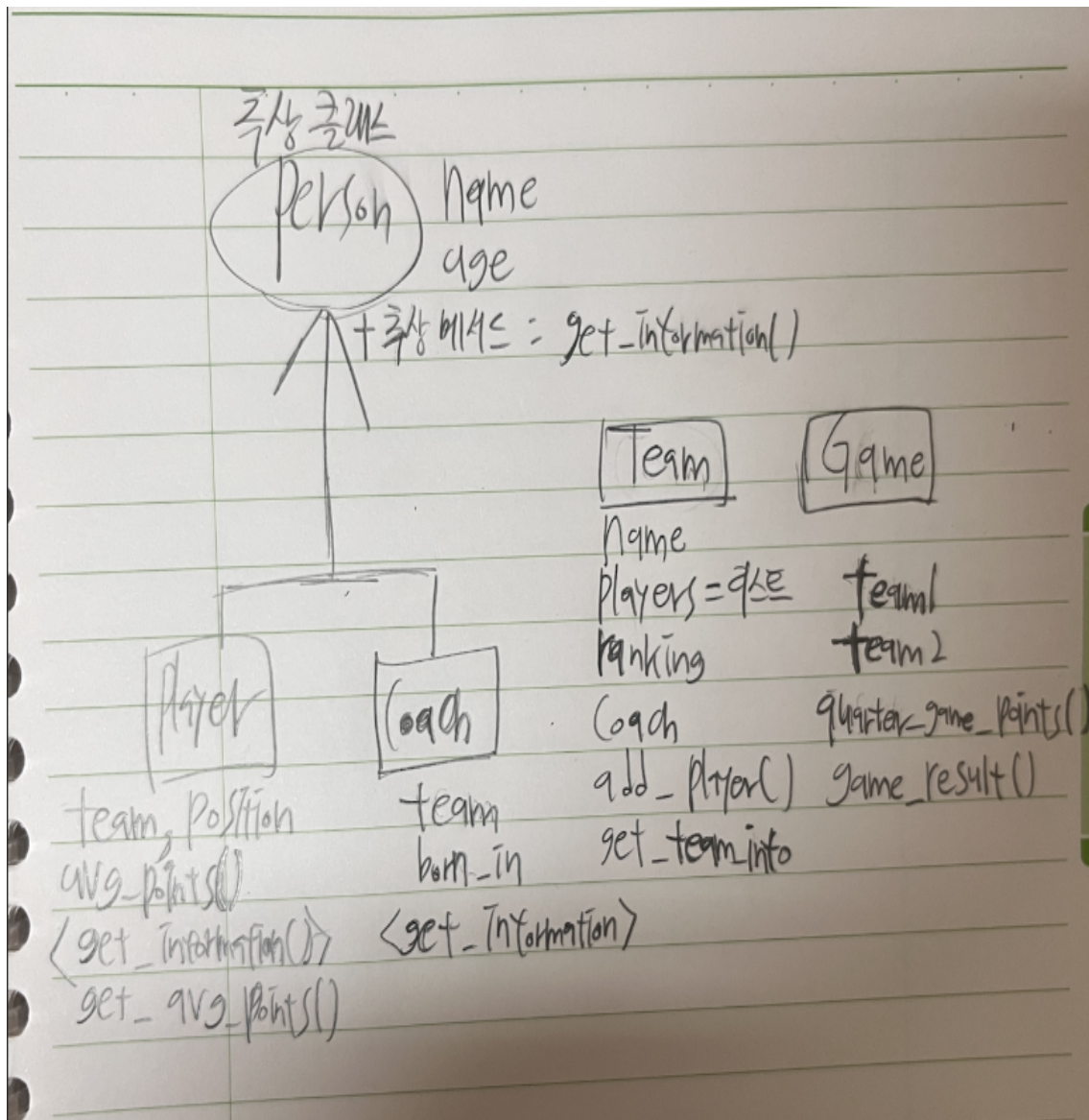


# NBA 도메인

20220311 김준영

1. NBA 도메인 설계입니다

글씨가 매우 나쁜 점에 대해 양해부탁드립니다.



```

from abc import ABC, abstractmethod

class Person(ABC):  2개의 사용 위치

    def __init__(self, name, age):
        self.name = name
        self.age = age

    @abstractmethod  2개의 사용 위치(2개의 동적)
    def get_information(self):
        pass

```

abc 라이브러리를 가져오고 Person 클래스를 **추상클래스**로 지정했습니다.  
 사람은 모두 이름과 나이를 갖는다는 공통적인 속성을 활용했습니다.  
 그리고 Person 클래스를 상속받는 하위 클래스에 쓰기 위한  
 get\_information()이라는 **추상메서드**를 만들었습니다

```

class Player(Person):  10개의 사용 위치

    def __init__(self, name, age, team, position):
        super().__init__(name, age)
        self.team = team
        self.position = position
        self.avg_score = 0
        self.assist_points = 0
        self.avg_rebound = 0

    def avg_points(self, avg_score, assist_points, avg_rebound):  10개의 사용 위치
        self.avg_score += avg_score
        self.assist_points += assist_points
        self.avg_rebound += avg_rebound

    def get_information(self):  2개의 사용 위치(2개의 동적)
        return f"이름:{self.name} 팀:{self.team} 포지션:{self.position}"

    def get_avg_points(self):  1개의 사용 위치(1개의 동적)
        return f"평균 득점:{self.avg_score} 평균 어시:{self.assist_points} 평균 리바:{self.avg_rebound}"

```

Person 클래스를 상속받은 Player 클래스를 만들었고 super.\_\_init\_\_를 써서  
 부모 클래스의 생성자를 호출했습니다.  
 그리고 avg\_points() 메서드를 통해 **더미 데이터**를 누적할 수 있도록 구현하  
 였습니다.  
 추상 메서드인 get\_information()을 **오버라이딩**하여 농구 선수의 개별 정보를  
 출력할 수 있게 구현하였습니다.

```
class Coach(Person): 2개의 사용 위치

    def __init__(self, name, age, team, born_in):
        super().__init__(name, age)
        self.team = team
        self.born_in = born_in

    def get_information(self): 2개의 사용 위치(2개의 동적)
        return f"이름:{self.name} 나이:{self.age} 팀:{self.team} 출신:{self.born_in}"
```

Person을 상속받은 Coach 클래스를 구현했습니다.

위에 Player 클래스와 마찬가지로 super를 써서 부모 클래스의 생성자 호출, 그 후 get\_information() 메서드를 코치에 대한 정보가 반환될 수 있도록 오버라이딩해줬습니다.

```
class Team: 2개의 사용 위치

    def __init__(self, name, ranking, coach):
        self.name = name
        self.players = []
        self.ranking = ranking
        self.coach = coach

    def add_player(self, player): 2개의 사용 위치
        self.players.append(player)

    def get_team_info(self): 2개의 사용 위치
        info = f"팀명:{self.name}, 순위:{self.ranking}\n"
        if self.coach:
            info += f"<코치> {self.coach.get_information()}\n"
        if self.players:
            info += "선수 목록:\n"
            for p in self.players:
                info += f"{p.get_information()} {p.get_avg_points()}\n"
        info += f"팀명:{self.name}, 순위:{self.ranking}\n"
        return info
```

팀의 이름/순위/코치를 속성으로 갖으며 농구 선수들의 목록을 저장할 players라는 리스트를 만들었습니다.

add\_player() 함수로 팀에 선수를 추가 할 수 있도록 구현했습니다.

get\_team\_info() 함수로 팀 이름, 순위, 코치, 선수 정보를 모두 출력할 수 있도록 구현했습니다.

```
class Game: 1개의 사용 위치
```

```
def __init__(self, team1, team2):  
    self.team1 = team1  
    self.team2 = team2  
    self.team1.scores = []  
    self.team2.scores = []
```

```
def quarter_game_points(self, results): 1개의 사용 위치  
    for score1, score2 in results:  
        self.team1.scores.append(score1)  
        self.team2.scores.append(score2)
```

```
def game_result(self): 1개의 사용 위치  
    total_scores1 = 0  
    total_scores2 = 0  
    scores1 = self.team1.scores  
    scores2 = self.team2.scores  
  
    for i in range(len(scores1)):  
        if scores1[i] > scores2[i]:  
            print(f"{i+1}쿼터는 {self.team1.name}의 승리입니다.")  
        else:  
            print(f"{i+1}쿼터는 {self.team2.name}의 승리입니다.")  
  
    for j in range(len(scores1)):  
        total_scores1 += scores1[j]  
        total_scores2 += scores2[j]  
    if total_scores1 > total_scores2:  
        print(f"이번 경기는 {self.team1.name}의 승리입니다.")  
    else:  
        print(f"이번 경기는 {self.team2.name}의 승리입니다.")
```

두 팀을 받아서 경기를 시뮬레이션하는 Game 클래스를 만들었습니다.  
quarter\_game\_points() 함수를 통해 쿼터별 점수를 입력받습니다.  
game\_result()를 통해 총 4쿼터동안 각 쿼터에서 승리한 팀과 최종 승리한 팀을 출력할 수 있도록 구현했습니다.

```

lebron = Player( name: "LeBron James", age: 40, team: "Lakers", position: "PF")
lebron.avg_points( avg_score: 24.4, assist_points: 8.2, avg_rebound: 7.8)
luka = Player( name: "Luka Dončić", age: 25, team: "Lakers", position: "PG")
luka.avg_points( avg_score: 28.2, assist_points: 7.7, avg_rebound: 8.2)
reaves = Player( name: "Austin Reaves", age: 26, team: "Lakers", position: "SG")
reaves.avg_points( avg_score: 20.2, assist_points: 5.8, avg_rebound: 4.5)
rui = Player( name: "Rui Hachimura", age: 26, team: "Lakers", position: "SF")
rui.avg_points( avg_score: 13.1, assist_points: 1.4, avg_rebound: 5.0)
hayes = Player( name: "Jaxson Hayes", age: 24, team: "Lakers", position: "C")
hayes.avg_points( avg_score: 6.8, assist_points: 0.8, avg_rebound: 4.8)

curry = Player( name: "Stephen Curry", age: 36, team: "Warriors", position: "PG")
curry.avg_points( avg_score: 26.4, assist_points: 6.1, avg_rebound: 4.9)
klay = Player( name: "Klay Thompson", age: 35, team: "Warriors", position: "SG")
klay.avg_points( avg_score: 17.9, assist_points: 2.2, avg_rebound: 3.3)
draymond = Player( name: "Draymond Green", age: 35, team: "Warriors", position: "PF")
draymond.avg_points( avg_score: 8.7, assist_points: 6.9, avg_rebound: 7.1)
wiggins = Player( name: "Andrew Wiggins", age: 30, team: "Warriors", position: "SF")
wiggins.avg_points( avg_score: 12.9, assist_points: 1.7, avg_rebound: 4.2)
looney = Player( name: "Kevon Looney", age: 29, team: "Warriors", position: "C")
looney.avg_points( avg_score: 4.5, assist_points: 1.7, avg_rebound: 6.0)

coach_lakers = Coach( name: "Darvin Ham", age: 50, team: "Lakers", born_in: "USA")
coach_warriors = Coach( name: "Steve Kerr", age: 58, team: "Warriors", born_in: "USA")
lakers = Team( name: "Lakers", ranking: 7, coach_lakers)
warriors = Team( name: "Warriors", ranking: 10, coach_warriors)

```

현재 프로그램에서 실제 DB 연동이 아닌 웹 서칭을 통해 현재 NBA 선수들의 이름과 평균 기록/코치/팀에 대한 정보로 더미 데이터를 생성하였습니다.

출력결과입니다.

팀명:Lakers, 순위:7

<코치> 이름:Darvin Ham 나이:50 팀:Lakers 출신:USA

선수 목록:

이름:LeBron James 팀:Lakers 포지션:PF 평균 득점:24.4 평균 어시:8.2 평균 리바:7.8

이름:Luka Dončić 팀:Lakers 포지션:PG 평균 득점:28.2 평균 어시:7.7 평균 리바:8.2

이름:Austin Reaves 팀:Lakers 포지션:SG 평균 득점:20.2 평균 어시:5.8 평균 리바:4.5

이름:Rui Hachimura 팀:Lakers 포지션:SF 평균 득점:13.1 평균 어시:1.4 평균 리바:5.0

이름:Jaxson Hayes 팀:Lakers 포지션:C 평균 득점:6.8 평균 어시:0.8 평균 리바:4.8

팀명:Lakers, 순위:7

팀명:Warriors, 순위:10

<코치> 이름:Steve Kerr 나이:58 팀:Warriors 출신:USA

선수 목록:

이름:Stephen Curry 팀:Warriors 포지션:PG 평균 득점:26.4 평균 어시:6.1 평균 리바:4.9

이름:Klay Thompson 팀:Warriors 포지션:SG 평균 득점:17.9 평균 어시:2.2 평균 리바:3.3

이름:Draymond Green 팀:Warriors 포지션:PF 평균 득점:8.7 평균 어시:6.9 평균 리바:7.1

이름:Andrew Wiggins 팀:Warriors 포지션:SF 평균 득점:12.9 평균 어시:1.7 평균 리바:4.2

이름:Kevon Looney 팀:Warriors 포지션:C 평균 득점:4.5 평균 어시:1.7 평균 리바:6.0

팀명:Warriors, 순위:10

1쿼터는 Warriors의 승리입니다.

2쿼터는 Lakers의 승리입니다.

3쿼터는 Lakers의 승리입니다.

4쿼터는 Warriors의 승리입니다.

이번 경기는 Lakers의 승리입니다.