

# PROGRAMOZÁS 2

3.hét

# ÁTTEKINTÉS

- ▶ Függvények, operátorok túlterhelése
- ▶ Feladatok

# FÜGGVÉNYEK TÚLTERHELÉSE

- ▶ A túlterhelés jelentése, hogy több azonos nevű függvényt hozunk létre különböző paraméterekkel.

- ▶ Példa:

- ▶ `def PrintData(self)`

- ▶ `def PrintData(self,enable_description)`

# OPERÁTOROK TÚLTERHELÉSE

| Math-like Operators           |                             |                                                 |
|-------------------------------|-----------------------------|-------------------------------------------------|
| Expression                    | Method name                 | Description                                     |
| $x + y$                       | <code>__add__()</code>      | Addition                                        |
| $x - y$                       | <code>__sub__()</code>      | Subtraction                                     |
| $x * y$                       | <code>__mul__()</code>      | Multiplication                                  |
| $x / y$                       | <code>__div__()</code>      | Division                                        |
| $x == y$                      | <code>__eq__()</code>       | Equality                                        |
| $x > y$                       | <code>__gt__()</code>       | Greater than                                    |
| $x >= y$                      | <code>__ge__()</code>       | Greater than or equal                           |
| $x < y$                       | <code>__lt__()</code>       | Less than                                       |
| $x <= y$                      | <code>__le__()</code>       | Less than or equal                              |
| $x != y$                      | <code>__ne__()</code>       | Not equal                                       |
| Sequence Operators            |                             |                                                 |
| <code>len(x)</code>           | <code>__len__()</code>      | Length of the sequence                          |
| <code>x in y</code>           | <code>__contains__()</code> | Does the sequence y contain x?                  |
| <code>x[key]</code>           | <code>__getitem__()</code>  | Access element <i>key</i> of sequence x         |
| <code>x[key]=y</code>         | <code>__setitem__()</code>  | Set element <i>key</i> of sequence x to value y |
| General Class Operations      |                             |                                                 |
| <code>x=myClass()</code>      | <code>__init__()</code>     | Constructor                                     |
| <code>print(x), str(x)</code> | <code>__str__()</code>      | Convert to a readable string                    |
|                               | <code>__repr__()</code>     | Print a Representation of x                     |
|                               | <code>__del__()</code>      | Finalizer, called when x is garbage collected   |

TABLE 12.1 Python Special Method Names

# OPERÁTOROK TÚLTERHELÉSE

- ▶ Lényege, hogy ezen műveleteket definiáljuk a saját osztályainkra. Máshogy megfogalmazva leimplementáljuk, hogy az adott műveletet hogyan lehet végrehajtani, és nem kell plusz külön metódusokat megírni, hanem használható lesz a beépített is.
- ▶ Példa:
  - ▶ Class ComplexNumber (lásd példakód)
  - ▶ `__add__` () operátor implementálása a ComplexNumber osztályra.
  - ▶ Lesz két példányunk ebből, amelyek `c1` és `c2` lesznek
  - ▶ Értelmezhetővé válik `c1 + c2`.