

## Lista 12

### Zadanie 1 (15 pkt)

- (2 pkt) W pliku `trasa.py` stwórz klasę `Trasa`, która:
  - posiada *prywatne* zmienne przechowujące kolejne odcinki drogi (przebyty dystans oraz czas)
  - posiada metodę, która przyjmuje dwa argumenty (dystans i czas) i aktualizuje trasę
  - posiada zaimplementowaną metodę `__iter__` tak, aby umożliwić iterację po odcinkach, np.

```
trasa = Trasa()
trasa.add(10, 5)
trasa.add(20, 3)
```

```
for s, t in trasa:
    print(f"Dystans = {s}, czas = {t}")
```

wydrukuję:

```
Dystans = 10, czas = 5
Dystans = 20, czas = 3
```

- (3 pkt) W pliku `pojazd.py` stwórz klasę `Pojazd`, która:
  - inicjalizowana jest kolejnym numerem *id* (wykorzystaj atrybut klasowy do zliczania istniejących już pojazdów)
  - posiada *prywatną* zmienną przechowującą prędkość (inicjalizowana w `__init__` z domyślną wartością 0)
  - posiada *prywatną* zmienną przechowującą przyspieszenie (inicjalizowana w `__init__`)
  - posiada *prywatną* zmienną przechowującą trasę (`Trasa`)
  - posiada metodę `jedz`, która przyjmuje jeden argument (czas), liczy przebytą drogę (zakładając ruch jednostajny) oraz aktualizuje trasę
  - posiada metodę `przyspieszaj`, która przyjmuje jeden argument (czas), liczy przebytą drogę (zakładając ruch jednostajnie przyspieszony) oraz aktualizuje trasę i prędkość
  - posiada metodę, która drukuje informacje o przebytej trasie - kolejne odcinki (dystans i czas) oraz całkowita droga i całkowity czas
- (3 pkt) W pliku `samochod.py` stwórz klasę `Samochod`, która dziedziczy z klasy `Pojazd` oraz:
  - posiada *prywatną* zmienną przechowującą ilość pozostałego paliwa (inicjalizowana w `__init__`)
  - posiada *prywatną* zmienną przechowującą informacje o spalaniu (inicjalizowana w `__init__`) - *należy przyjąć stałe spalanie*
  - posiada metodę, która aktualizuje stan paliwa na podstawie przebytej drogi

- posiada zmodyfikowane funkcje `jedz` / `przyspieszaj`, które wywołują powyższą metodę oraz metodę `jedz` / `przyspieszaj` z klasy bazowej
- posiada odpowiednie zabezpieczenie, które uniemożliwia wywołanie funkcji `jedz` / `przyspieszaj`, jeśli brakuje paliwa na pokonanie odpowiedniego dystansu (*wsk. metody `jedz` / `przyspieszaj` w klasie `Pojazd` mogą oprócz aktualizowania trasy zwracać obliczoną drogę*)
- (2 pkt) W pliku `hulajnoga.py` stwórz klasę `Hulajnoga`, która dziedziczy z klasy `Pojazd` oraz:
  - posiada *prywatną* zmienną przechowującą maksymalną prędkość (inicjalizowana w `__init__`)
  - uniemożliwia dalsze przyspieszanie, jeśli maksymalna prędkość została osiągnięta
- (3 pkt) Napisz program do *jazdy wybranym pojazdem*, który:
  - na początku pozwala użytkownikowi wybrać pojazd: samochód albo hulajnoga
  - w zależności od wyboru prosi użytkownika o podanie odpowiednich danych potrzebnych do inicjalizacji obiektu
  - następnie (aż do wyboru opcji ‘Zakończ’) daje użytkownikowi poniższe opcje:
    - \* ‘Jedź’
    - \* ‘Przyspiesz’
    - \* ‘Drukuj trasę’
    - \* ‘Zakończ’
  - dwie pierwsze dodatkowo pytają o czas
  - w przypadku samochodu program powinien blokować opcje ‘Jedź’ i ‘Przyspiesz’, gdy skończy się paliwo
  - w przypadku hulajnogi program powinien blokować opcję ‘Przyspiesz’, jeśli maksymalna prędkość zostanie osiągnięta
  - opcja `Drukuj` powinna wywoływać metodę z ostatniego podpunktu do klasy `Pojazd`
- (2 pkt) Uzupełnij powyższy program o możliwość rysowania wykresu zależności drogi od czasu - w formie punktów, końców odcinków (przebyta droga vs przebyty czas na końcach kolejnych odcinków). *Wsk. `matplotlib`, `scatter`.*