

WeightedSumConcept
name : str
__and__(value: typing.Self): typing.Self __hash__(): int __init__(weights: list[float], concepts: list[Concept]): None __neg__(): Concept __or__(value: typing.Self): typing.Self clone(): typing.Self compute_atomic_concepts(): set[Concept] compute_name(): str get_roles(): set[str] replace(a: Concept, c: Concept): Concept

Concept
DEFAULT_NAME : str SPECIAL_STRING : str _name : str _type : ConceptType name num_new_concepts : int type
__and__(value: typing.Self): typing.Self __eq__(value: typing.Self): bool __iand__(value: typing.Self): typing.Self __init__(c_type: ConceptType, name: str): None __ior__(value: typing.Self): typing.Self __irshift__(value: typing.Self): typing.Self __ne__(value: typing.Self): bool __or__(value: typing.Self): typing.Self __rshift__(value: typing.Self): typing.Self __str__(): str is_atomic(): bool is_complemented_atomic(): bool

HasWeightedConceptsInterface
_weights : NoneType, typing.Optional[list[float]] weights
__init__(weights: typing.Optional[typing.Iterable[float]], concepts: typing.Iterable[Concept]): None