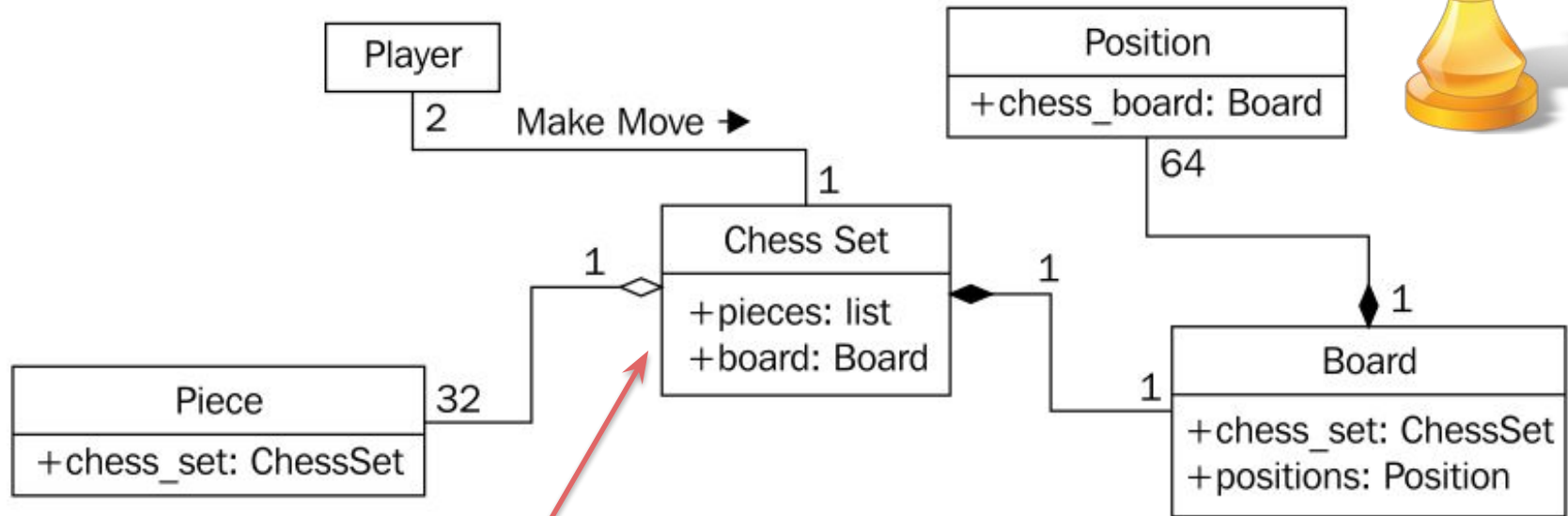


Herencia

Repaso - Jueves 30 de abril 2020

Composición *versus* agregación



Agregación ocurre cuando el objeto contenido en otro puede tener sentido como objeto único (rombo vacío).

```
class ChessSet:
```

```
    def __init__(self, piezas):
        self.piezas = piezas
        self.tablero = Board()
```

```
piezas = list((Pieza("Torre"), Pieza("Alfil"), ...))
juego = ChessSet(piezas)
```

Al hacer **composición**, se define que un objeto está compuesto por otro que “existe” al mismo tiempo que él.

Al hacer **agregación** se *agregan* objetos ya existentes a otro objeto.

Herencia

Vehículo

Ruedas
Tamaño
Color
Encender
Mover

```
class Vehículo:  
    def __init__(self):  
        self.ruedas = ...  
        ...  
  
    def encender(self):  
        print("Encendiendo vehículo")
```



Marca
Modelo
Airbag
Abrir maleta



Nº estudiantes
Filas asientos
Lista Colegios



Cilindros
Tipo Llantas
Encender



Armamento
Radar
Mover

```
class Moto(Vehículo):  
    def __init__(self):  
        self.cilindros = ...  
        ...
```

```
    def encender(self):  
        print("Encendiendo moto")
```

Overriding

Otros ejercicios

¡Si tienen dudas resolviendolos, no duden en hacer una *issue* con su duda!

- Ejercicios propuestos sobre *herencia*. ([Contenidos: semana 6](#))
- [Actividad 2 en 2017-1](#) (Actividad sobre OOP, con herencia)
- [Actividad 2 en 2017-2](#) (Actividad sobre OOP, con herencia)
- [Actividad 1 en 2018-1](#) (Actividad sobre OOP, con herencia)
- [Actividad 5 en 2018-2](#) (Actividad sobre OOP, con herencia)
- [Actividad 2 en 2019-1](#) (Actividad sobre OOP, con herencia)
- [Actividad 1 en 2019-2](#) (Actividad sobre OOP, con herencia)