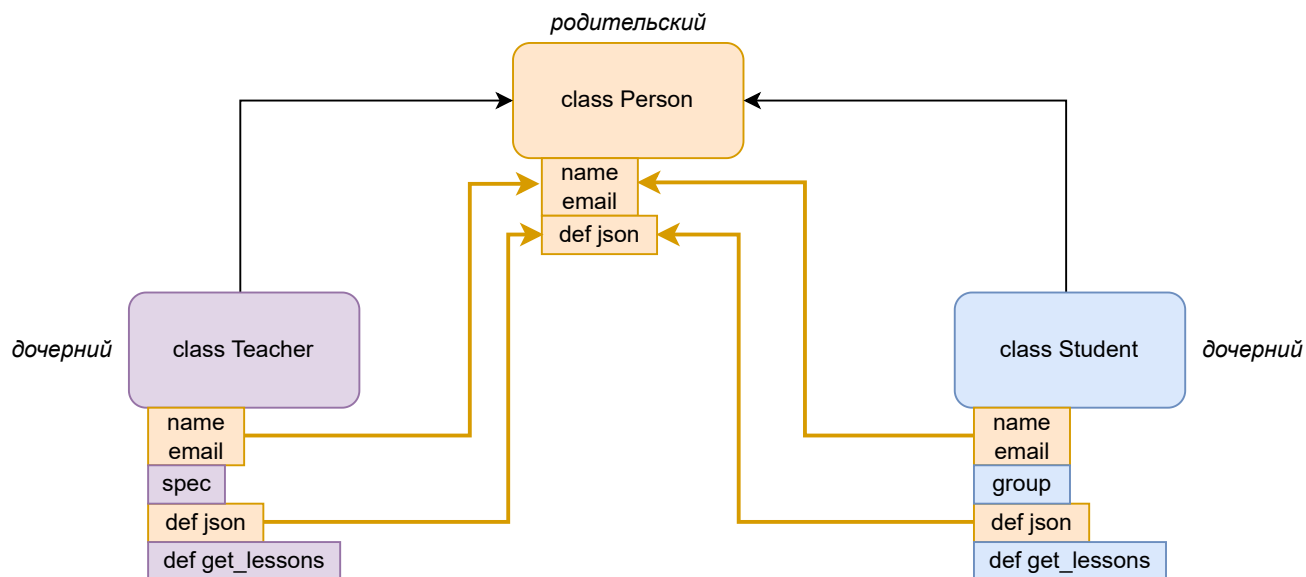


# НАСЛЕДОВАНИЕ

Наследование — это способ создания нового класса, используя атрибуты и методы уже существующего класса без внесения в него изменений.

Создаваемый класс является дочерним (или «производным») классом, а существующий класс является родительским (или «базовым») классом.



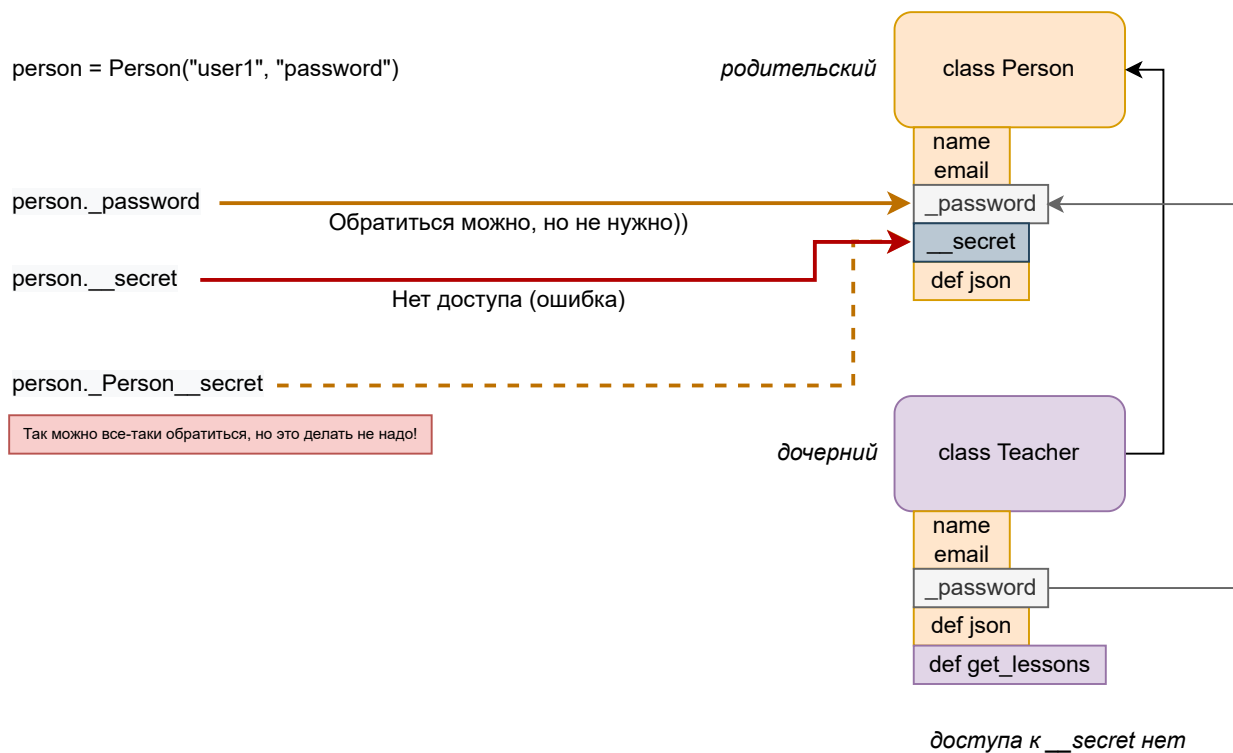
# ИНКАПСУЛЯЦИЯ

Инкапсуляция — это сокрытие атрибутов и методов внутри класса, которое предотвращает изменение и доступ к атрибутам и методам данного класса извне и разрешает использовать их только внутри класса.

В Python такие атрибуты и методы обозначаются с помощью одинарного `_` или двойного `__` подчеркиваний в качестве префиксов.

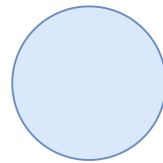
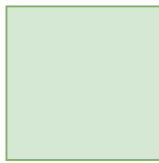
```
# Приватный атрибут
# К нему необходимо обращаться только внутри данного или унаследованного класса
self._password = password
```

```
# Защищенный атрибут
# К нему можно обращаться только внутри данного класса
self.__id = random.randint(1000, 9999)
```



# ПОЛИМОРФИЗМ

Полиморфизм – это возможность работы с совершенно разными объектами единым образом.



```
class Square:
    def __init__(self, x: int | float):
        self.x = x

    def area(self) -> int | float:
        return self.x**2
```

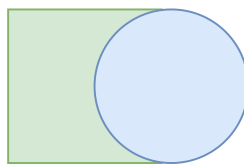
```
class Circle:
    PI = 3.14 # Глобальный атрибут

    def __init__(self, radius: int | float):
        self.radius = radius

    def area(self) -> int | float:
        return self.PI * self.radius**2
```

```
s1 = Square(10)
```

```
c1 = Circle(1)
```



```
.area()
```

У обоих есть метод area