

Dokumentacja

IPinStateObserver – interfejs powyższej klasy, definiuje metodę do przechwycenia aktualizacji stanu poszczególnego pina:

- updatePinState

PinStateObserver – klasa zawiera metody, które odpowiadają za przechwycenie aktualizacji stanu pinu i trzymanie tego stanu.

IComponentFactory – interfejs powyższej klasy, definiuje metody do tworzenia komponentów tj. układy scalone itd.

- createChip

ComponentFactory – klasa zawiera metody, które odpowiadają za tworzenie komponentów, posiada pełną implementację, ma za zadanie zwracać odpowiednie unikalne identyfikatory dla stworzonego komponentu.

IPinHeaderFactory - interfejs powyższej klasy, definiuje metody do tworzenia komponentów tj. listwy kołkowe itd.

- createInputPinHeader, createOutputPinHeader

PinHeaderFactory - klasa zawiera metody, które odpowiadają za tworzenie listwy kołkowych, posiada pełną implementację, ma za zadanie zwracać odpowiednie unikalne identyfikatory dla stworzonego komponentu.

ComponentService – klasa zawiera metody, które odpowiadają za łączenie ze sobą komponentów za pomocą pinów. Dodatkowo sprawdza poprawność łączenia układów ze sobą.

- connect, checkConnection

SimulationService – klasa zawiera metody, które odpowiadają za symulację i optymalizację działania układów. Dodatkowo zawiera też set obserwatorów, które zawierają obecny stan pinu.

- simulation, notifyObserverAsTick, simulateTick, stationaryState, simulateTick

OptimizationService - klasa zawiera metody, które odpowiadają za optymalizację działania układów.

- optimize

Pin – klasa abstrakcyjna, po której będą dziedziczyć tworzone piny.