

# Метод поиска шаблонов проектирования в объектно- ориентированных программах

Студент: Сиромаша Роман Валерьевич

Руководитель: Рудаков Игорь Владимирович

# Цель и задачи

**Цель работы** – разработать метод поиска шаблонов проектирования в объектно-ориентированных программах на основе алгоритма поиска изоморфных подграфов

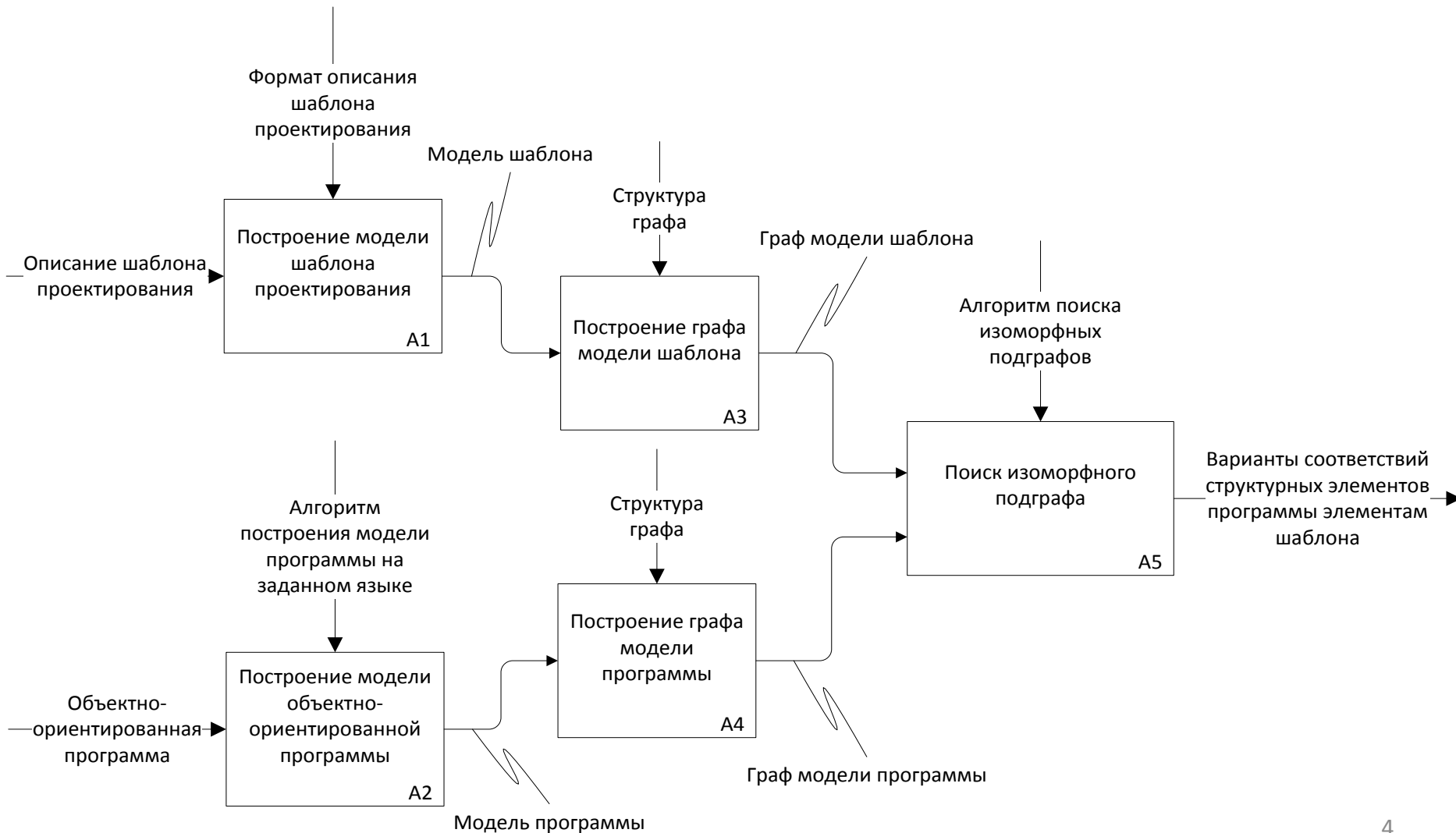
## **Задачи**

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"><li>• Полиномиальная сложность</li></ul>	<ul style="list-style-type: none"><li>• <b>Неточный результат</b></li><li>• Находят отдельные элементы шаблона</li></ul>
На основе алгоритма поиска изоморфного подграфа с вычислением расстояния между графами	<ul style="list-style-type: none"><li>• Полиномиальная сложность</li><li>• Находят все элементы шаблона</li></ul>	<ul style="list-style-type: none"><li>• <b>Неточный результат</b></li></ul>

# Метод поиска шаблонов проектирования на основе поиска изоморфных подграфов

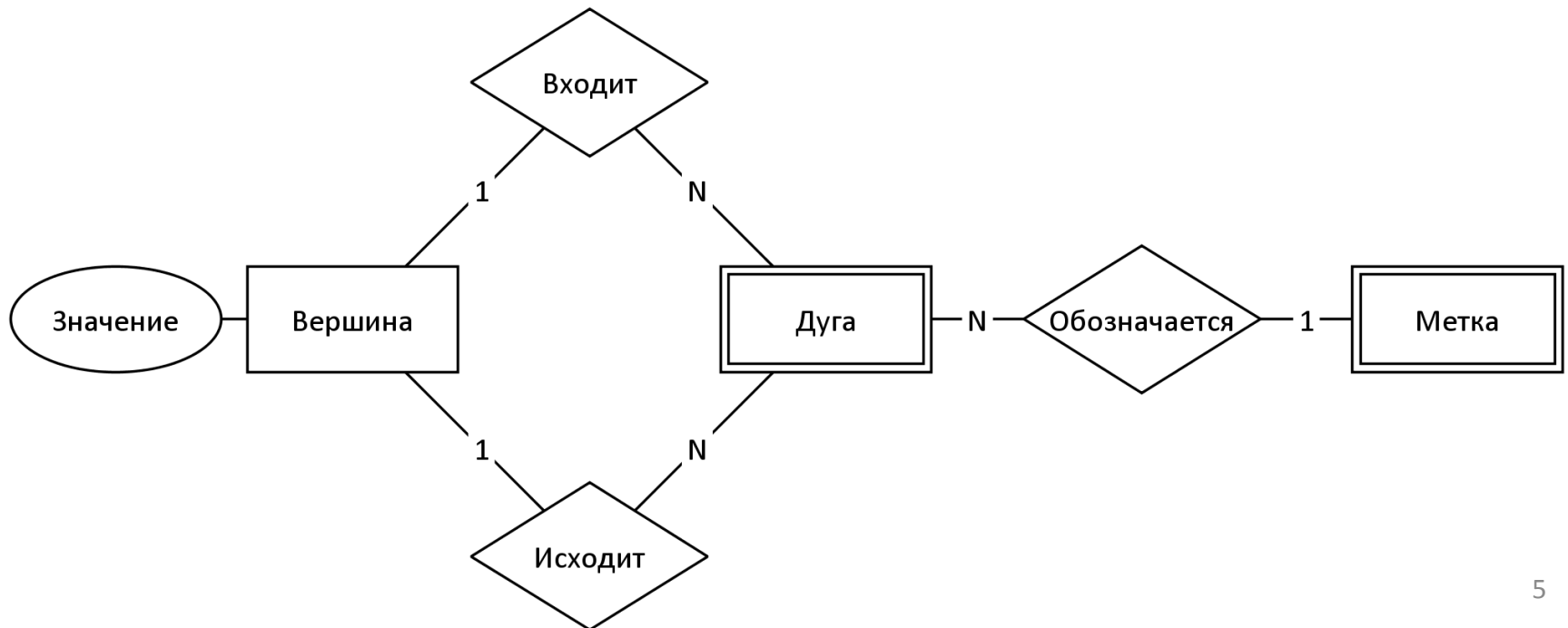


# Ориентированный граф с множеством типов дуг

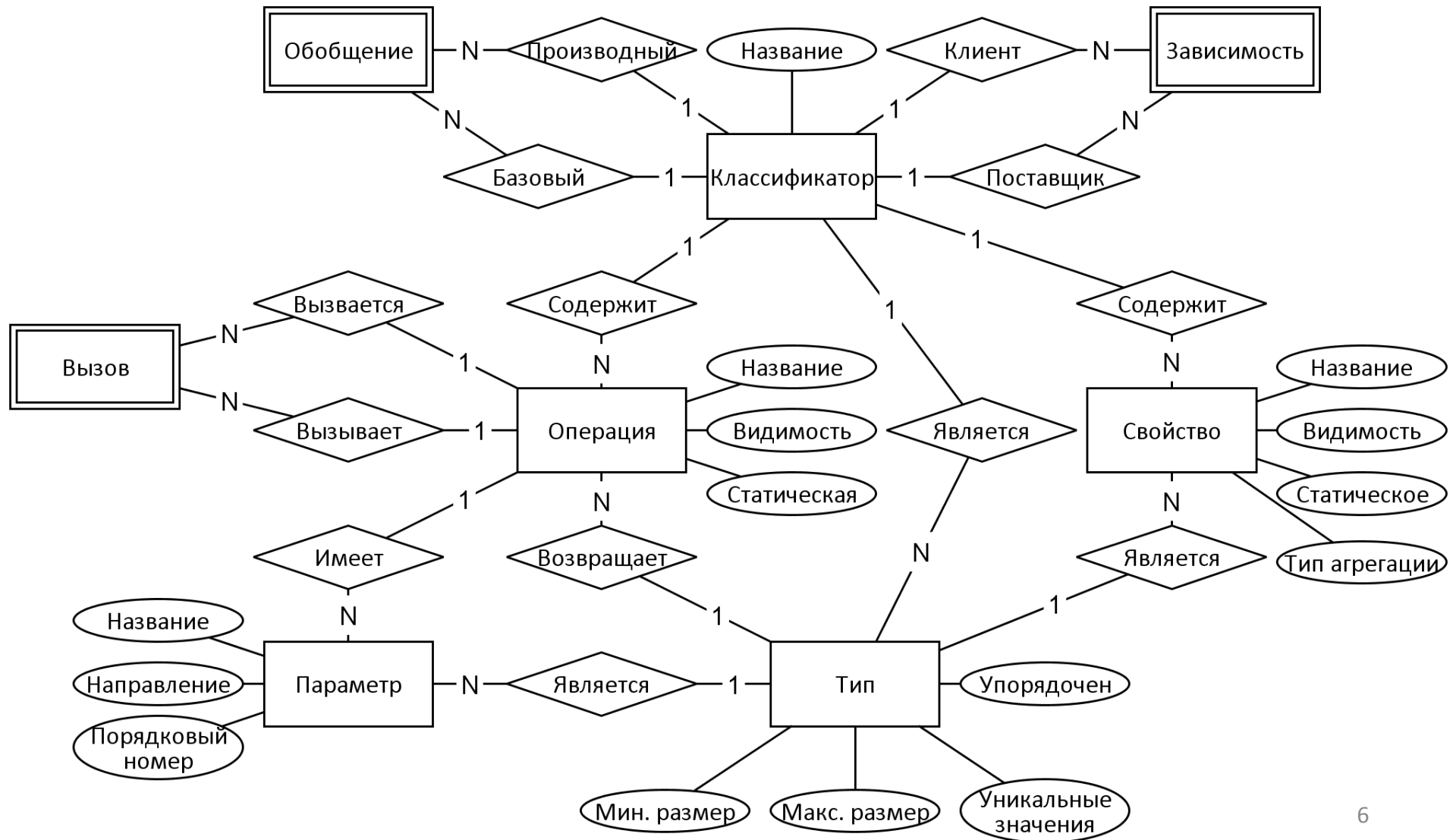
$$G = (V, L, E)$$

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

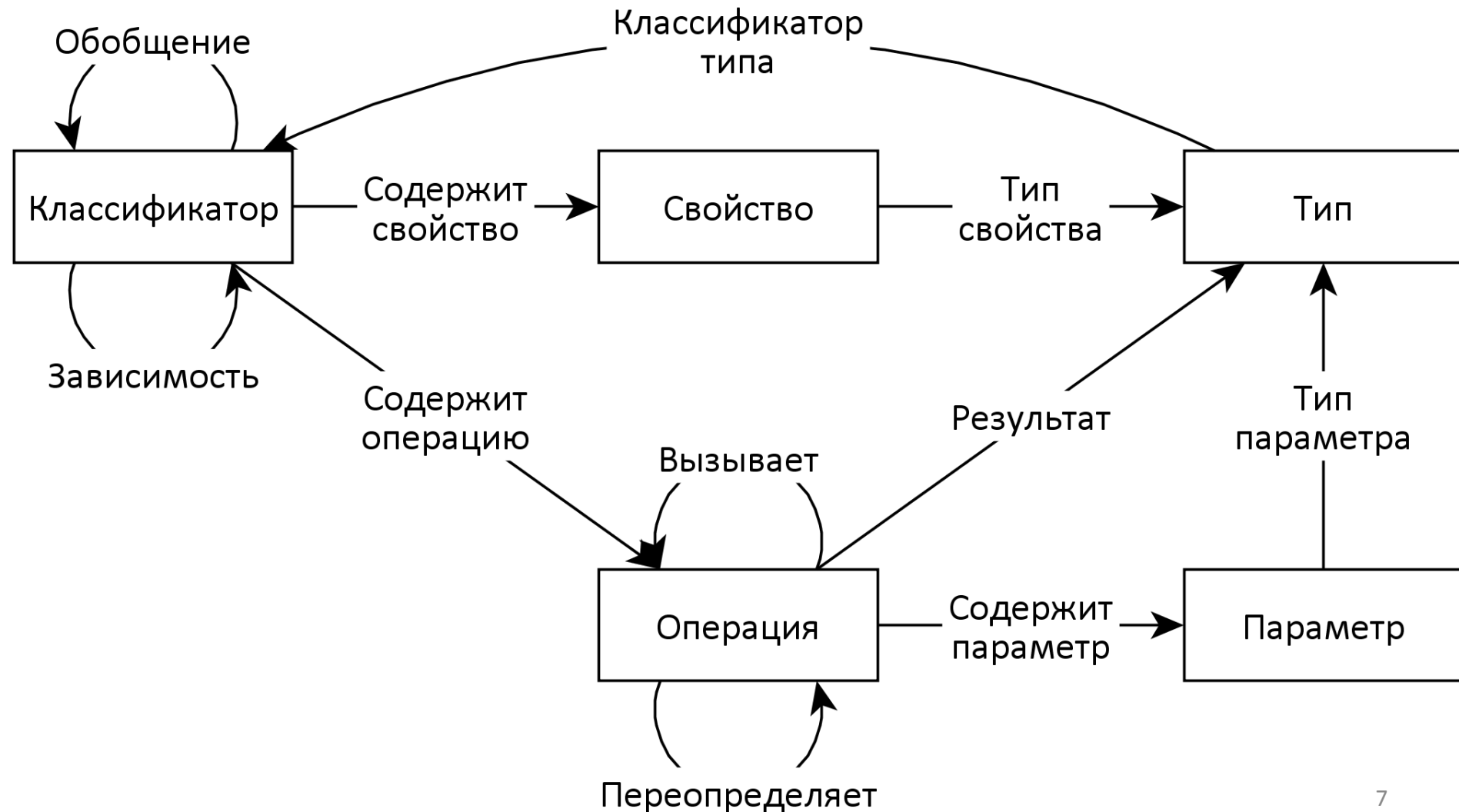
$E = \{(v_1, v_2, l) : v_1, v_2 \in V \wedge l \in L\}$  – множество дуг  
( $\forall v_1, v_2 \in V, l \in L$ )( $|\{(v_1, v_2, l)\}| = 1$ )



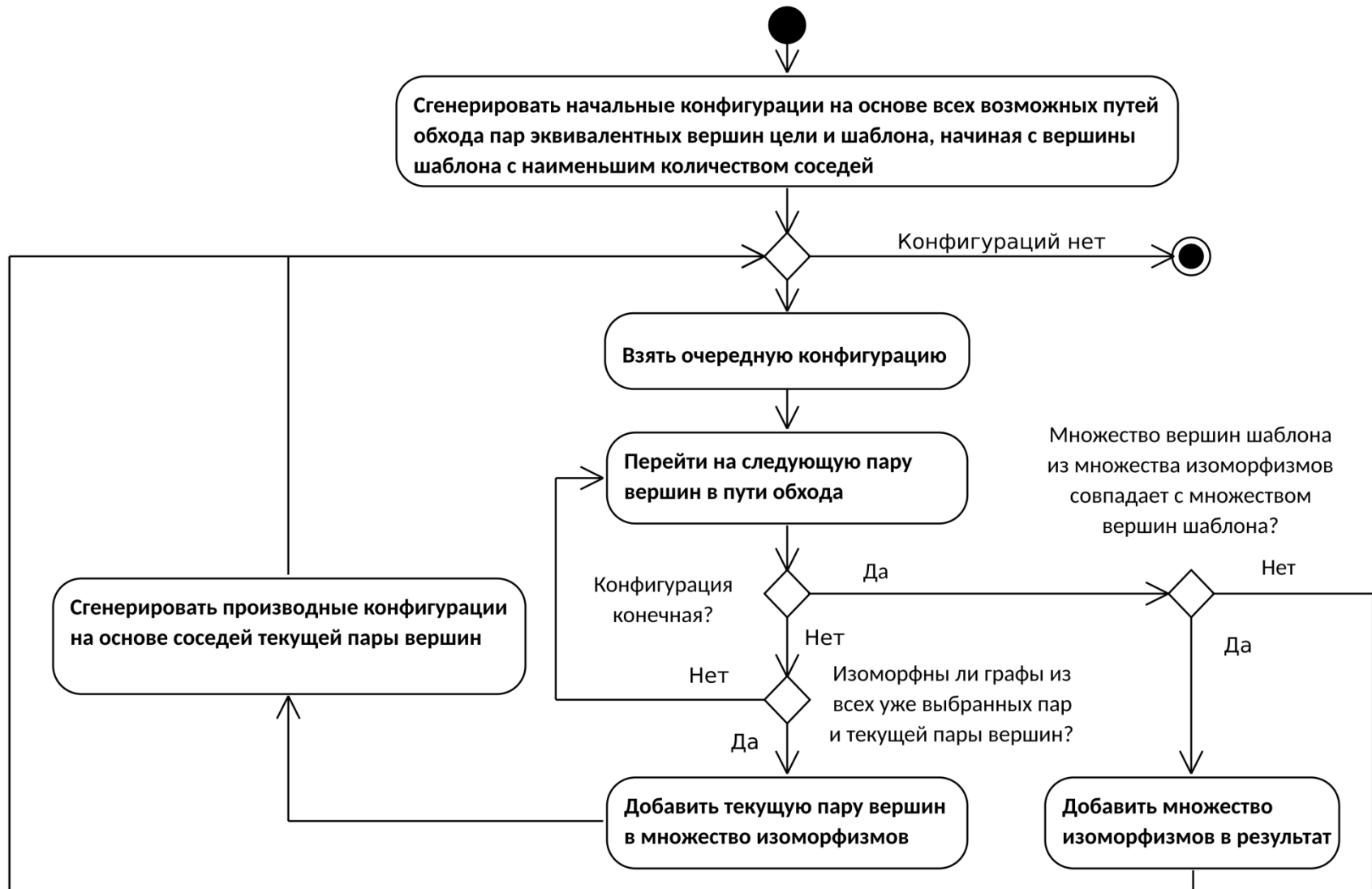
# Модель объектно-ориентированной системы



# 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

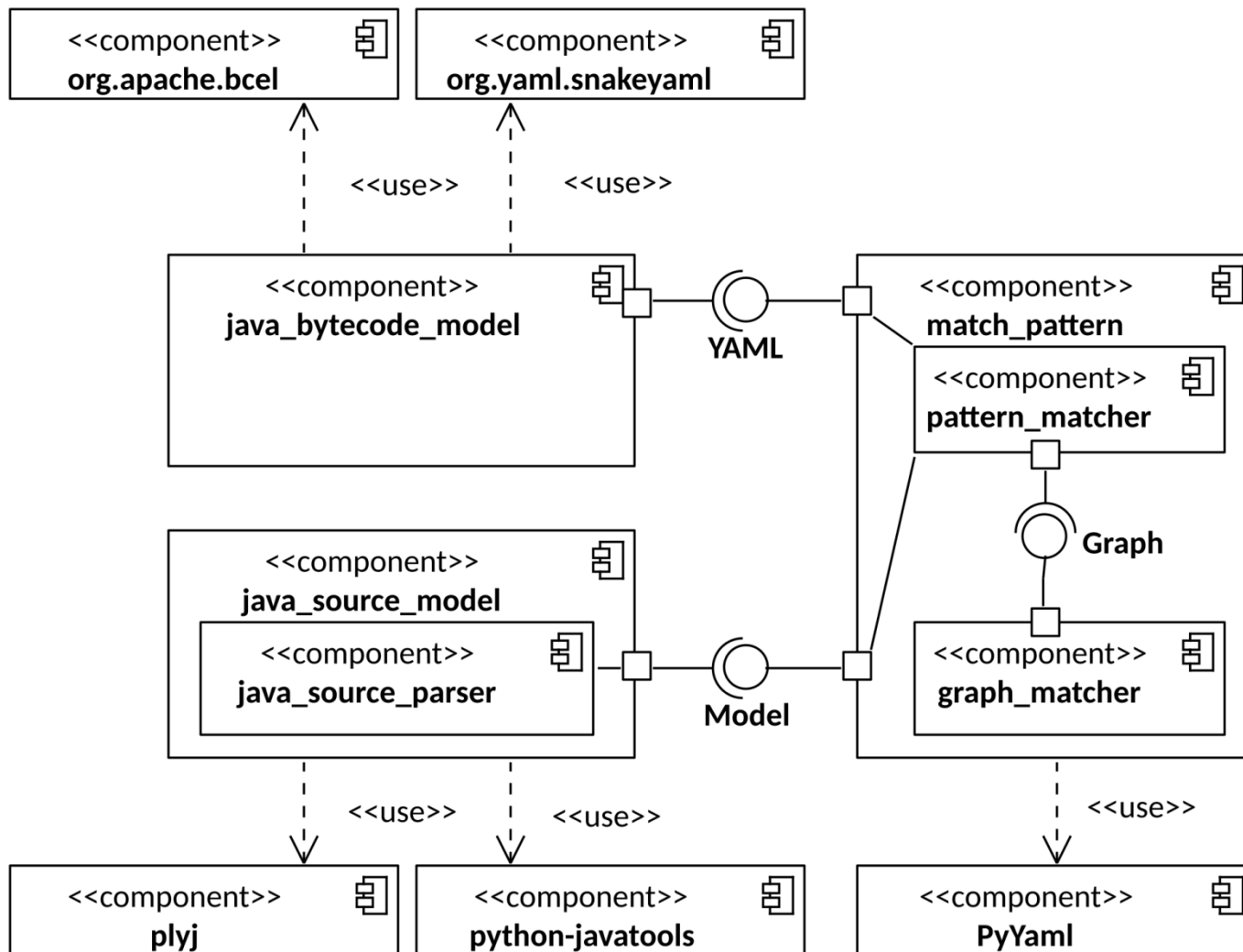
$$\forall (t, p) \in I$$

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

$$\wedge$$

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

# Компоненты и зависимости программного комплекса



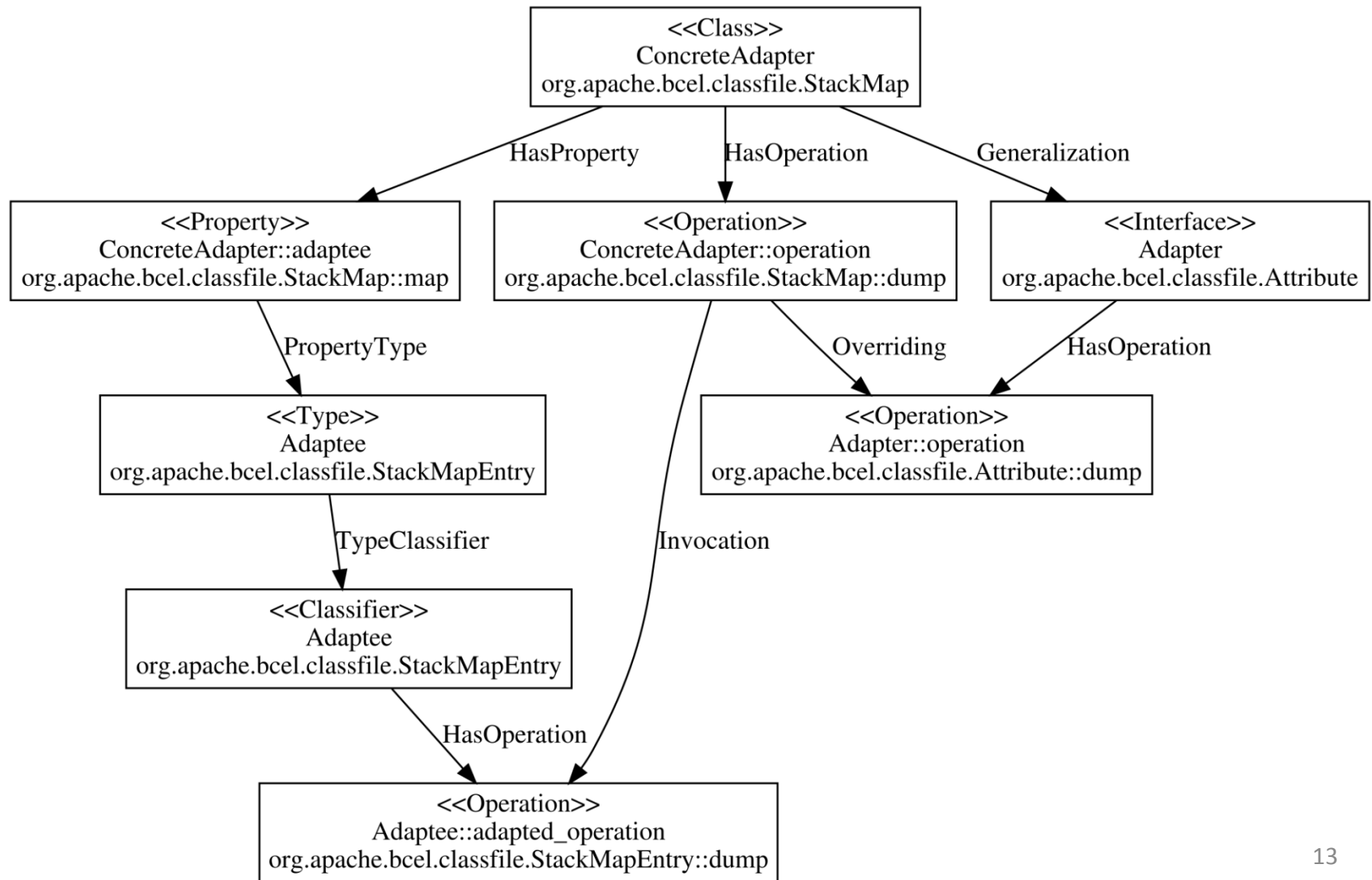
# Интерфейсы запуска программ

- **java\_bytecode\_model** строит модель программы собранной под виртуальную машину Java по .jar и .class файлам
  - `java -jar java_bytecode_model.jar <путь к файлу или директории> ... > model.yaml`
- **java\_source\_model** строит модель программы на языке Java
  - `python java_source_model <путь к директории с .java файлами> > model.yaml`
- **pattern\_model** строит модель шаблона проектирования по названию
  - `python pattern_model <имя шаблона> > pattern.yaml`
- **match\_pattern** выполняет поиск шаблона проектирования в модели программы
  - `python match_pattern.py model.yaml pattern.yaml`

# 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	Apache BCEL	Apache Zookeeper	java- design- patterns	Netflix EVCache	scodec
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 и реализован алгоритм поиска изоморфных подграфов
- Разработана программа для поиска шаблона проектирования в модели объектно-ориентированной программы
- Реализовано построение модели программы на основе байт-кода для виртуальной машины Java
- Реализовано построение модели программы на основе исходного кода Java
- Проведено исследование: выполнен поиск шаблонов проектирования в ряде существующих программ и библиотек