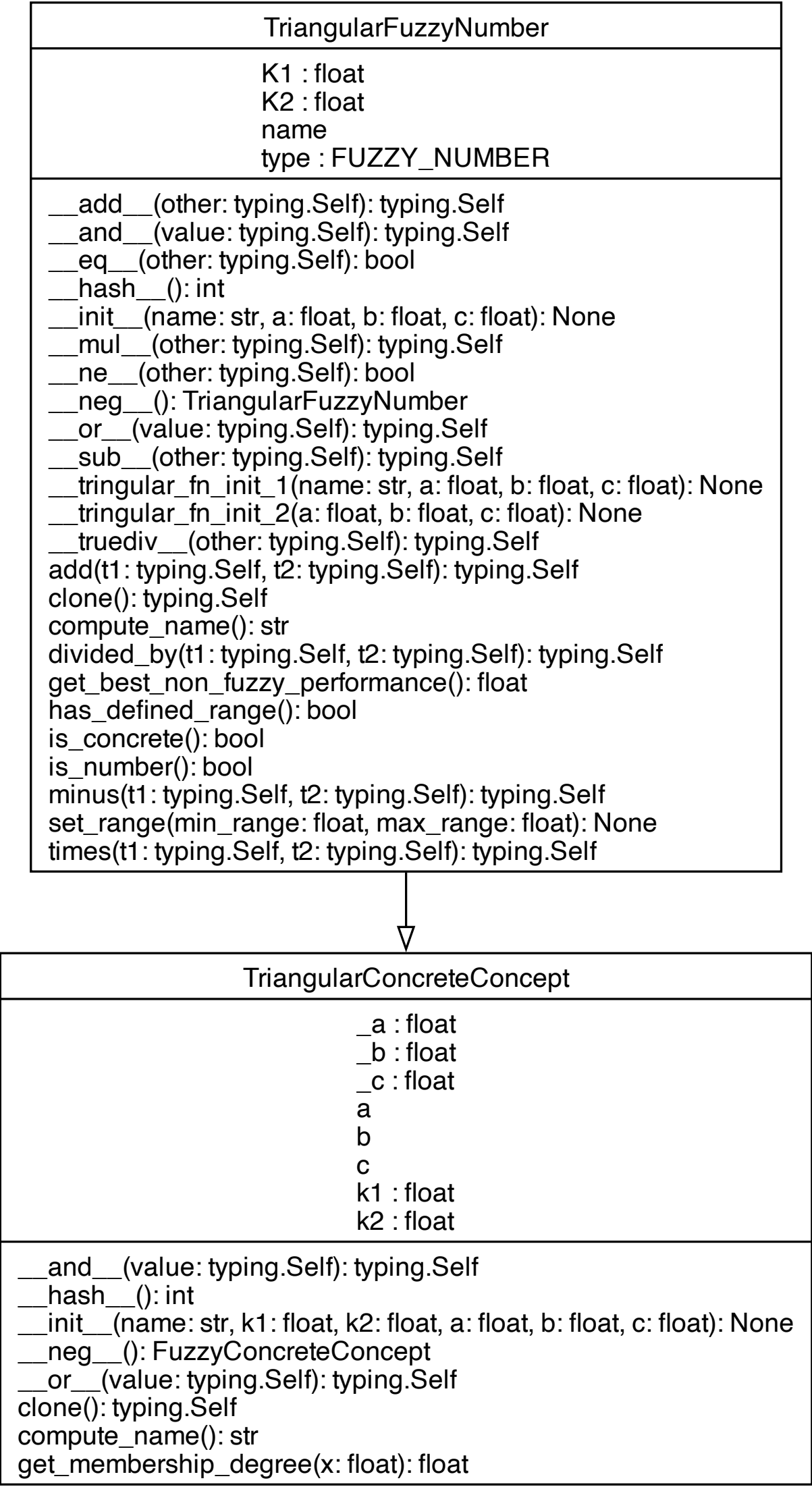
TriangularFuzzyNumber
K1 : float
K2 : float
name
type : FUZZY_NUMBER
__add__(other: typing.Self): typing.Self
__and__(value: typing.Self): typing.Self
__eq__(other: typing.Self): bool
__hash__(): int
__init__(name: str, a: float, b: float, c: float): None
__mul__(other: typing.Self): typing.Self
__ne__(other: typing.Self): bool
__neg__(): TriangularFuzzyNumber
__or__(value: typing.Self): typing.Self
__sub__(other: typing.Self): typing.Self
__tringular_fn_init_1(name: str, a: float, b: float, c: float): None
__tringular_fn_init_2(a: float, b: float, c: float): None
__truediv__(other: typing.Self): typing.Self
add(t1: typing.Self, t2: typing.Self): typing.Self
clone(): typing.Self
compute_name(): str
divided_by(t1: typing.Self, t2: typing.Self): typing.Self
get_best_non_fuzzy_performance(): float
has_defined_range(): bool
is_concrete(): bool
is_number(): bool
minus(t1: typing.Self, t2: typing.Self): typing.Self
set_range(min_range: float, max_range: float): None
times(t1: typing.Self, t2: typing.Self): typing.Self
TriangularConcreteConcept
_a : float
_b : float
_c : float
a
b
c
k1 : float
k2 : float
__and__(value: typing.Self): typing.Self
__hash__(): int
__init__(name: str, k1: float, k2: float, a: float, b: float, c: float): None
__neg__(): FuzzyConcreteConcept
__or__(value: typing.Self): typing.Self
clone(): typing.Self
compute_name(): str
get_membership_degree(x: float): float