**Паттерны проектирования**

**ИСиТ,ПОИБМС-2**

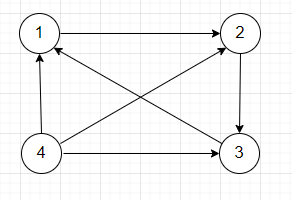
**Лекция 9**

**Поведенческие паттерны(продолжение)**

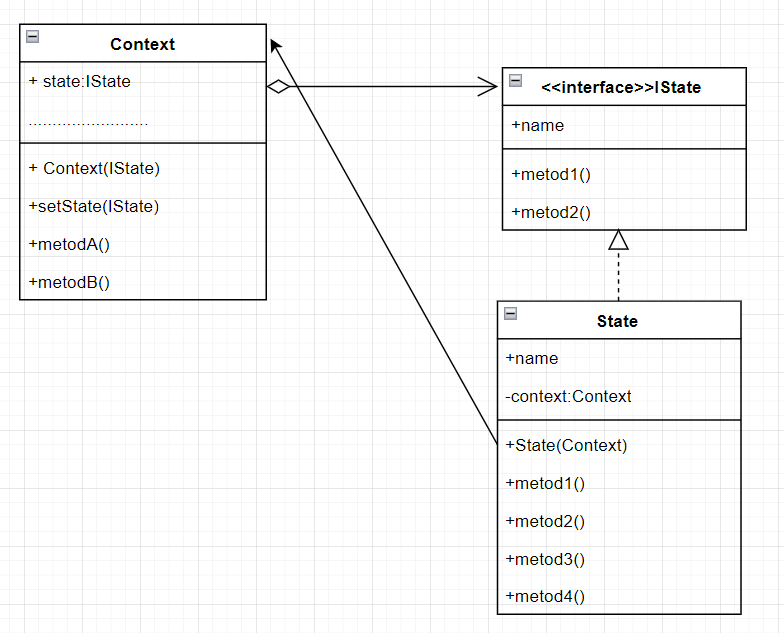
1. **Поведенческие паттерны:** взаимодействие программ.
2. **Поведенческие паттерны:** Chain of Responsibility (цепочка обязанностей), Command (Команда), Iterator (Итератор), Mediator (Посредник), Memento (Снимок), Observer (Обозреватель), State (Состояние), Strategy (Стратегия), Template method (шаблонный метод), Visitor (Посетитель).

**State**

1. **State:** программныйобъект, реализующий конечный автомат



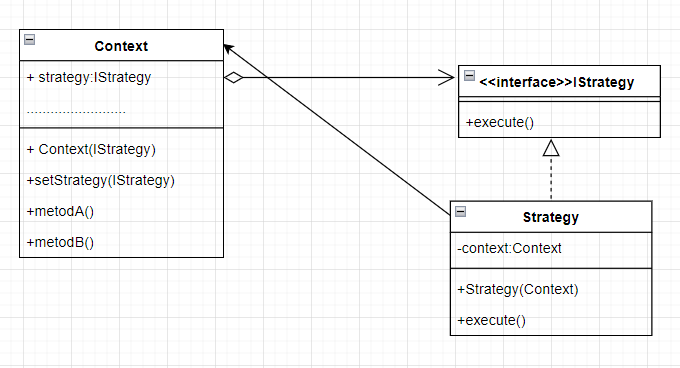
1. **State:UML**



1. **State:** автоматное программирование

**Strategy**

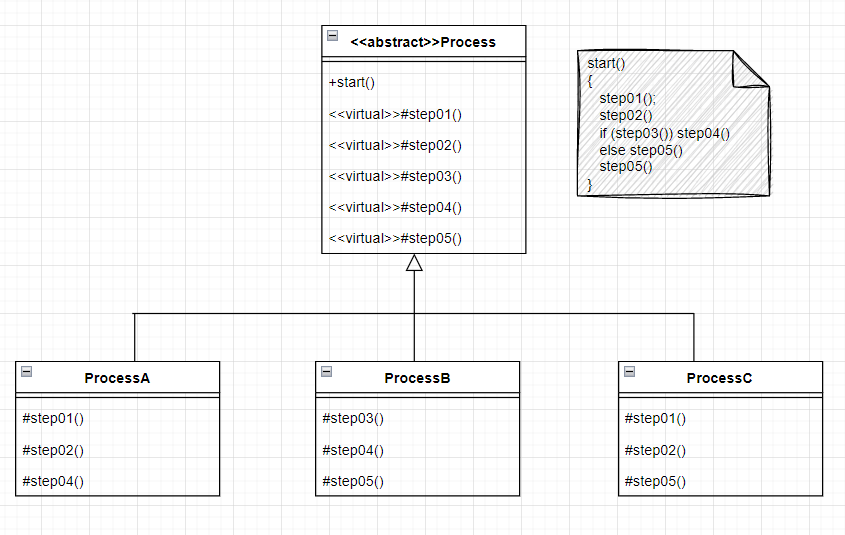
1. **Strategy:** алгоритм переносится в отдельный класс и динамически внедряется в контекст выполнения
2. **Strategy: UML**



1. **Strategy:** XMLHTTPREQUEST, callback-функция

**Template Method**

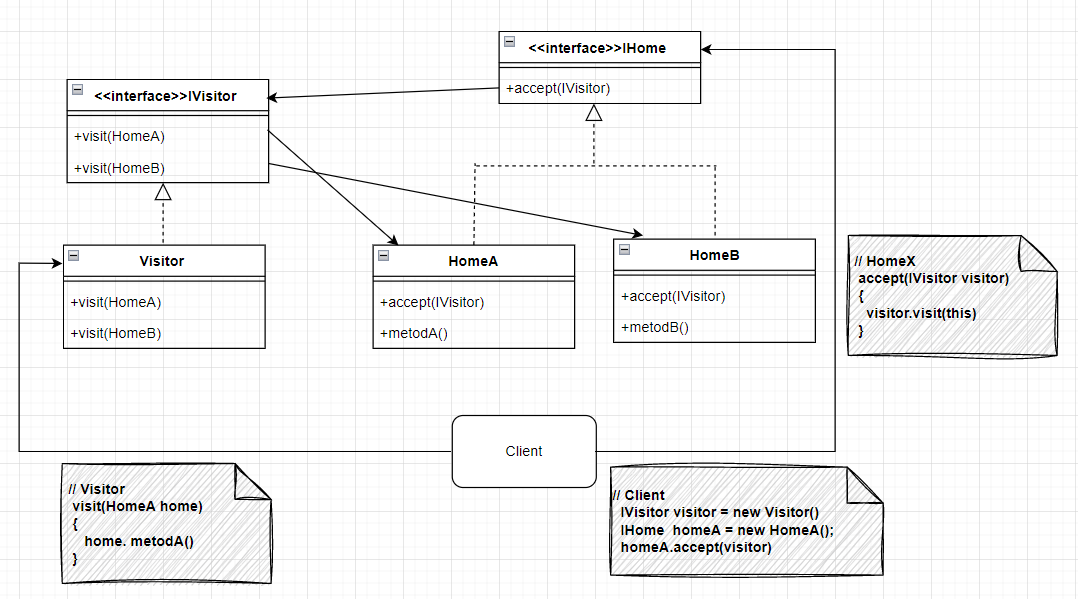
1. **Template Method:** скелет алгоритма, шаги которого могут быть переопределены



1. **Template Method:** Global.asaх**,** триггер в СУБД

**Visitor**

1. **Visitor:** набор алгоритмов (Visitor.visit()), которые могут исполнять методы разных окружений (HomeX)
2. **Visitor:** конкретный visitor «знает» в какой среде он исполняется



1. **КОНЕЦ**