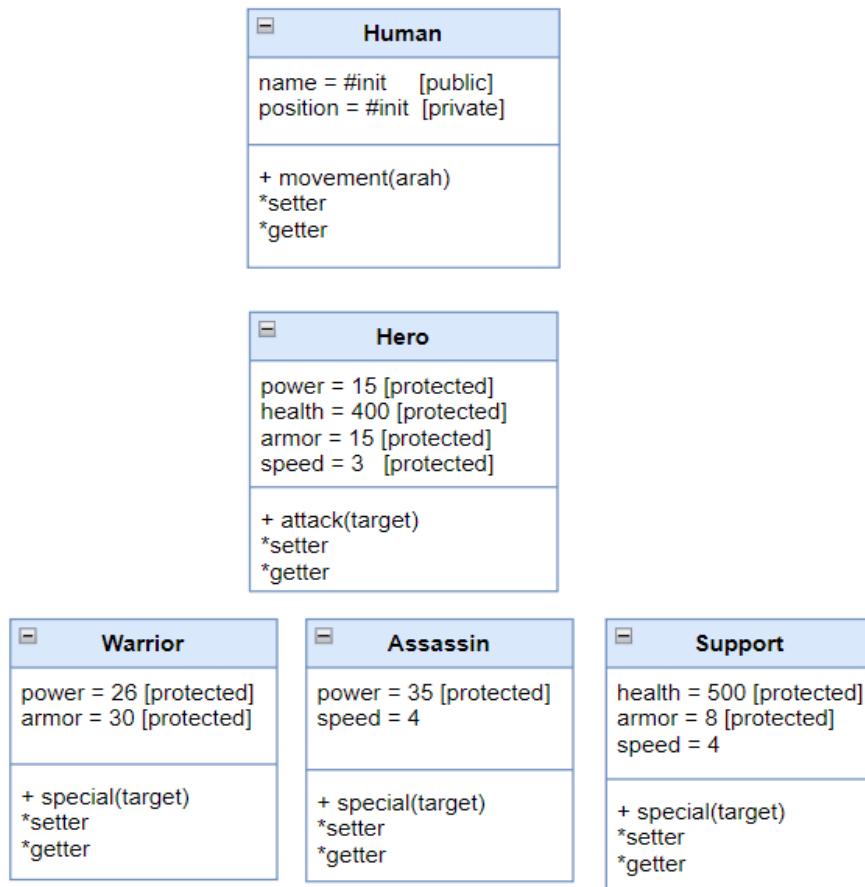


1. Buatlah class Human, Hero, Warrior, Assassin, Support berdasarkan aturan berikut. (Nb: Tambahan, pada class Human terdapat tambahan atribut **speed = #init [protected]**)



Method :

- Method attack() mengurangi health target sebanyak nilai power
target.health = target.health - self.power (menggunakan metod setter)
- Method movement() menerima string "L" kiri (mengurangi nilai pos_x sebanyak speed) dan "R" kanan (menambah nilai pos_x sebanyak speed)

Tabel untuk method special() pada masing masing class

Special	
Warrior	mengubah nilai armor menjadi 45 dan power menjadi 32 lalu mengurangi health target sebanyak nilai power
Assassin	mengubah nilai speed menjadi 7 dan power menjadi 42 lalu mengurangi health target sebanyak nilai power
Support	mengubah nilai speed menjadi 6 dan menambah health target sebanyak 150

Catatan :

- Nama dan posisi setiap objek diisi pada saat inisialisasi objek atau menggunakan `__init__()`. Gunakan `super().__init__(parameter)` agar constructor class parent tidak tertimpa/override
- Tambahkan setter/getter jika ingin mengakses variabel private atau protected dari luar class. Cukup definisikan di class parent saja

Contoh program menggunakan class Warrior, Assassin, dan Support:

```
from hero import Warrior, Assassin, Support

warrior = Warrior("bambang", pos_x=10)
assassin = Assassin("joko", pos_x=25)
support = Support("udin", pos_x=30)

# sebelum
print("health (before)", warrior.getHealth())
assassin.attack(warrior)
# sesudah
print("health (after)", warrior.getHealth())

print("-"*10)

# sebelum
print("Warrior (health)", warrior.getHealth())
print("Support (speed) : ", support.getSpeed())

support.special(warrior)

# sesudah
print("Warrior (health)", warrior.getHealth())
print("Support (speed): ", support.getSpeed())
```

Output :

```
health (before) 400
health (after) 365
-----
Warrior (health) 365
Support (speed) : 4
Warrior (health) 515
Support (speed): 6
```

2. Buatlah sebuah class Bernama Mahasiswa yang memiliki atribut sebagai berikut:

- Nama
- Nim
- Jurusan
- Ipk

Class Mahasiswa juga harus memiliki method sebagai berikut :

- Tampilkan_info() : Menampilkan informasi lengkap mahasiswa, termasuk nama, nim, jurusan, dan ipk.
- Hitung_predikat() : Menghitung predikat berdasarkan IPK mahasiswa. Jika IPK ≥ 3.5 , predikatnya adalah "Cumlaude". Jika IPK ≥ 3.0 , predikatnya adalah "Sangat Memuaskan". Jika IPK ≥ 2.5 , predikatnya adalah "Memuaskan". Jika IPK ≥ 2.0 , predikatnya "Cukup". Jika IPK < 2.0 , predikatnya adalah "Kurang"