

Odometer

An odometer (from the Greek *hodos* = road) is an apparatus that is used to measure the distance covered. This usually happens by counting the number of rotations of a wheel and multiplying this number by the circumference of the wheel. The best known form of an odometer is the speedometer as it is used in cars, for example. In a number of countries (like the US), the distance covered is expressed in miles. In other countries it is measured in kilometres. Many vehicles that are equipped with an electronic odometer, have the possibility to change between kilometres and miles. The accuracy with which the distance is displayed is 1/10 of a mile or a kilometre. When an odometer is replaced, a new odometer can be set up on a specified number of kilometres or miles.

Assignment

Your assignment consists of writing a class `OdoMeter`. Your class must support the following methods:

1. The initializing method `__init__` gets the total distance covered and the unit used as an optional parameter. The unit is given as a Boolean value that is `True` if the distance covered was measured in kilometres and `False` in the case that the distance is expressed in miles. Deduct the standard values for the parameters of the constructor from the example below.
2. The methods `__str__` and `__repr__` print the string representation of the odometer. Use the example below as a base for the format in which the data is printed.
3. The methods `__add__` and `__sub__`. These methods have an `OdoMeter` object as a first parameter and a numeric value that must be added to or subtracted from the distance covered as a second parameter. Note that the distance covered can not be negative. When the numeric value that is subtracted is larger than the current distance covered, the new distance in the odometer is zero.
4. The method `switch` that changes the unit of the odometer (from miles to kilometres or from kilometres to miles). The distance covered is also adjusted, knowing that 1 mile equals 1609,344 metres.

Note that the odometer asked shows the distance covered to 1/10 of a mile or a kilometre sharp, but that the odometer intern *floating point* uses precision 6 (that is six digits after the comma).

Example

```
>>> h = OdoMeter()
>>> print(h)
0.0 km
>>> h
OdoMeter(0.000000, True)
>>> h = h + 10.4
>>> print(h)
10.4 km
>>> g = OdoMeter(10, False)
>>> print(g)
10.0 mi
>>> g
OdoMeter(10.000000, False)
>>> print(g - 50)
```

```
0.0 mi
>>> h.switch()
>>> print(h)
6.5 mi
>>> h
OdoMeter(6.462260, False)
```

Een hodometer (van het Griekse *hodos* = weg, onder invloed van het Engels ook odometer genoemd) is een apparaat waarmee een afgelegde afstand gemeten wordt. Dit gebeurt in het algemeen door het aantal omwentelingen van een wiel te tellen en dit aantal te vermenigvuldigen met de omtrek van dat wiel. De meest bekende vorm van een hodometer is de kilometerteller zoals die gebruikt wordt in bijvoorbeeld een auto. In een aantal landen (zoals de VS), wordt de afgelegde afstand weergegeven in mijl. In andere landen wordt dan weer gemeten in kilometer. Veel voertuigen die uitgerust zijn met een elektronische hodometer hebben echter de mogelijkheid te schakelen tussen kilometer of mijl. De nauwkeurigheid waarmee de afgelegde afstand wordt weergegeven is 1/10 van een mijl of van een kilometer. Wanneer een hodometer wordt vervangen, dan kan een nieuwe hodometer ingesteld worden op een gespecificeerd aantal kilometer of mijl.

Opgave

Je opdracht bestaat erin een klasse `HodoMeter` aan te schrijven. Je klasse moet de volgende methoden ondersteunen:

1. De initialisatiemethode `__init__` krijgt als optionele parameters de totaal afgelegde afstand en de gebruikte eenheid mee. De eenheid wordt meegegeven als een Booleaanse waarde die `True` is als de afgelegde afstand in kilometer werd gemeten en `False` in het geval de afstand in mijl wordt uitgedrukt. Leidt de standaardwaarden voor de parameters van de constructor af uit onderstaand voorbeeld.
2. De methoden `__str__` en `__repr__` geven de stringrepresentatie van de hodometer terug. Baseer je voor de vorm waarin de gegevens worden uitgeschreven op het voorbeeld hieronder.
3. De methoden `__add__` en `__sub__`. Deze methoden hebben als eerste parameter een `HodoMeter` object en als tweede parameter een numerieke waarde die bij de afgelegde afstand moet opgeteld of afgetrokken worden. Merk op dat een afgelegde afstand niet negatief kan worden. Wanneer de numerieke waarde die wordt afgetrokken groter is dan de huidige afgelegde afstand, dan wordt de nieuwe afstand in de hodometer nul.
4. De methode `switch` die de eenheid van de hodometer verandert (van mijl naar kilometer of van kilometer naar mijl). Ook de afgelegde afstand wordt aangepast, wetende dat 1 mijl gelijk is aan 1609,344 meter.

Merk op dat de gevraagde hodometer de afgelegde afstand toont tot op 1/10 van een mijl of een kilometer nauwkeurig, maar dat de hodometer intern *floating point* precisie 6 gebruikt (dat is zes cijfers na de komma).

Voorbeeld

```
>>> h = HodoMeter()
>>> print(h)
0.0 km
>>> h
```

```
HodoMeter(0.000000, True)
>>> h = h + 10.4
>>> print(h)
10.4 km
>>> g = HodoMeter(10, False)
>>> print(g)
10.0 mi
>>> g
HodoMeter(10.000000, False)
>>> print(g - 50)
0.0 mi
>>> h.switch()
>>> print(h)
6.5 mi
>>> h
HodoMeter(6.462260, False)
```