력입니다. 다시 입력하세요. 입력: █</pre>	<p>(2)</p> <pre>===== 당신이 사용할 포스몬을 선택하세요. 현재 0 마리/최대 3 마리 0. Ponix 1. Normie 2. Swania 3. Rocky ===== 입력: 4 잘못된 입력입니다. 다시 입력하세요. ===== 당신이 사용할 포스몬을 선택하세요. 현재 0 마리/최대 3 마리 0. Ponix 1. Normie 2. Swania 3. Rocky ===== 입력: -1 잘못된 입력입니다. 다시 입력하세요. ===== 당신이 사용할 포스몬을 선택하세요. 현재 0 마리/최대 3 마리 0. Ponix 1. Normie 2. Swania 3. Rocky ===== 입력: █</pre>
<p>(3)</p> <pre>===== 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: 1</pre>	<p>(4)</p> <pre>===== 당신이 사용할 포스몬을 선택하세요. 현재 3 마리/최대 3 마리 0. Ponix 1. Normie 2. Swania 3. Rocky -1. 그만두기 ===== 입력: 0 포스몬을 더 이상 추가할 수 없습니다. ===== 당신이 사용할 포스몬을 선택하세요. 현재 3 마리/최대 3 마리 0. Ponix 1. Normie 2. Swania 3. Rocky -1. 그만두기 ===== 입력: -1 ===== 당신의 포스몬 목록: Ponix Rocky Normie ===== </pre>

<p>(5)</p> <pre> o)Y o Y(T T Y Y l_j _/_ _/_ j_l_j_j _/_ l_j_j ===== 0. 포스몬 선택 1. 배틀하기 2. 종료하기 ===== 입력: 1 ===== 당신의 포스몬 목록: Ponix Swania Normie 컴퓨터 포스몬 목록: Swania Swania Normie ===== 배틀이 시작됩니다. ##### 컴퓨터 포스몬: [000] 3 / 3 Swania < Scissors 80 / 80 VS Ponix < Paper 86 / 86 당신의 포스몬: [000] 3 / 3 +++++ 기술: (0) Tackle (1) Growl (2) SwordDance ##### 입력: █ </pre>	<p>(6)</p> <pre> 입력: e ##### (0) Ponix < Paper -8 / 86 (1) Swania < Scissors 80 / 80 (2) Normie < Nothing 80 / 80 입력: s 0 ##### 이미 쓰러진 포스몬입니다. 다시 입력하세요. 입력: s 1 ##### 자기자신과는 교대할 수 없습니다. 다시 입력하세요. 입력: s 2 ##### 당신의 포스몬 Normie로 교대 컴퓨터 Swania: SwordDance 기술 사용 - 컴퓨터 포스몬의 [공격력] 10 증가 (60 -> 70) ##### 컴퓨터 포스몬: [000] 3 / 3 Swania < Scissors 80 / 80 VS Normie < Nothing 80 / 80 당신의 포스몬: [x00] 2 / 3 +++++ 기술: (0) Tackle (1) Swift (2) TailWhip ##### 입력: █ </pre>
<p>(7)</p> <pre> 입력: s 2 ##### 당신의 포스몬 Normie로 교대 컴퓨터 Swania: SwordDance 기술 사용 - 컴퓨터 포스몬의 [공격력] 10 증가 (30 -> 40) ##### 컴퓨터 포스몬: [000] 3 / 3 Swania < Scissors 80 / 80 VS Normie < Nothing 80 / 80 당신의 포스몬: [000] 3 / 3 +++++ 기술: (0) Tackle (1) Swift (2) TailWhip ##### 입력: █ </pre>	<p>(8)</p> <pre> ##### 컴퓨터 포스몬: [000] 3 / 3 Swania < Scissors 80 / 80 VS Normie < Nothing 80 / 80 당신의 포스몬: [000] 3 / 3 +++++ 기술: (0) Tackle (1) Swift (2) TailWhip ##### 입력: o 2 ##### 당신의 Normie: TailWhip 기술 사용 - 컴퓨터 포스몬의 [방어력] 5 감소 (10 -> 5) 컴퓨터 Swania: SwordDance 기술 사용 - 컴퓨터 포스몬의 [공격력] 10 증가 (40 -> 50) </pre>


```

입력: 0 0
#####
당신의 Normie: Tackle 기술 사용
- 컴퓨터 포스몬의 [체력] 40 감소 (80 -> 40)
컴퓨터 Swania: ScissorsCross 기술 사용
- 당신 포스몬의 [체력] 60 감소 (80 -> 20)

#####
컴퓨터 포스몬: [000] 3 / 3
Swania <|Scissors 40 / 80|
VS
Normie <|Nothing 20 / 80|
당신의 포스몬: [000] 3 / 3
+++++
기술: (0) Tackle (1) Swift (2) TailWhip
#####
입력: 0 0
#####
당신의 Normie: Tackle 기술 사용
- 컴퓨터 포스몬의 [체력] 40 감소 (40 -> 0)
컴퓨터 Swania: 쓰러짐
컴퓨터 포스몬 Swania 로 교대

#####
컴퓨터 포스몬: [X00] 2 / 3
Swania <|Scissors 80 / 80|
VS
Normie <|Nothing 20 / 80|
당신의 포스몬: [000] 3 / 3
+++++
기술: (0) Tackle (1) Swift (2) TailWhip
#####
입력: 

```

```
| o )Y Y( \ / | T T / \ | Y Y | _ Y  
| / | O \ / | T \ | | | O | | |  
| | | ! \ | | | | | | | | |  
l j \ / j l j j \ / l j j
```

0. 포스몬 선택
1. 배틀하기
2. 종료하기

입력: 2
Program ended. Bye!

```
#####
컴퓨터 포스몬: [XXO] 1 / 3
Normie <|Nothing 5 / 80|
VS
Normie <|Nothing 10 / 80|
당신의 포스몬: [OOO] 3 / 3
+++++
기술: (0) Tackle (1) Swift (2) TailWhip
#####
입력: o 0
#####
당신의 Normie: Tackle 기술 사용
- 컴퓨터 포스몬의 [체력] 25 감소 (5 -> -20)
컴퓨터 Normie: 쓰러짐
#####
컴퓨터 포스몬: [XXX] 0 / 3
Normie <|Nothing -20 / 80|
VS
Normie <|Nothing 10 / 80|
당신의 포스몬: [OOO] 3 / 3
+++++
기술: (0) Tackle (1) Swift (2) TailWhip
#####
[배틀 결과] 당신이 이겼습니다!
```

```
=====
o)Y( Y( T T Y Y Y
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
l j l j l j l j l j l j
=====
```

0. 포스몬 선택
1. 배틀하기
2. 종료하기

```
=====
```

입력 : 1

싸울 포스몬이 없습니다! 먼저 포스몬을 선택해 주세요.

5. 토론

- 문제 설명 문서에서 제공된 클래스 설명에는 없지만, 상속 및 연관 관계도에 있는 `is_knocked_out()` 메서드를 추가하였다. 이 메서드는 해당 포스몬이 살아 있는지를 boolean 타입으로 반환한다. 이 이외에도 각 인스턴스 변수를 반환하는 `get_health()`와 `get_moves()` 메서드를 만들어 사용하였다.
- `PhysicalMove` 클래스에서 제시된 설명에 따라 `get_power()` 메서드를 작성하였다. 그러나 실제로 해당 메서드는 사용하지 않았다. `power` 변수는 해당 클래스에서만 사용하기에, 클래스 내에서는 `get_power()` 메서드를 호출하지 않고 변수에 직접 접근하였다.
- 체력은 음수가 될 수 있지만, 방어력과 공격력은 음수가 불가능하다. 따라서 방어력이나 공격력을 5 감소하는 기술을 사용할 때, 3 -> -2 와 같이 출력하면 안 되며 3 -> 0처럼 출력되어야 한다. 즉, `posmon.defense -> posmon.defense + self.amount`와 같이 코드를 작성하지 말고, `posmon.defense + self.amount`가 0 미만인지 확인하고 다르게 출력해야만 한다.

6. 결론

- 본 과제에서는 객체지향프로그래밍(OOP)의 핵심인 클래스에 대해 다루고, 클래스의 핵심 개념인 생성자와 상속, 오버라이딩을 학습하였다. 같은 역할의 코드를 줄이고 객체를 구분하여 관리하기 위해 대주제로 상위 클래스를 만든 다음 상위 클래스를 상속하면서 세세한 구분으로 나뉘는 하위 클래스를 작성하였다. 클래스를 이용한 객체를 통해 데이터를 다룸으로써 각 객체의 데이터와 연산을 쉽게 수정하고 사용할 수 있었다.
- 어느 기술이나 포스몬을 추가하고 싶을 때, 클래스와 간단한 변수만 수정함으로써 쉽게 추가할 수 있도록 하는 것이 좋다. 예를 들어 단순히 `Ponix`인 경우, `Normie`인 경우 각각을 출력하기 보다, 전체 포스몬의 이름 인덱스를 담아두는 변수를 만들고 반복문을 이용해 변수에 대한 정보를 출력하는 등의 방법을 사용할 수 있다.

7. 개선 방향

- 메뉴 0번에서 포스몬을 선택한 후, 선택한 포스몬을 배틀에 사용하기 위해 동적으로 인스턴스를 생성하기 위해 `globals()` 함수를 사용하였다. 이 경우 포스몬의 이름이 `int`나 `Exception`같이 예약어가 되지 않도록 주의하며 사용해야 하며, 이를 방지하고 안정성을 늘리는 방법을 추후 찾아볼 것이다.
- `battle()` 함수에서, 플레이어와 컴퓨터 둘의 기술을 가져와 사용할 때 서로 비교해야 할 상황이 매우 복잡하다. 각각의 속도를 비교하고, 기술을 사용한 뒤 포스몬이 쓰러졌는지 확

인해야 하며, 쓰러졌을 경우 남은 포켓몬이 있는지 확인해야 한다. 이때 공격자가 플레이어인지 컴퓨터인지 나누어 생각해서 경우를 나눠야 하므로 코드가 반복되면서 독성이 매우 떨어지는 문제가 있었다. 공격자와 그 대상을 받아 범용적으로 판단을 내리는 함수를 새롭게 정의하여 사용하면 더 깔끔하게 코드를 만들 수 있을 것이다.

- 결론 부분에서 추후 확장성을 고려하면 좋다고 서술했지만, 해당 코드에서는 현재 작동하긴 하지만 추후 확장하기 위해서 여러 변수를 변경해야 하는 번거로움이 있다. 주로 예외 처리할 때 가능한 경우를 깔끔하게 작성하지 않았기 때문이며, 이를 수정한다면 더욱 좋은 코드가 될 수 있을 것이다.