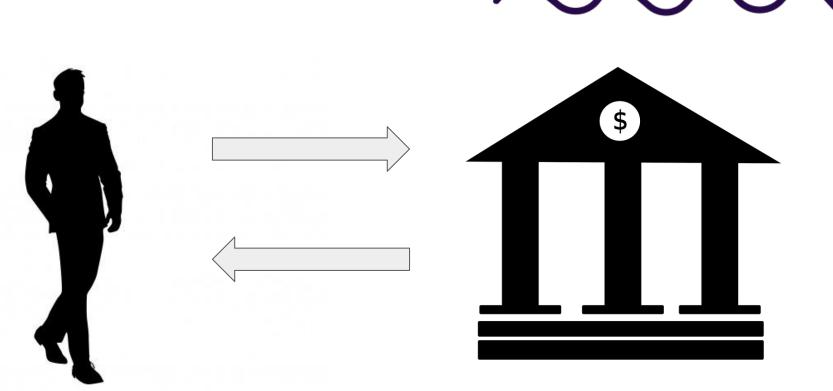


# Pystart.pl

Enkapsulacja

lekcja trzydziesta dziewiąta

O co chodzi?



#### Czego używamy zamiast?







O co chodzi?





**Enkapsulacja** - ukrywanie pól danej klasy dla innych klas. W ten sposób chronimy siebie i innych programistów przed **nieprzewidzianym przez nas** modyfikowaniem właściwości lub wywoływaniem metod.

#### **Enkapsulacja w Pythonie**

```
class Bank:
def __init__(self):
     self._money = 1000 * 1000
def withdraw(self, amount: int):
     if amount > self._money:
         return 0
     self._money -= amount
     return amount
 def deposit(self, amount: int):
     self._money += amount
```



- → Symbol podkreślenia przed nazwą zmiennej lub metody.
- → Pycharm nie podpowiada takich zmiennych.
- → Nie chodzi tu o bezpieczeństwo, ale o wygodę

#### **Enkapsulacja w Pythonie**



→ Metody także mogą być "prywatne"

→ Będzie można wywoływać je wewnątrz klasy poprzez self.\_convert\_to\_pln, a z "zewnątrz" już nie.

**Enkapsulacja w Pythonie** 





A co jeśli spróbujemy się do tego odwołać mimo tego?

#### PyStart #39 Enkapsulacja Enkapsulacja w Pythonie





A co jeśli spróbujemy się do tego odwołać mimo tego?

Zadziała zwyczajnie, ale sprawdźmy to.

# PyStart #39 Enkapsulacja Zadania dla nabrania wprawy



Przygotuj program, który będzie przechowywał listę zakupów. Każdy wpis jest osobnym obiektem klasy ListItem. Jeśli dany produkt znajduje się już na liście nie powinien być dodany drugi raz, zamiast tego powinna być zwiększana jego ilość. Produkty do zakupienia przechowuj w zmiennej prywatnej.

Obiekt klasy List musi posiadać następujące metody:

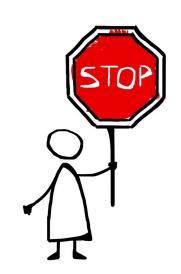
addItem(product: Product, quantity: float)

removeItem(product:Product, quantity: float)

listItems(): list

calculateTotalCost()

**39.1** 



Każdy obiekt klasy Product posiada cenę oraz nazwę