

*Zadania przeznaczone dla rozwiązywania w trakcie zajęć*

1. Rozbuduj implementację klasy `Complex` (reprezentującej liczby zespolone) o przeciążone operatory.

Zaimplementuj metody:

- `__add__(self, other)`,
- `__radd__(self, other)`,
- `__iadd__(self, other)`,
- `__sub__(self, other)`,
- `__rsub__(self, other)`,
- `__isub__(self, other)`,
- `__mul__(self, other)`,
- `__rmul__(self, other)`,
- `__imul__(self, other)`,
- `__truediv__(self, other)`,
- `__rtruediv__(self, other)`,
- `__itruediv__(self, other)`,
- `__neg__(self)`,
- `__abs__(self)`.

Pozwalają one wykonywać dodawanie, odejmowanie, mnożenie, dzielenie (operatory dwuargumentowe) oraz obliczać liczbę przeciwną (operator jednoargumentowy), oraz obliczać moduł. Zadbaj o prawidłowe działanie operatorów dwuargumentowych, gdy drugi argument wywołania jest innego typu niż `Complex` (rozważ, jak powinny być wykonywane działania dla liczb rzeczywistych, całkowitych oraz łańcuchów znakowych; rozważ zgłaszanie błędów z wykorzystaniem mechanizmu wyjątków).

*Zadania przeznaczone do pracy własnej*

2. Zaimplementuj klasę `Fraction`, która będzie reprezentowała liczby wymierne. W klasie zaimplementuj odpowiednie operatory umożliwiające wykonywanie działań arytmetycznych (zwróć uwagę, aby odpowiednio skracać ułamki).