Gate + qubit count: int + basis_size: int + dtype: type + eval_bs(basis_state: int): State + __call__(state: State): State (*)(gate: Gate): Gate + mulit_gate(qubit_count: int, gate: Gate, apply_qubits: [int]): Gate + from eval bs(qubit count: int, eval bs: function): Gate + controlled gate(gubit count: int, gate: Gate, apply gubits: [int], control gubits[int]): Gate MatrixGate _init__(qubit_count: int, matrix) + join_gates(qubit_count: int, gate_list: [Gate]): Gate **FunctionalGate** _init__(qubit_count: int, _eval_bs: function, eval state: function) + <u>from_eval_state</u>(qubit_count: int, _eval_state: function): Gate

State

- + amplitudes: np.ndarray
- + dtype: type
- __init__(state:np.ndarray)
- + norm(): float
- + is_normalized(): bool
- + prob_of_bs(bs: int): float
- + prob_of_state(state: State): float
- + random_measure_bs(): int
- + random measure qubit(qubit. int): int
- + from_basis_state(bs: int): state
- ([])(bs: int): dtype
- (==)(state: State): bool
- (+)(state: State): State
- (-)(state: State): State
- (-)(): State
- (*)(number: complex): State
- (*)(state: State): complex
- (/)(number: complex): State