

```

class Vozilo(ABC):
    def __init__(self, marka, model, godina) -> None:
        self.marka = marka
        self.model = model
        self.godina = godina
        self.spremnik = 0

    @abstractmethod
    def trubi(self):
        pass

    def vozi(self, n_km):
        potroseno_l = self.potrosnja * (n_km / 100)
        self.spremnik = 0 if self.spremnik <= potroseno_l else self.spremnik - potroseno_l

    def natoci(self, l_goriva):
        self.spremnik += l_goriva

    def __str__(self) -> str:
        return self.marka + ", " + self.model + ", " + str(self.godina)

    def srednja_vrijednost_godina(vozila):
        if not vozila:
            return 0
        ukupno_godina = sum(vozilo.godina for vozilo in vozila)
        return ukupno_godina / len(vozila)

    def rangiraj_po_godinama(vozila):
        return sorted(vozila, key=lambda x: x.godina)

```

Nadklasa je klasa "Vozilo"; to je apstraktna klasa što znači da ne možemo instancirati objekt te klase.

Time dajemo semantiku da vozilo mora biti neko konkretno, kao auto, kamion, bus, itd.

3 glavna atributa koja zadajemo pri inicijalizaciji su marka, model i godina proizvodnje vozila.

Dodatni atribut je "spremnik" koji označava trenutnu količinu goriva u litrama. Na početku je ta količina 0, no spremnik se može napuniti metodom "natoci()". (radi jednostavnosti pretpostavljen je beskonačni kapacitet). Klasa također ima metodu "vozi()" u kojoj se zadaje duljina puta u km te se pomoću atributa "potrosnja" izračuna koliki je spremnik pun (ili prazan) na kraju puta. Svaka podklasa definira svoju vlastitu vrijednost tog atributa. (radi jednostavnosti uzeta vrijednost je odprilike zadana i definirana je ista za cijelu klasu. Tako imamo: auto = 8, kamion = 30, motor = 5)

Metode "srednja_vrijednost_godina" i "rangiraj po godinama" vrše funkcionalnost zadanu u zadatku, i to po atributu "godina proizvodnje".S obzirom da su ostale 3 klase izvedene iz ove klase one također imaju ove 2 metode ujedno sa min 4 atributa tako da su ti uvjeti zadovoljeni.

```
class Auto(Vozilo): # nasljeđivanje

    def __init__(self, marka, model, godina,) -> None:
        super().__init__(marka, model, godina)
        self.potrosnja = 8

    def trubi(self):
        print ("honk honk")
```

Podklasa auto je "basic" podklasa razreda Vozilo gdje samo nadjačavamo metodu trubi. Ovdje imamo prvi slučaj nasljeđivanja.

```
class Kamion(Vozilo): # nasljeđivanje
    def __init__(self, marka, model, godina,) -> None:
        super().__init__(marka, model, godina)
        self.potrosnja = 30
        self.teret = 0

    def trubi(self):
        print("Beeeeeep")

    def utovari_teret(self, kg):
        self.teret += kg
    def istovari_teret(self):
        self.teret = 0
```

U podklasi Kamion također nadjačavamo metodu trubi te dodajemo novi atribut "teret" koji prevozimo. Također imamo 2 metode vezane za utovar i istovar tereta.

```

class Motor(Vozilo): # nasljeđivanje
    def __init__(self, marka, model, godina, sjedalo_za_suvozaca) -> None:
        super().__init__(marka, model, godina)
        self.potrosnja = 5
        self.sjedalo_za_suvozaca = sjedalo_za_suvozaca

    def trubi(self):
        print("Vroooooom")

    def pokupi_prijatelja(self):
        if self.sjedalo_za_suvozaca :
            print("nemam mjesta")
        else: print("skoci gore")

```

U inicijalizaciji podklase Motor dodajemo još jedan atribut “sjedalo_za_suvozaca” koji je boolean i označava da li motor ima mjesta za još jednog.

U metodi “pokupi prijatelja()” ako imamo mjesta pokupit ćemo prijatelja a ako nemamo ništa s toga.

Idemo sad na program testiranja.

```

auti = [
    Auto("Volkswagen", "Golf", 2015),
    Auto("Toyota", "Corolla", 2018),
    Auto("Ford", "Focus", 2017),
    Auto("BMW", "3 Series", 2020),
    Auto("Audi", "A4", 2019)
]

# Testiranje funkcionalnosti klase Auto

srednja_vrijednost = Vozilo.srednja_vrijednost_godina(auti)
rangirani = Vozilo.rangiraj_po_godinama(auti)

print(srednja_vrijednost)
for v in rangirani:
    print(v)
print()

for auto in auti:
    auto.natoci(100) # Natoci spremnik goriva
    auto.vozi(50)   # Vozi automobil 100 km
    auto.trubi()    # Svaki automobil trubi

```

Instancirali smo 5 objekta tipa "Auto" te testirali metode "srednja_vrijednost_godina" i rangiraj po godinama.

```
2017.8
Volkswagen,Golf,2015
Ford,Focus,2017
Toyota,Corolla,2018
Audi,A4,2019
BMW,3 Series,2020
```

Ovo je izlaz gdje vidimo kako su vozila točno sortirana po godinama.

U metodama natoci() i vozi() primjenjujemo "nasljeđivanje".Kako to? Zato jer su to metode nadklase a mi ih možemo koristiti jer smo ih nasljedili.

```
onk honk
honk honk
honk honk
honk honk
honk honk
```

Izlaz truba.

```
kamioni = [
    Kamion("MAN", "TGX", 2016),
    Kamion("Scania", "R-Series", 2017),
    Kamion("Volvo", "FH", 2019),
    Kamion("Mercedes-Benz", "Actros", 2018),
    Kamion("Iveco", "Stralis", 2020)
]

srednja_vrijednost = Vozilo.srednja_vrijednost_godina(kamioni)
rangirani = Vozilo.rangiraj_po_godinama(kamioni)

print(srednja_vrijednost)
for v in rangirani:
    print(v)

print()

for kamion in kamioni:
    kamion.natoci(200) # Natoci spremnik goriva
    kamion.utovari_teret(1000)
    kamion.vozi(100) # Vozi kamion 200 km
    kamion.trubi() # Svaki kamion trubi
    kamion.istovari_teret()

print()
```

U testiranju kamiona većina stvari je isto samo svaki kamion dodatno utovaruje teret te ga istovaruje nakon vožnje.

```
2018.0
MAN, TGX, 2016
Scania, R-Series, 2017
Mercedes-Benz, Actros, 2018
Volvo, FH, 2019
Iveco, Stralis, 2020

Beeep
Beeep
Beeep
Beeep
Beeep
```

Izlaz srednje vrijednosti, sortiranih kamiona i beeeep beeeep truba.

```
motori = [
    Motor("Honda", "CBR600RR", 2015, True),
    Motor("Kawasaki", "Ninja ZX-10R", 2017, False),
    Motor("Yamaha", "YZF-R6", 2018, True),
    Motor("Suzuki", "GSX-R1000", 2019, False),
    Motor("Ducati", "Panigale V4", 2020, True)
]

srednja_vrijednost = Vozilo.srednja_vrijednost_godina(motori)
rangirani = Vozilo.rangiraj_po_godinama(motori)

print(srednja_vrijednost)
for v in rangirani:
    print(v)
print()

# Testiranje funkcionalnosti klase Motor
for motor in motori:
    motor.natoci(20)    # Natoci spremnik goriva
    motor.pokupi_prijatelja()
    motor.vozi(50)      # Vozi motor 50 km
    motor.trubi()       # Svaki motor trubi
```

U instanciranju motora dodajemo boolean atribut. Sve je isto samo što prvo pokušamo pokupiti prijatelja prije vožnje

```
2017.8
```

```
Honda, CBR600RR, 2015
Kawasaki, Ninja ZX-10R, 2017
Yamaha, YZF-R6, 2018
Suzuki, GSX-R1000, 2019
Ducati, Panigale V4, 2020
```

```
nemam mjesta
Vrooom
skoci gore
Vrooom
nemam mjesta
Vrooom
skoci gore
Vrooom
nemam mjesta
Vrooom
```

Izlaz. Neki motori imaju mjesta, neki nemaju. Svi motori idu vroom.

```
v1 = Auto("Volkswagen", "Golf", 2015)
v2 = Kamion("MAN", "TGX", 2016)
v3 = Motor("Honda", "CBR600RR", 2015, True)

v1:Vozilo; v2: Vozilo; v3:Vozilo

print()
#polimorfizam

v1.trubi()
v2.trubi()
v3.trubi()
```

Ovjde demonstriram polimorfizam. Definiram 3 podklase razreda Vozilo te svaki objekt "castam" u objekt Vozilo. Sada svaki objekt v1,v2,v3 gleda na svijet kroz naočale klase Vozilo. Kada pozovemo metodu trubi() za svaki objekt pozove se ta nadjačana metoda.

```
honk honk
Beeeeeep
Vrooom
```

Marko Đukin