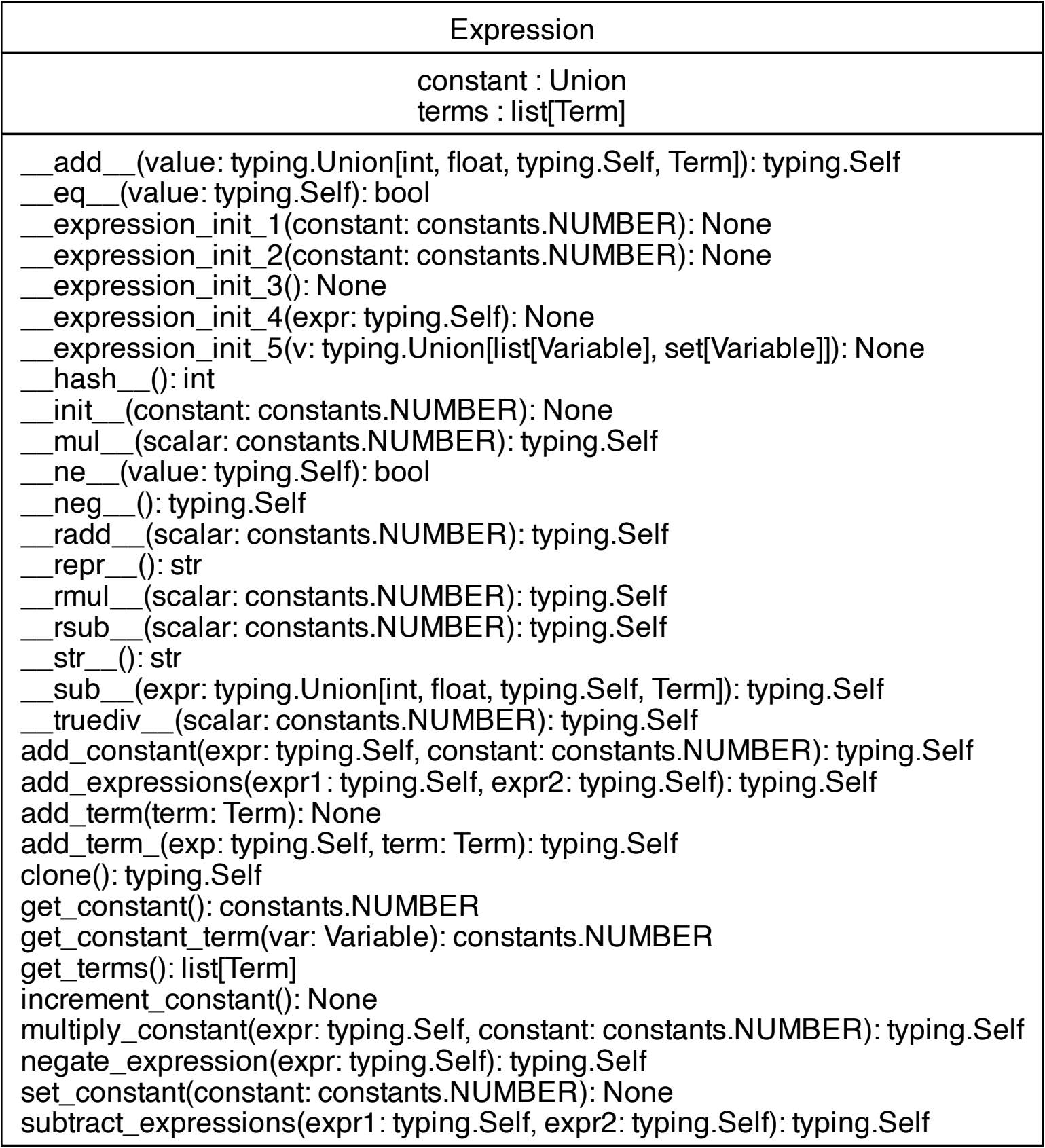
Expression
constant : Union
terms : list[Term]
__add__(value: typing.Union[int, float, typing.Self, Term]): typing.Self
__eq__(value: typing.Self): bool
__expression_init_1(constant: constants.NUMBER): None
__expression_init_2(constant: constants.NUMBER): None
__expression_init_3(): None
__expression_init_4(expr: typing.Self): None
__expression_init_5(v: typing.Union[list[Variable], set[Variable]]): None
__hash__(): int
__init__(constant: constants.NUMBER): None
__mul__(scalar: constants.NUMBER): typing.Self
__ne__(value: typing.Self): bool
__neg__(): typing.Self
__radd__(scalar: constants.NUMBER): typing.Self
__repr__(): str
__rmul__(scalar: constants.NUMBER): typing.Self
__rsub__(scalar: constants.NUMBER): typing.Self
__str__(): str
__sub__(expr: typing.Union[int, float, typing.Self, Term]): typing.Self
__truediv__(scalar: constants.NUMBER): typing.Self
add_constant(expr: typing.Self, constant: constants.NUMBER): typing.Self
add_expressions(expr1: typing.Self, expr2: typing.Self): typing.Self
add_term(term: Term): None
add_term_(exp: typing.Self, term: Term): typing.Self
clone(): typing.Self
get_constant(): constants.NUMBER
get_constant_term(var: Variable): constants.NUMBER
get_terms(): list[Term]
increment_constant(): None
multiply_constant(expr: typing.Self, constant: constants.NUMBER): typing.Self
negate_expression(expr: typing.Self): typing.Self
set_constant(constant: constants.NUMBER): None
subtract_expressions(expr1: typing.Self, expr2: typing.Self): typing.Self