

02465public

02465material > 02465public > Pipelines > #7585

passed Pipeline #7585 triggered 1 month ago by tsh

Final commit before refactoring of the toolbox to gym>=0.26.x

12 jobs for master in 140 minutes and 2 seconds (queued for 5 seconds)

ec4f528d

No related merge requests found.

Pipeline	Needs	Jobs 12	Tests 8
pull	build	test	deploy
✓ setup_38	✓ build_38	✓ test_38	✓ pages
✓ setup_39	✓ build_39	✓ test_39	pages.deploy
✓ setup_310	✓ build_310	✓ test_310	

1 Working with fractions (fractions.py)

In this problem, you have to implement the methods in the following class:

```
1 # cpp_course/fractions.py
2 class Fraction:
3     def __init__(self, n, m):
4         self.n = n
5         self.m = m
6
7     def __add__(self, other):
8         # Computes f = f1 + f2 (where f1 and f2 are both Fraction-objects and
9         #   f1= self, f2=other)
10        return f
11
12    def __mul__(self, other):
13        # Overwrite to implement f = f1 * f2
14        return f
15
16    def __truediv__(self, other):
17        # Overwrite to implement f = f1/f2, or more specifically self/other.
18        return f
19
20    def __str__(self):
21        """ Creates a string representation. You can use it as
22            -> print(str(Fraction(1,2))) to output 1/2"""
23        return f"({self.n} / {self.m})"
```

When done, you should be able to create two fractions objects and add them using the `+`-operator, which will be transformed to a call of the form `f1.__add__(f2)` as so:

```
1 # cpp_course/fractions.py
2 f1 = Fraction(1, 2) # Represents 1/2
```