LU04.L03 - Rechtecke

rectangle.py

```
"""Reference solution: Rectangle - Constructor and Equality.

Contains two implementations of the same concept:
- Rectangle:    "normal" class with manual constructor and __eq__
- RectangleDC:  dataclass with automatically generated constructor and
__eq__
"""


class Rectangle:
    """Rectangle with width and height (classic implementation)."""

    def __init__(self, width: float = 1.0, height: float = 1.0) ->
None:
        if width <= 0 or height <= 0:
            raise ValueError("width and height must be greater than 0")
        self.width = width
        self.height = height

    def __eq__(self, other: object) -> bool:
        if not isinstance(other, Rectangle):
            return NotImplemented
        return self.width == other.width and self.height ==
other.height

    def __repr__(self) -> str:
        return f"Rectangle(width={self.width}, height={self.height})"

    @property
    def width(self):
        """ returns the width """
        return self._width

    @width.setter
    def width(self, value):
        """ sets the width """
        self._width = value

    @property
    def heigth(self):
        """ returns the heigth """
        return self._heigth

    @heigth.setter
    def heigth(self, value):
```

```
        """ sets the heigth """
        self._heigth = value
```

rectangle_dc.py

```
from dataclasses import dataclass, field


@dataclass
class RectangleDC:
    """Rectangle with width and height (dataclass implementation).

    @dataclass automatically generates the constructor and __eq__ based
on
    the declared attributes. A custom __eq__ method is therefore not
    needed - the generated comparison checks class and all attributes.
    """

    _width: float = field(default=1.0)
    _height: float = field(default=1.0)

    def __post_init__(self) -> None:
        print (f"RectangleDC created with width={self.width} and
height={self.height}")
        if self.width <= 0 or self.height <= 0:
            raise ValueError("width and height must be greater than 0")

    @property
    def width(self):
        """ returns the width """
        return self._width

    @width.setter
    def width(self, value):
        """ sets the width """
        self._width = value

    @property
    def height(self):
        """ returns the height """
        return self._height

    @height.setter
    def height(self, value):
        """ sets the height """
        self._height = value
```

M320-LU04

René Probst, bearbeitet durch Marcel Suter

From:
https://wiki.bzz.ch/ - **BZZ - Modulwiki**

Permanent link:
https://wiki.bzz.ch/de/modul/m320_2024/learningunits/lu04/loesungen/rectangle?rev=1788433799

Last update: **2026/09/03 13:09**