



HUMBOLDT UNIVERSITY OF BERLIN

EINFÜHRUNG IN DAS WISSENSCHAFTLICHE RECHNEN

Documentation of Fraction Application Programming Interface and Command Line Interface Calculator

Christian Parpart & Kei Thoma

June 2, 2022

Contents

1	User Manual	3
2	Documentation	3
2.1	tools3.py	3
2.2	bruch.py	3
2.2.1	class Bruch()	3
2.2.2	Free Functions	4

1 User Manual

2 Documentation

2.1 tools3.py

In this module, we have the two functions to compute the greatest common divisor and the least common multiple. Here, there are no classes, just free functions.

- `ggt(arg1, arg2)` computes the greatest common divisor via Euclidean algorithm.
 - Arguments
 1. `arg1` (int): first integer
 2. `arg2` (int): second integer
 - Returns (int): greatest common divisor of the first and second integer
- `kgv(arg1, arg2)` determines the least common multiple, utilizing the greatest common divisor, computed by the function `ggt(arg1, arg2)`.
 - Arguments
 1. `arg1` (int): first integer
 2. `arg2` (int): second integer
 - Returns (int): least common multiple of the first and second integer.
- `main()` for testing purposes. Takes no arguments and returns none.

2.2 bruch.py

In this module, we have implemented the class `Bruch` that represents fractions.

2.2.1 class `Bruch()`

The objects of this class represent fractions.

Attributes

- `zaehler` (int): the numerator
- `nenner` (int): the denominator

Methodes

- `kuerzen(self)` reduces the fraction. Takes no arguments except for `self` and returns none.
- `__add__(self, other)` adds two fractions together via finding the greatest common divisor and reduces afterwards. The result is a new `Bruch` object.

- Arguments
 1. `other` (Bruch): another fraction
- Returns (Bruch): the sum of the two fractions
- `__repr__(self)` returns a printable string.
 - Arguments: none except for self
 - Returns (str): printable string
- `check_validity(self)` checks the fraction for validity. Returns false if the denominator is 0.
 - Arguments: none except for self
 - Returns (boolean): false if the denominator is 0, in any other case true

2.2.2 Free Functions

- `addiere(bruch_1, bruch_2)` adds two fractions into a new fractions
 - Arguments
 1. `bruch_1` (Bruch): first summand
 2. `bruch_2` (Bruch): second summand
 - Returns (Bruch): the sum of the two fractions

References

- [1] Bosch, Siegfried. *Algebra*. Springer-Verlag Berlin Heidelberg, 7th Edition, 2009.
- [2] Knuth, Donald. *The Art of Computer Programming Volume 2*. Prentice Hall, 3rd Edition, 1997.