

Метод поиска шаблонов проектирования в объектно- ориентированных программах

Студент: Сиромеха Роман Валерьевич

Руководитель: Рудаков Игорь Владимирович

Цель и задачи

Цель – разработать метод поиска шаблонов проектирования в объектно-ориентированных программах

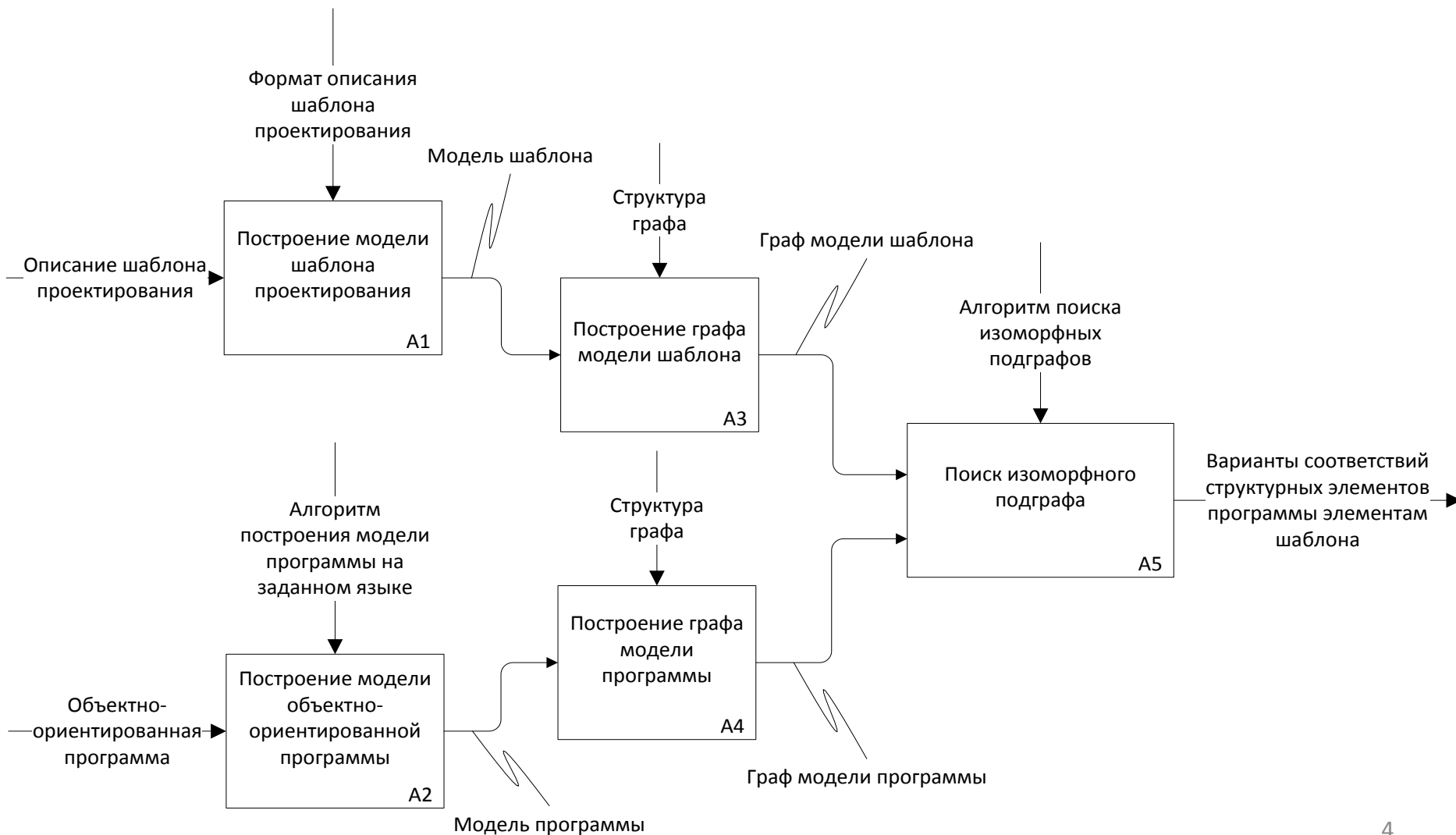
Задачи

1. Выполнить обзор существующих методов
2. Разработать модель объектно-ориентированной системы
3. Разработать алгоритм поиска шаблонов проектирования на основе алгоритма поиска изоморфных подграфов
4. Реализовать программный комплекс на основе разработанного метода
5. Провести исследование: выполнить поиск шаблонов проектирования в существующих проектах

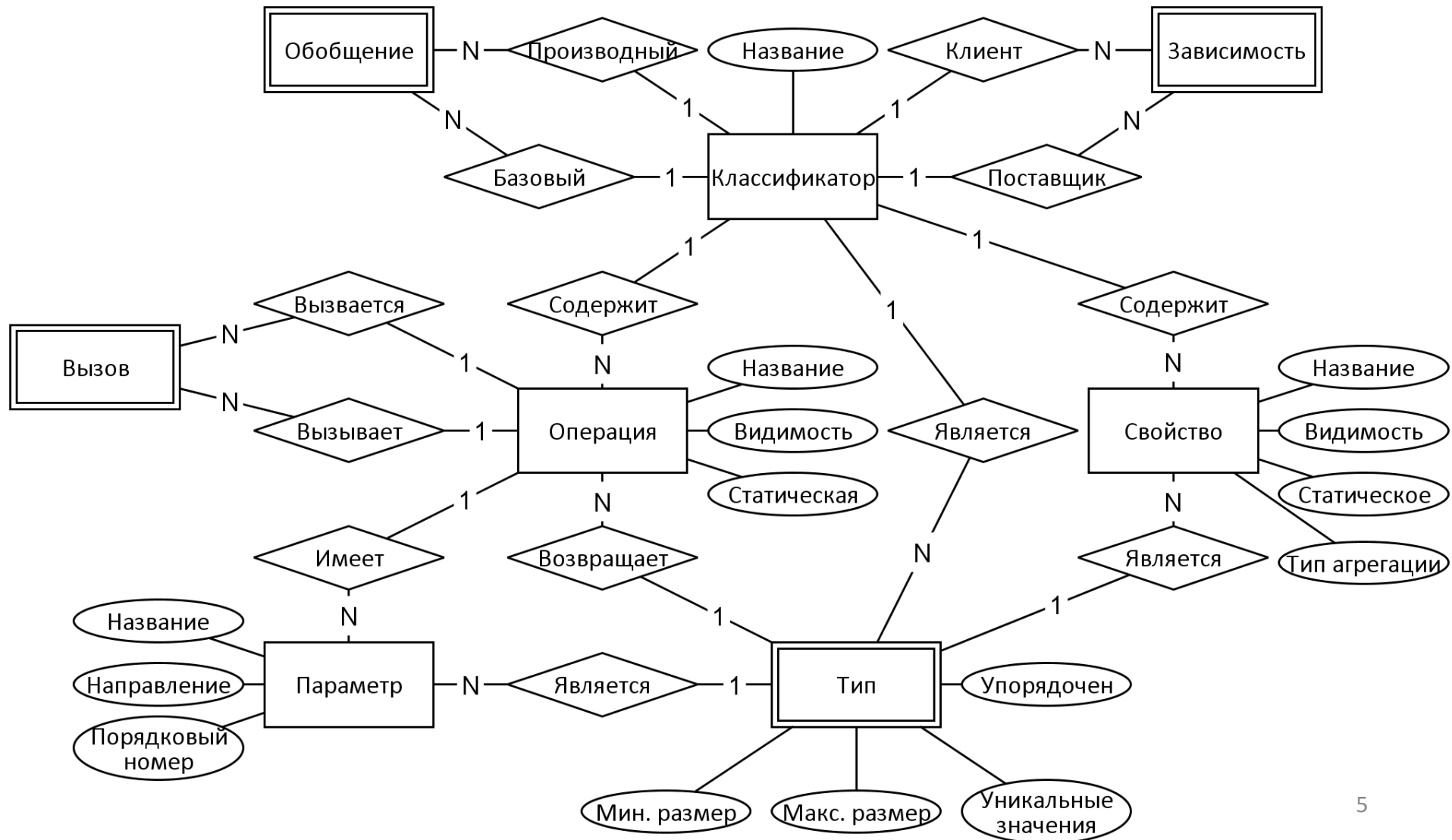
Существующие методы поиска шаблонов проектирования

Класс метода	Достоинства	Недостатки
На основе алгоритма поиска изоморфного подграфа с использованием меры схожести вершин	<ul style="list-style-type: none">• Полиномиальная сложность	<ul style="list-style-type: none">• Неточный результат• Находят отдельные элементы шаблона
На основе алгоритма поиска изоморфного подграфа с вычислением расстояния между графами	<ul style="list-style-type: none">• Полиномиальная сложность• Находят все элементы шаблона	<ul style="list-style-type: none">• Неточный результат

Метод поиска шаблонов проектирования на основе поиска изоморфных подграфов



Модель объектно-ориентированной системы

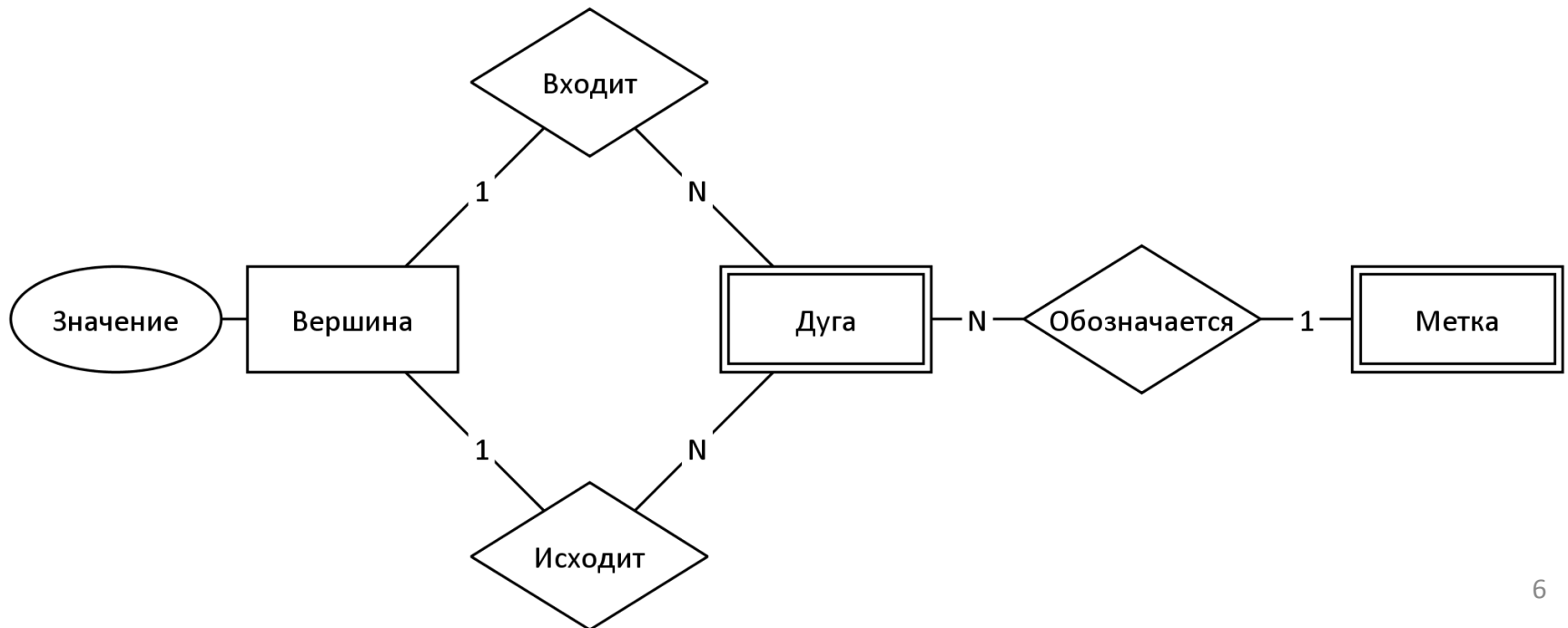


Ориентированный граф с множеством типов дуг

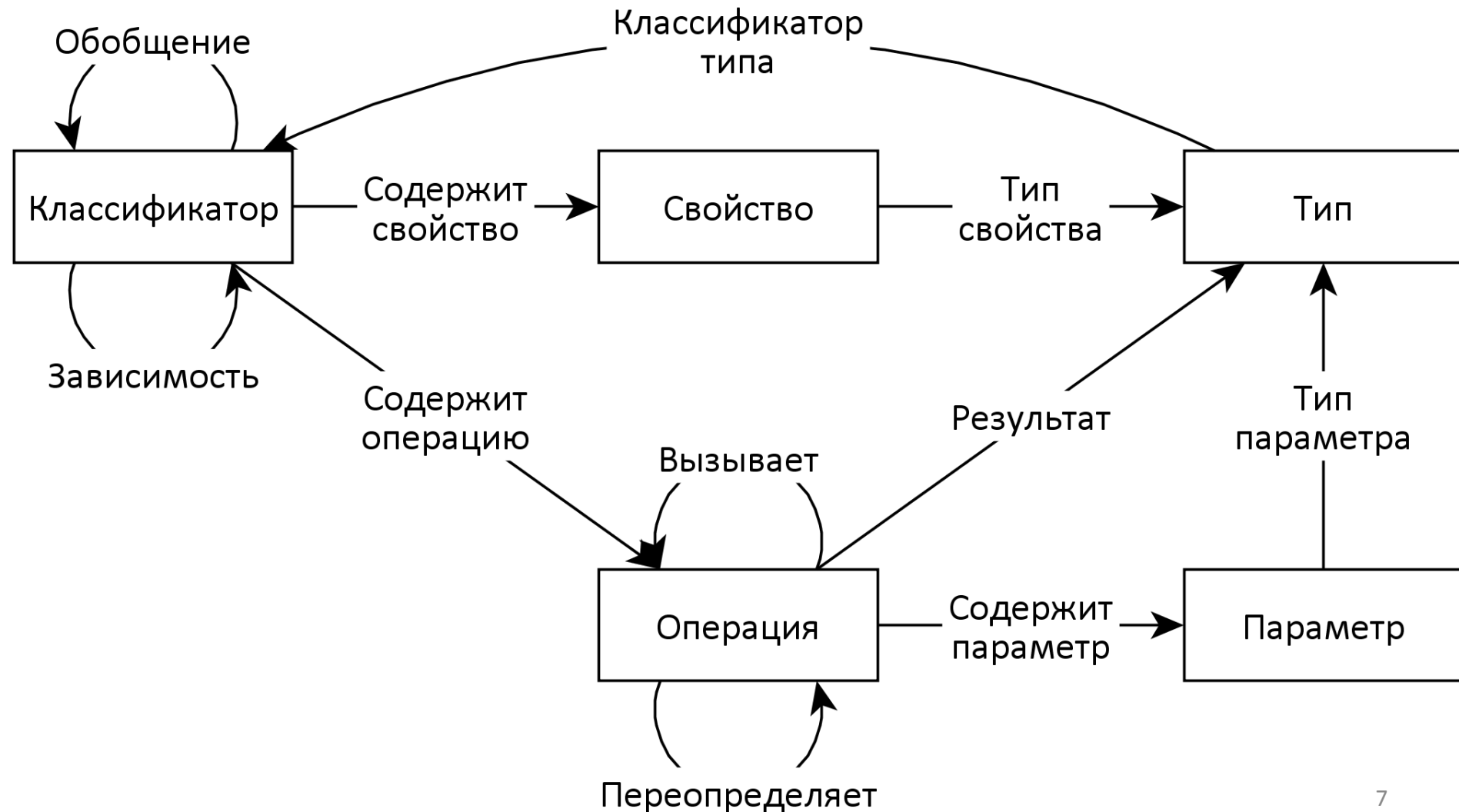
$$G = (V, L, E)$$

$V = \{v\}$ – множество вершин, $L = \{l\}$ – множество меток

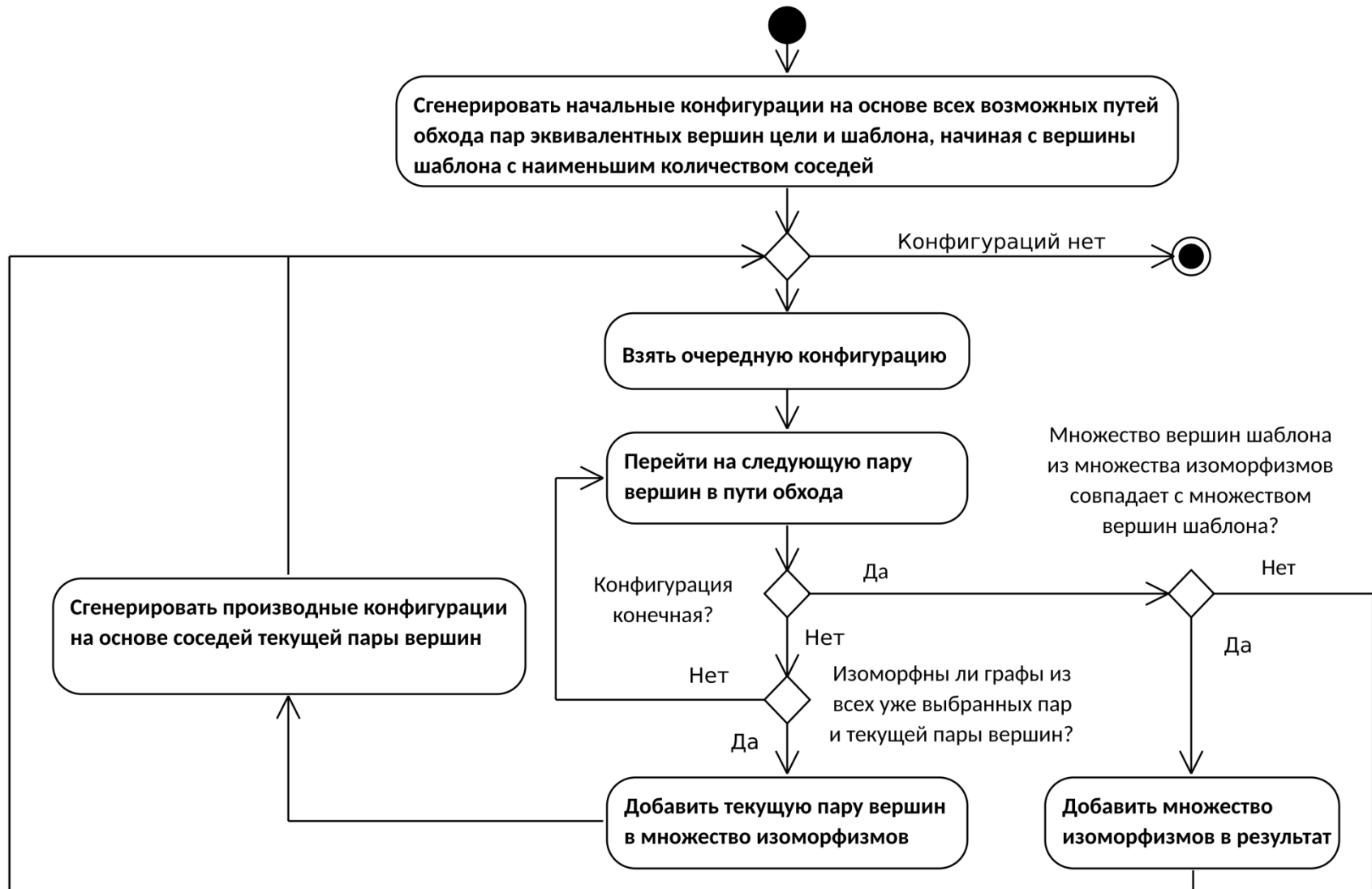
$E = \{(v_1, v_2, l) : v_1, v_2 \in V \wedge l \in L\}$ – множество дуг
($\forall v_1, v_2 \in V, l \in L$)($|\{(v_1, v_2, l)\}| = 1$)



Граф модели объектно-ориентированной системы



Алгоритм поиска изоморфных подграфов



Условие корректности результата алгоритма поиска изоморфных подграфов

$G_t = (V_t, L_t, E_t)$ – целевой граф

$G_p = (V_p, L_p, E_p)$ – граф шаблона

$I = \{(t, p): t \in V_t \wedge p \in V_p\}$ – изоморфизм

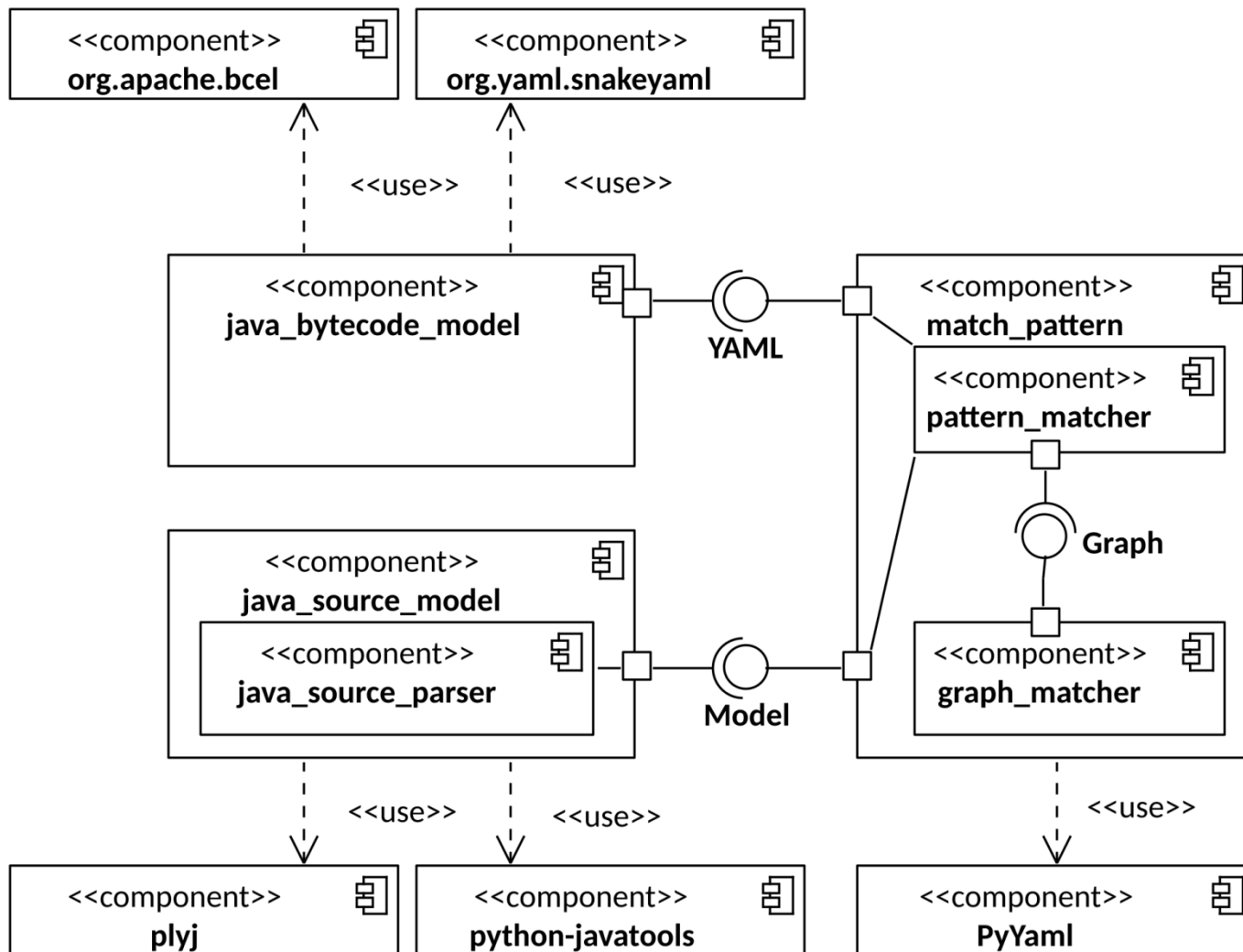
$$\forall (t, p) \in I$$

$$\left((\forall (p, p_x, l) \in E_p) (\exists (t, t_y, l) \in E_t: (t_y, p_x) \in I) \right)$$

$$\wedge$$

$$\left((\forall (p_a, p, l) \in E_p) (\exists (t_b, t, l) \in E_t: (t_b, p_a) \in I) \right)$$

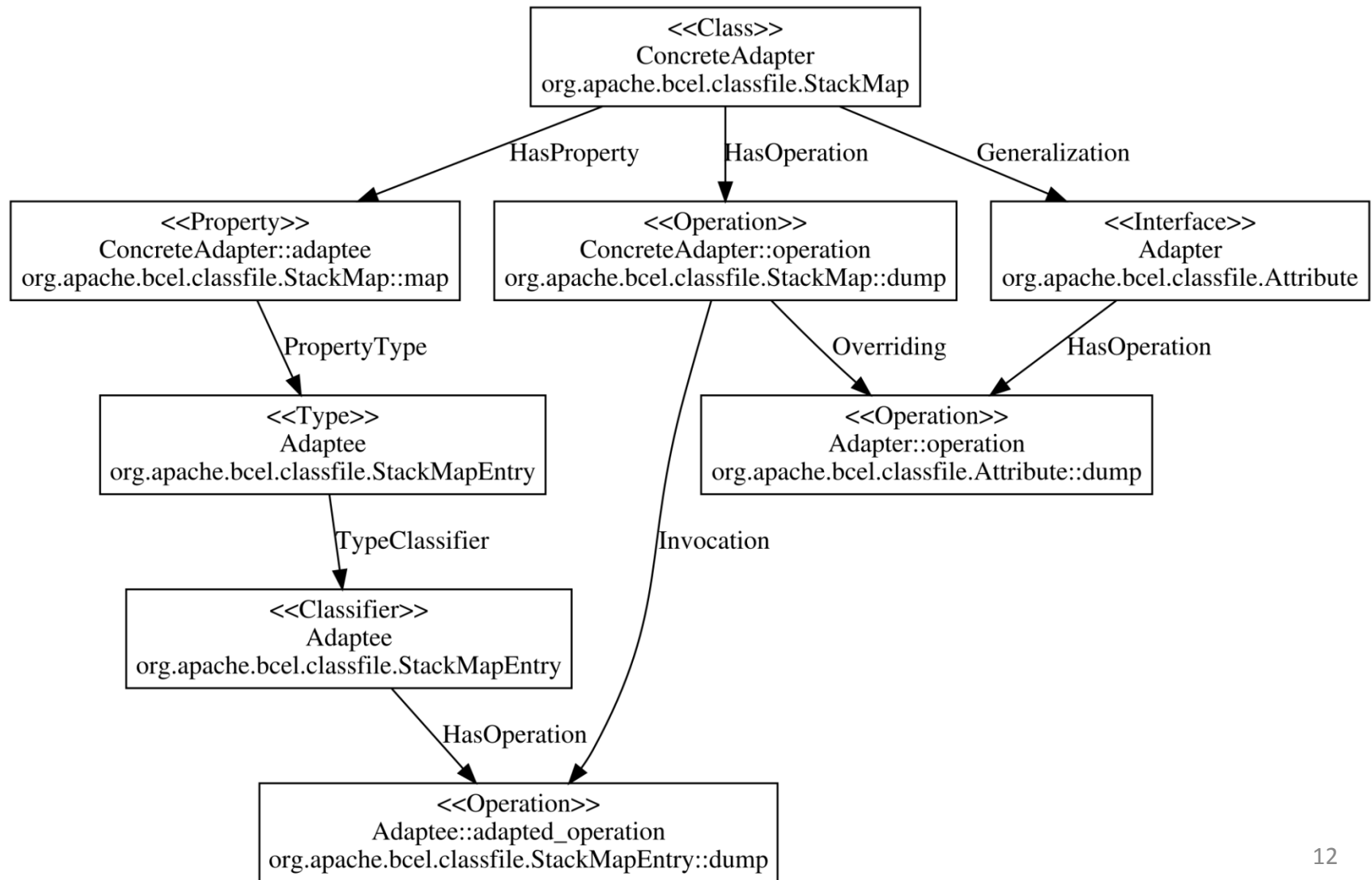
Компоненты и зависимости программного комплекса



Тестирование программного комплекса

- Модульные тесты (unittest, PyHamcrest)
 - graph_matcher
 - pattern_matcher
 - java_source_parser
- Функциональные тесты (pytest)
 - java_bytecode_module
 - match_pattern

Пример результата поиска шаблона «Адаптер» в «Apache BCEL»



Результаты поиска шаблонов проектирования

Проект	Airbnb Aerosolve (Java)	Apache BCEL (Java)	Apache Zookeeper (Java)	java- design- patterns	Netflix EVCache (Java)	scodec (Scala)
Абстрактная фабрика				+		
Адаптер		+		+		
Вызов переопределенного метода	+	+	+	+	+	+
Декоратор						
Мост	+	+	+	+	+	
Посетитель				+		
Хранитель						
Цепочка ответственности						

Выводы

- Выполнен обзор существующих методов
- Разработана модель объектно-ориентированной системы на основе UML-диаграммы классов
- Разработан алгоритм поиска шаблонов проектирования на основе алгоритма поиска изоморфных подграфов
- Разработан программный комплекс для поиска шаблонов проектирования в программах, компилируемых в байт-код виртуальной машины Java
- Проведено исследование: выполнен поиск шаблонов проектирования в ряде существующих программ и библиотек