

Università degli studi di Trieste



Programmazione a oggetti  
in SAGE

Stefano Piani

13 maggio 2014

In SAGE esistono molti tipi di variabili:

- ▶ interi
- ▶ decimali
- ▶ stringhe
- ▶ ...

In SAGE esistono molti tipi di variabili:

- ▶ interi
- ▶ decimali
- ▶ stringhe
- ▶ ...

E se ne volessimo creare delle altre?

In SAGE esistono molti tipi di variabili:

- ▶ interi
- ▶ decimali
- ▶ stringhe
- ▶ ...

E se ne volessimo creare delle altre? Possiamo usare le classi!

Una classe permette di creare nuovi tipi di “variabili”, detti oggetti, che sono composti dalle variabili “base” (o da altri oggetti).

Una classe permette di creare nuovi tipi di “variabili”, detti oggetti, che sono composti dalle variabili “base” (o da altri oggetti).

```
class lunghezza():  
    def __init__(self, un_numero, unita_di_misura):  
        self.number = un_numero  
        self.unit = unita_di_misura
```

## Esercizio

---

Scrivere una classe "tempo" che permetta di salvare oggetti che rappresentano intervalli di tempo. Il costruttore accetta in input due valori, un numero ed una stringa che può essere

- ▶ "secondi"
- ▶ "minuti"
- ▶ "ore"

La classe deve avere i metodi:

- ▶ `in_secondi()`
- ▶ `in_minuti()`
- ▶ `in_ore()`

## Metodi speciali di SAGE:

<code>__add__</code>	<code>x + y</code>	<code>__sub__</code>	<code>x - y</code>
<code>__mul__</code>	<code>x * y</code>	<code>__truediv__</code>	<code>x / y</code>
		<code>__floordiv__</code>	<code>x // y</code>
<code>__neg__</code>	<code>- x</code>		
<code>__init__</code>		<code>__call__</code>	<code>x()</code>
<code>__repr__</code>	<code>print x</code>	<code>__str__</code>	<code>str(x)</code>
<code>_latex_</code>	<code>latex(x)</code>		
<code>__cmp__</code>	<code>x &lt; y</code>	<code>__eq__</code>	<code>x == y</code>



## Esercizio

---

Creare una classe `triangolo` il cui costruttore prenda in input una lista di 3 punti. Definire poi un metodo `perimetro` che restituisca il perimetro del triangolo, un metodo `plot` che lo disegni in un grafico e sovrascrivere l'operatore `==` in modo che sia possibile verificare se due oggetti "triangoli" sono uguali (nel senso della geometria euclidea, cioè congruenti).

## Esercizio

---

Procurarsi il file `quanto_tempo.sws` e caricarlo in sage. Esso implementa, oltre alle già viste classi `lunghezza`, `tempo` e `velocità`, una nuova classe "astro" per rappresentare oggetti celesti. Si dia un'occhiata al codice e si scriva una funzione che prenda in input un astro e una velocità e dica quanto tempo ci vuole per arrivare su quell'astro con tale velocità.

## Esercizio

---

Aggiungere alla classe `triangoli` il metodo `area()` e sovrascrivere gli operatori di confronto (`<`, `>`, `<=` e `>=`) in base all'area dei medesimi.

## Esercizio

---

Creare una classe "poligono\_regolare" il cui costruttore prende in input due punti ed un numero naturale: i 2 punti sono il centro del poligono e un suo vertice, il numero rappresenta il numero di lati. Aggiungere un metodo `plot()`, un metodo `perimetro()`. Sovrascrivere l'operatore `==` con un metodo che verifichi se due poligoni sono uguali (nel senso che sono il medesimo sottoinsieme del piano cartesiano) e sovrascrivere gli operatori di ordine in base all'area. Implementare infine un metodo `sovrapposti()` che prende in input un angolo  $\alpha$  e restituisce l'area della figura che si ottiene dall'intersezione del poligono stesso e del poligono ottenuto ruotando il precedente attorno al centro di un angolo  $\alpha$ .