

**Графическая нотация наследования
автоматных классов**

Данил Шопырин
ЗАО «Транзас Технологии»

Анатолий Шальто
СПбГУ ИТМО

Типы программных систем

- Преобразующие системы
- Интерактивные системы
- Реактивные системы

Реактивные системы

- Средства проектирования и реализации:
 - *SDL*
 - *UML*
 - *SyncCharts*
 - *SWITCH*-технология

- Реактивные подсистемы ОО-систем
 - *State Design Pattern* и варианты его развития

Автоматные классы

- Инкапсуляция
- Наследование
 - Структурирование
 - Декомпозиция
- Полиморфизм

- ***Iris*** - каркас для построения редакторов

- Манипуляторы:
 - проектирование
 - реализация

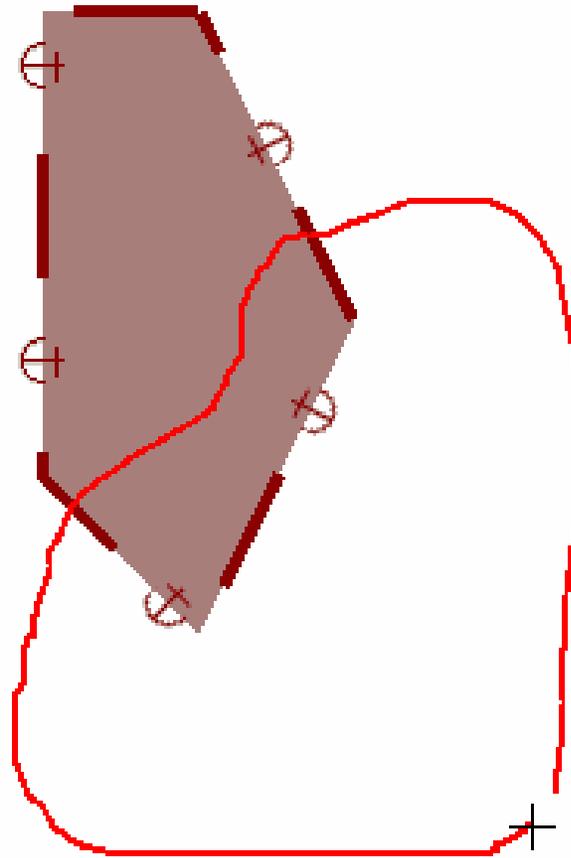
- **Механизмы редактирования**
 - выделение объектов
 - перенос объектов
 - вращение объектов
 - и т.п.

- Механизм редактирования:
 - инструмент
 - ***манипулятор***
 - метка
 - призрак
 - команда

- Манипуляторы выделения
 - выделение кликом
 - выделение прямоугольником
 - **выделение полигоном**
 - **выделение «лассо»**

Практическое использование: манипуляторы

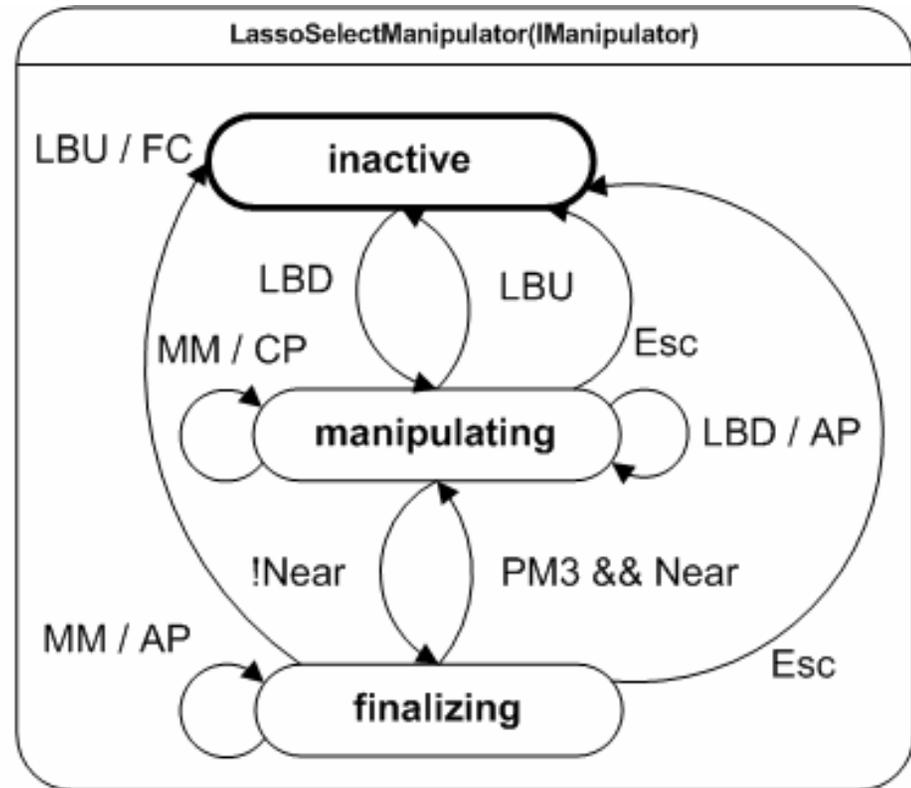
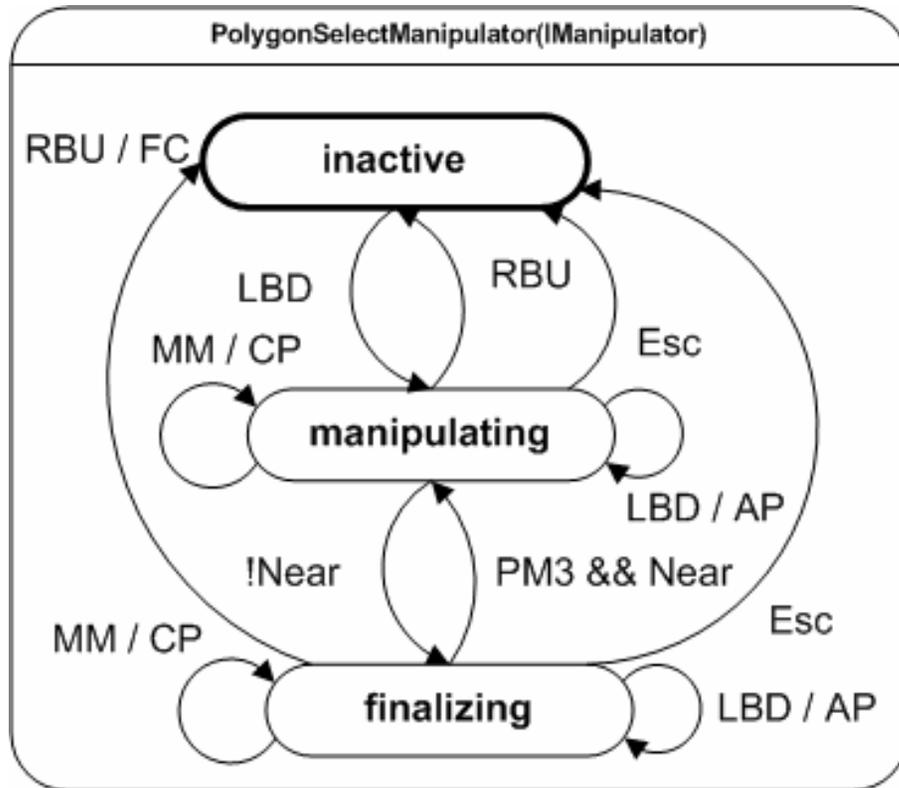
- Механизм выделения «лассо»



- Механизм выделения «лассо»
 - SelectObjectTool
 - LassoSelectionManipulator
 - ISelectable
 - PolygonSelectableGhost
 - SelectObjCommand

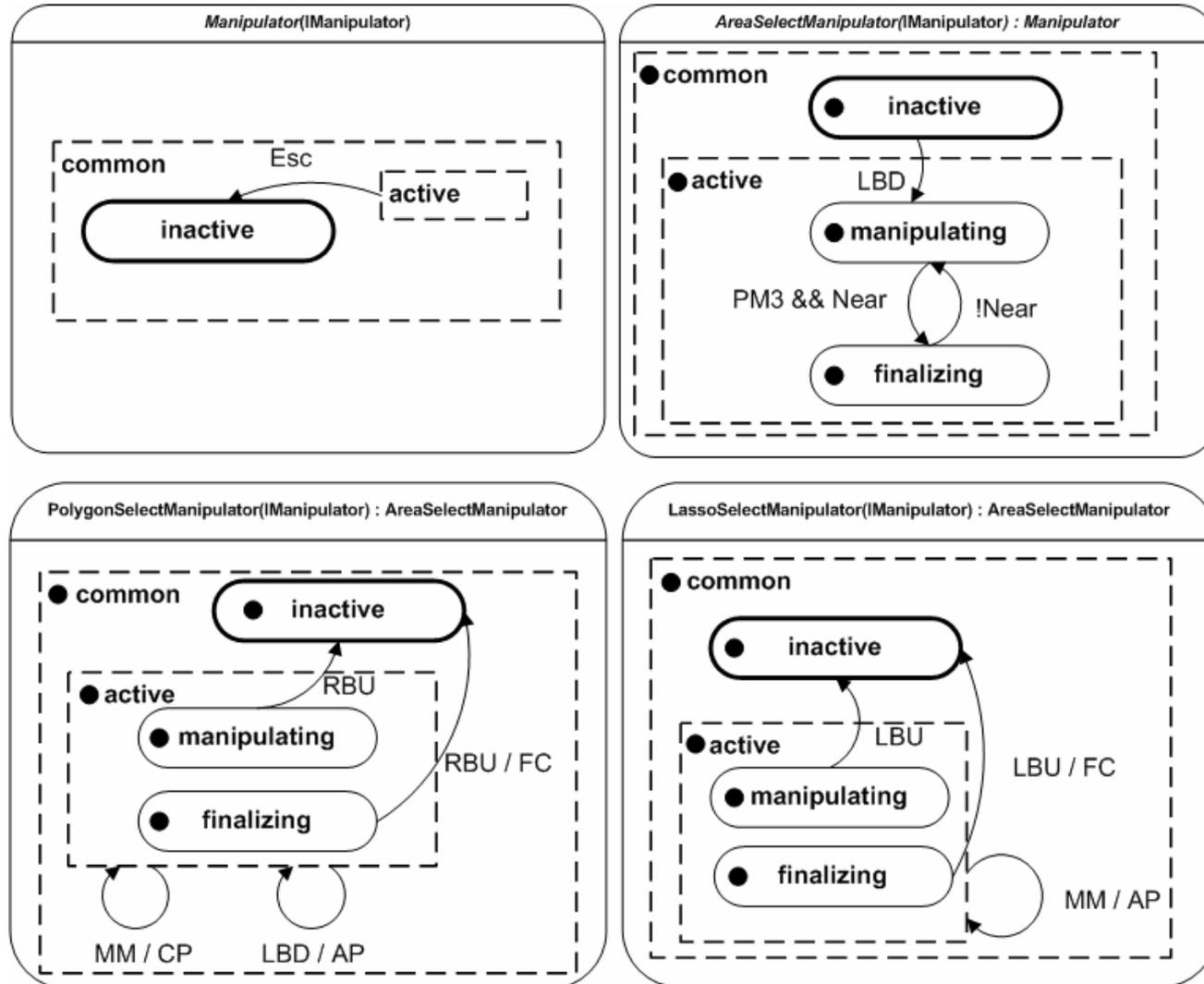
Практическое использование: манипуляторы

- Без использования наследования



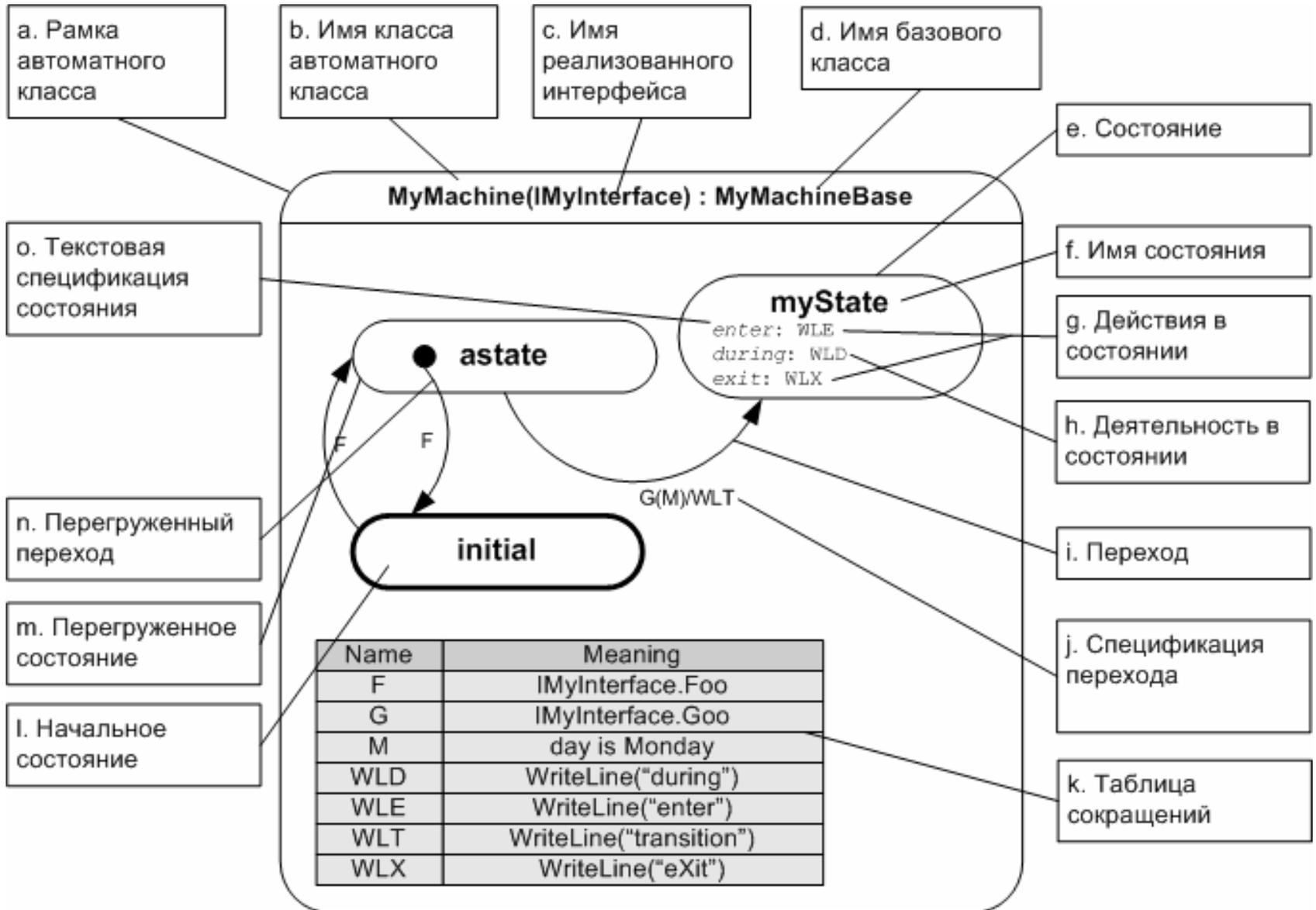
Практическое использование: манипуляторы

■ С использованием наследования



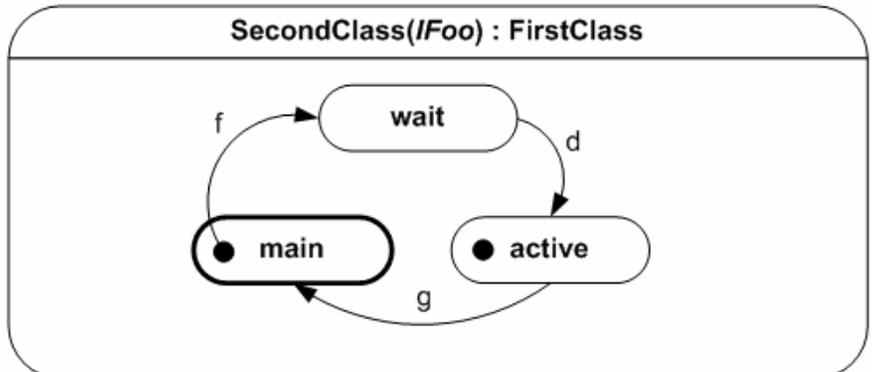
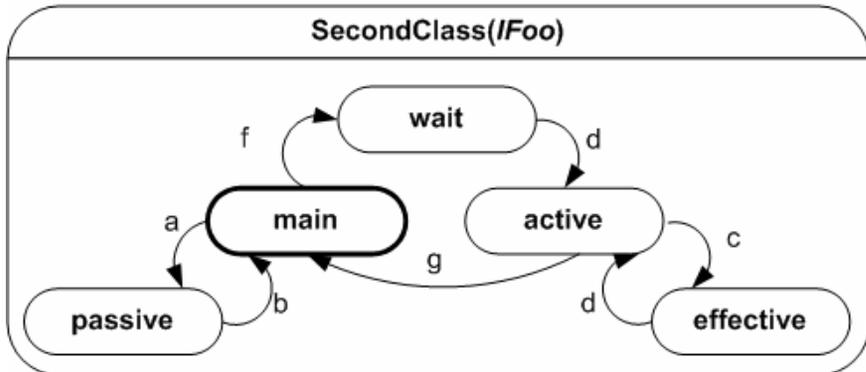
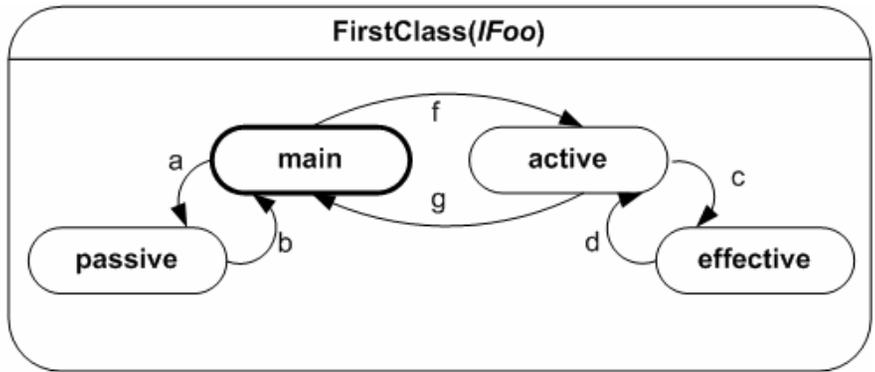
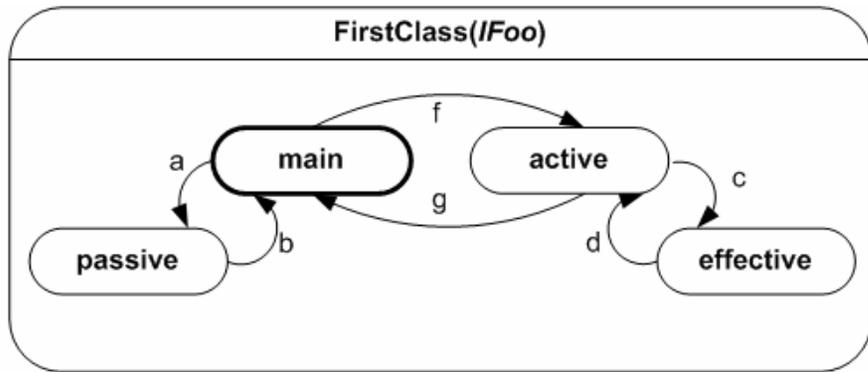
Графическая нотация

■ Основные элементы предлагаемой графической нотации



Графическая нотация: наследование

- добавление переходов
- перегрузка переходов
- добавление состояний

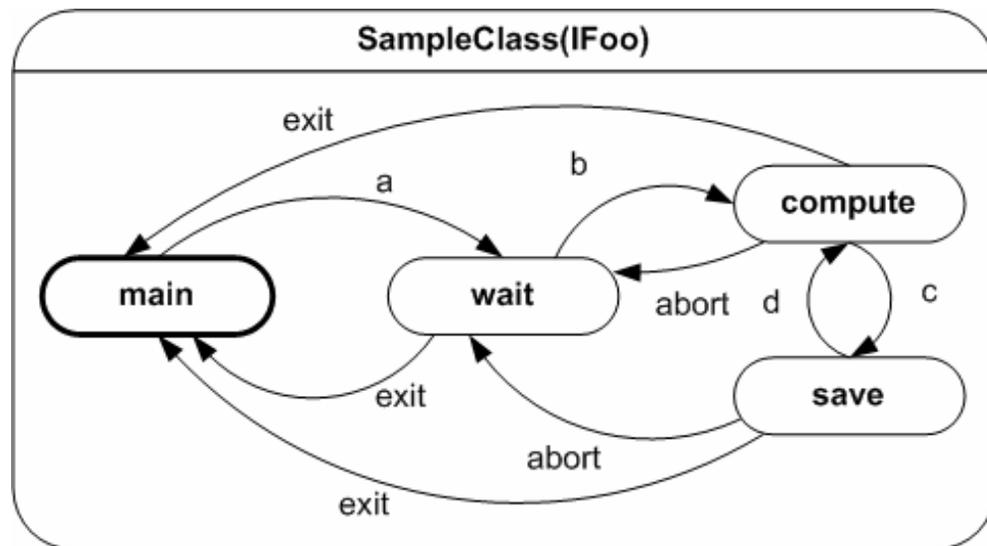


а) без использования наследования

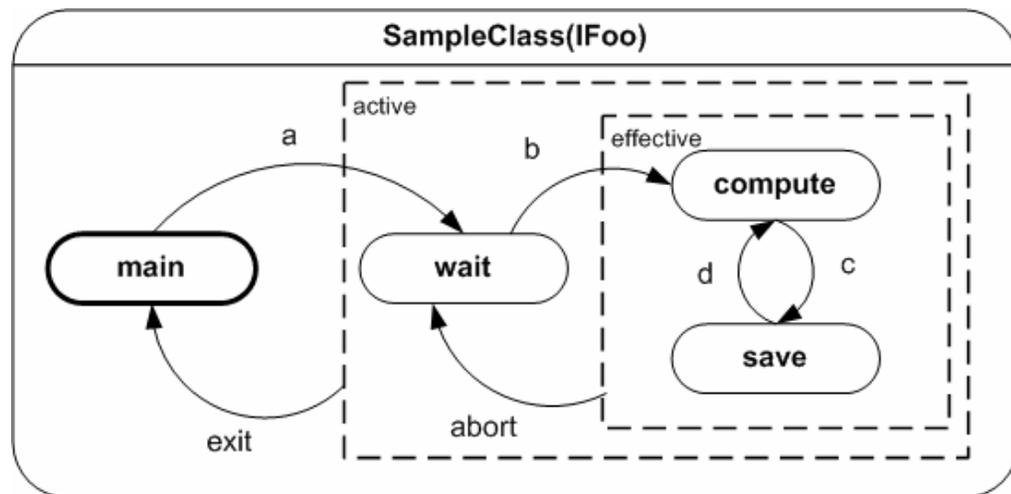
б) с использованием наследования

Графическая нотация: структурирование

- Без использования структурирования

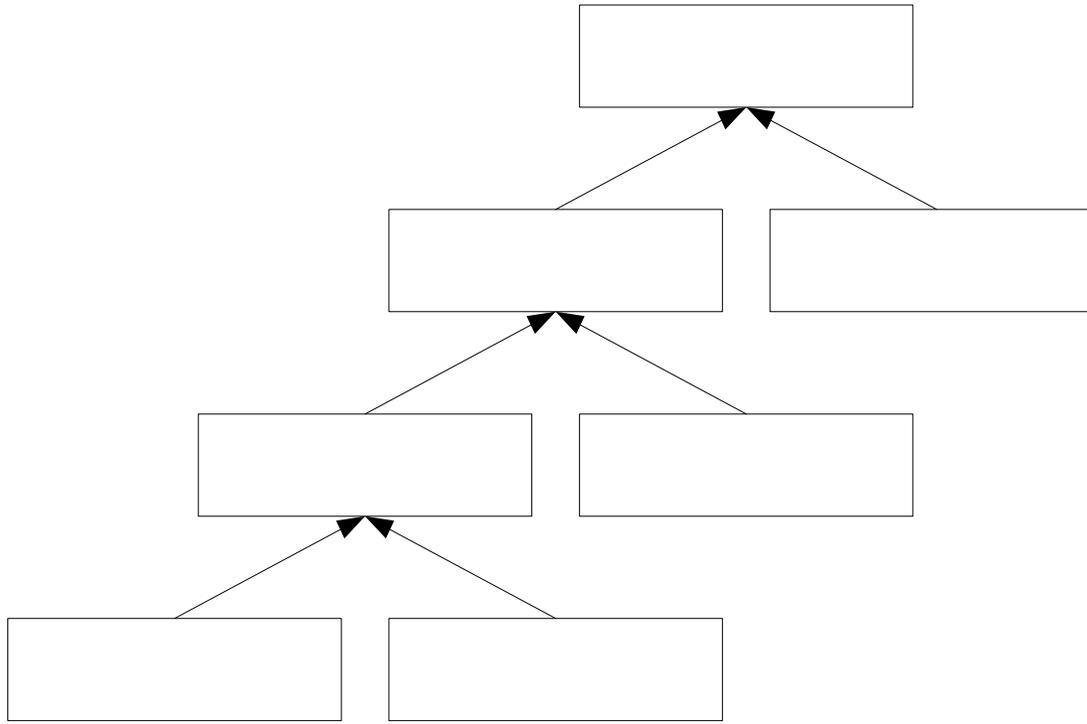
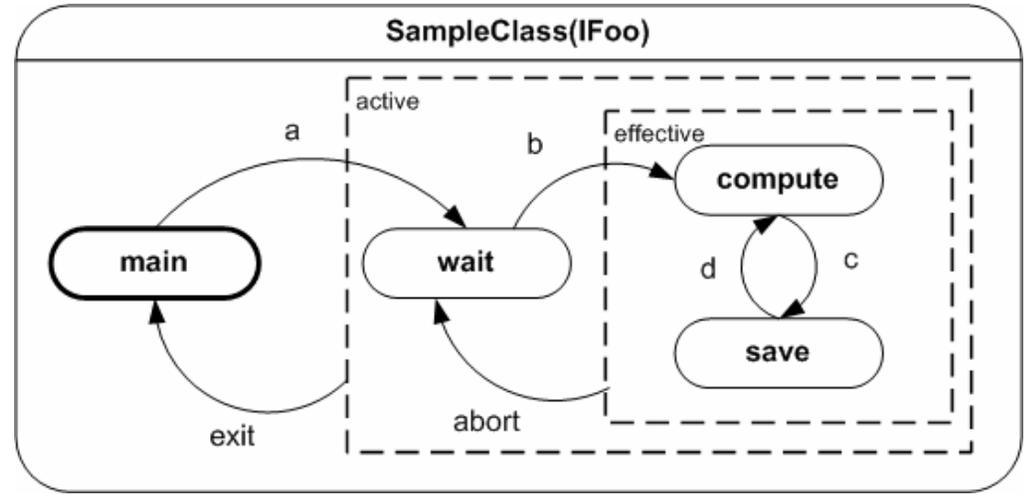


- С использованием структурирования

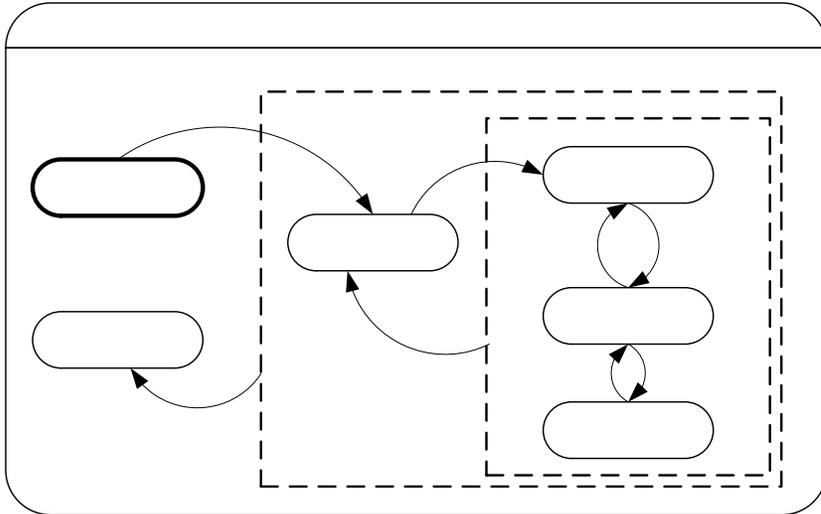
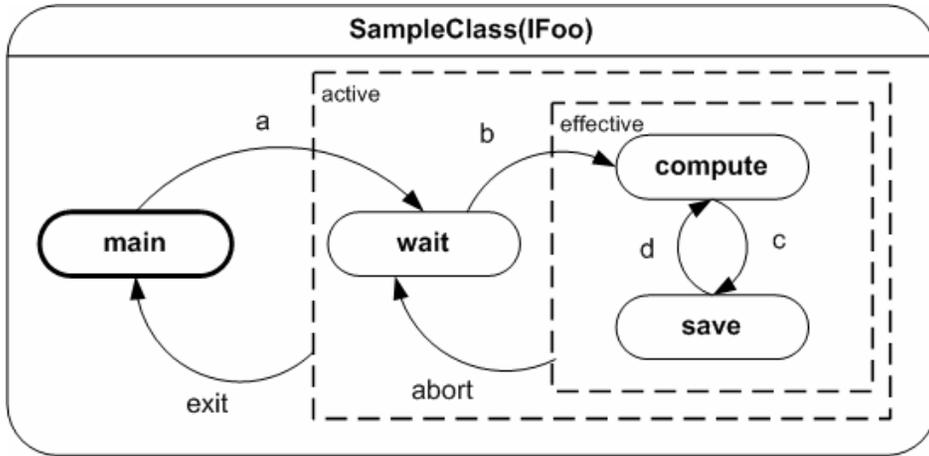


Графическая нотация: структурирование

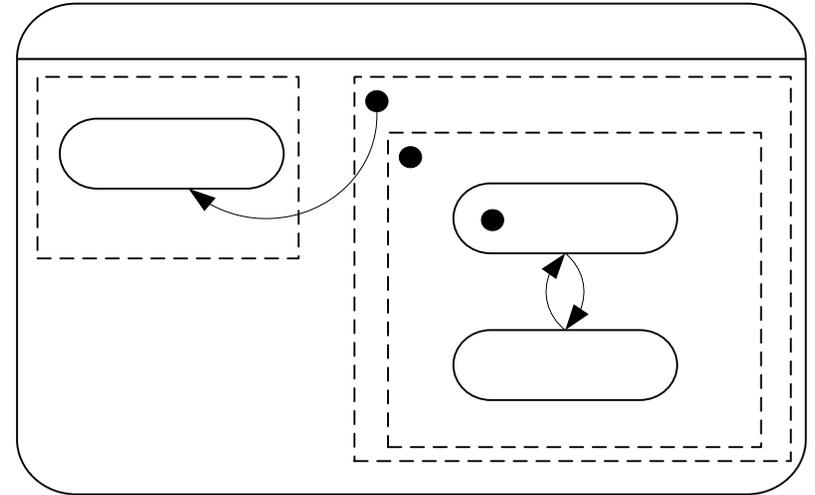
- Иерархия состояний, возникающая при использовании структурирования



Графическая нотация: наследование + структурирование



а) без совместного использования



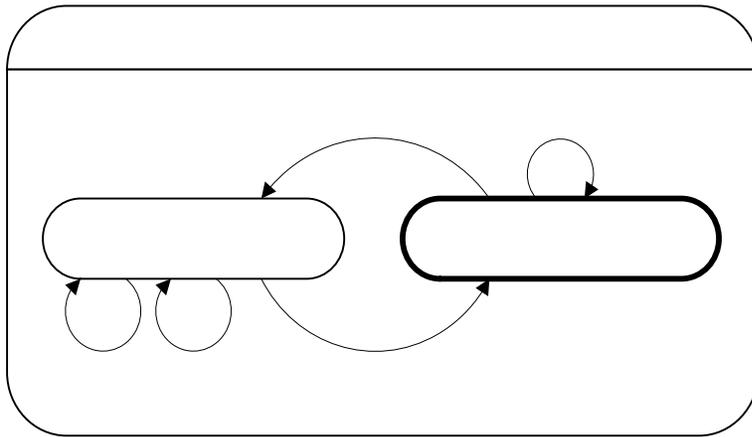
б) с совместным использованием

Графическая нотация: пример

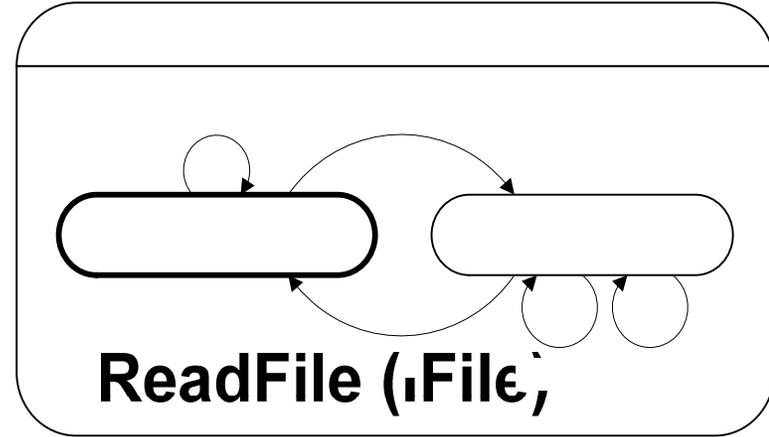
- Доступ к файлу:
 - доступ на чтение (`ReadFile`);
 - доступ на запись (`WriteFile`);
 - доступ на чтение, запись и чтение/запись (`ReadWriteFile`).

Графическая нотация: пример

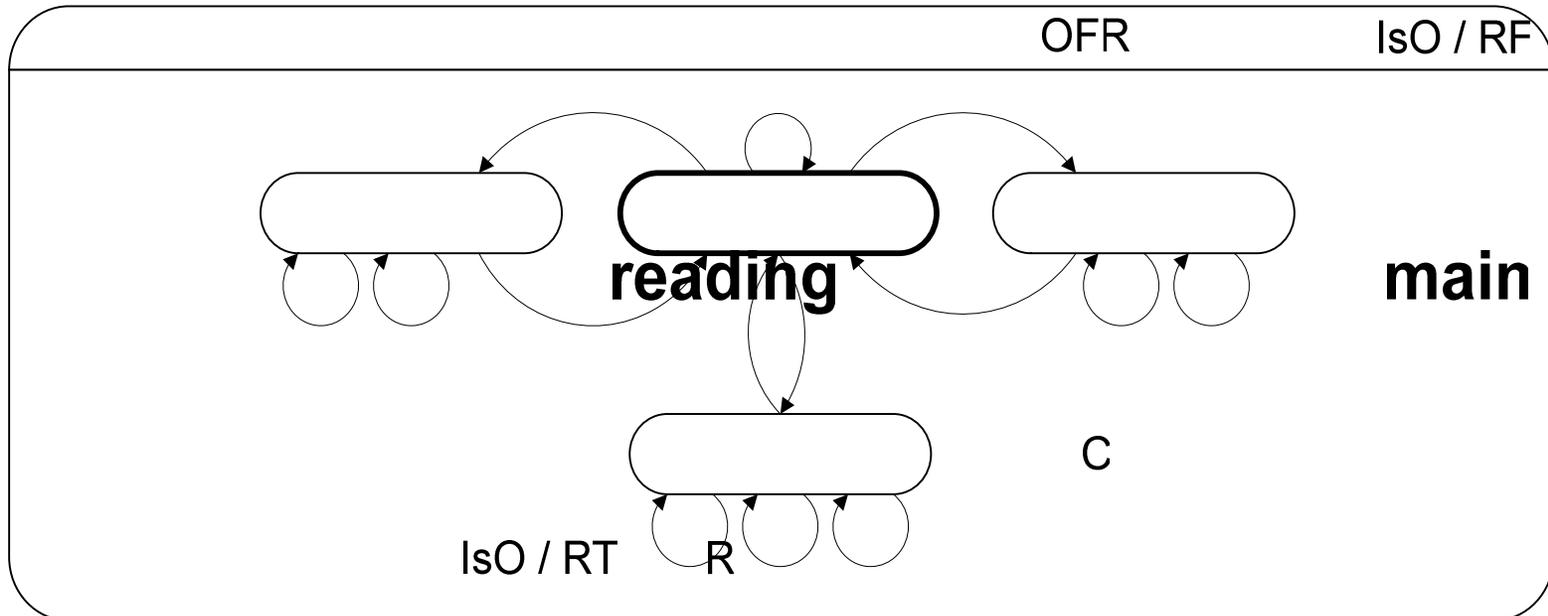
■ Диаграммы поведения без наследования



а) Доступ на чтение



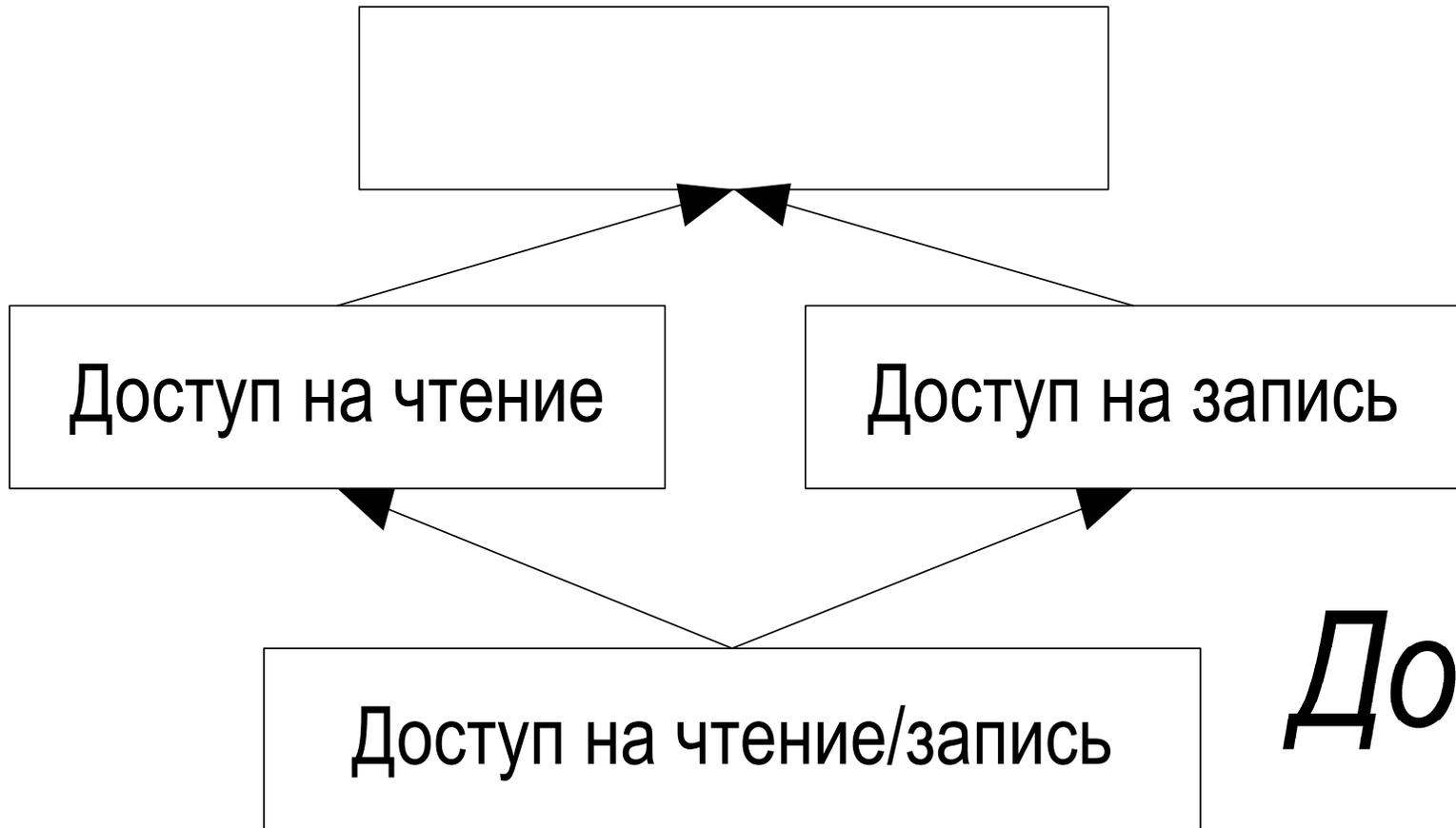
б) Доступ на запись



в) Доступ на чтение/запись

Графическая нотация: пример

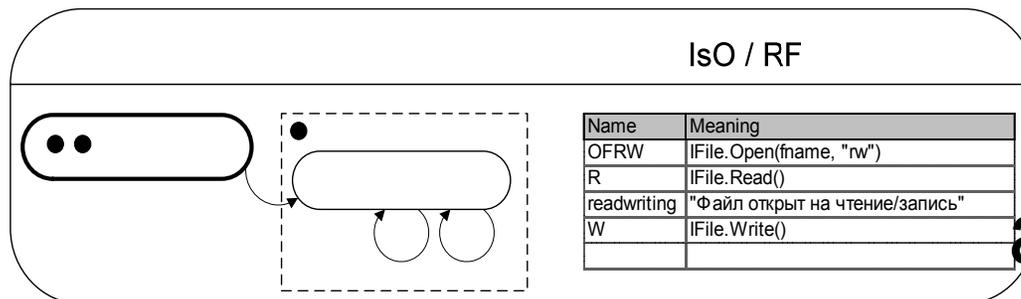
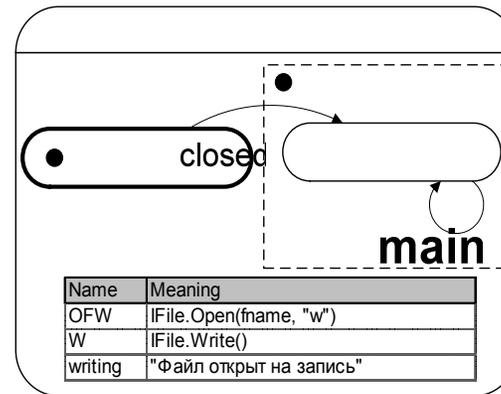
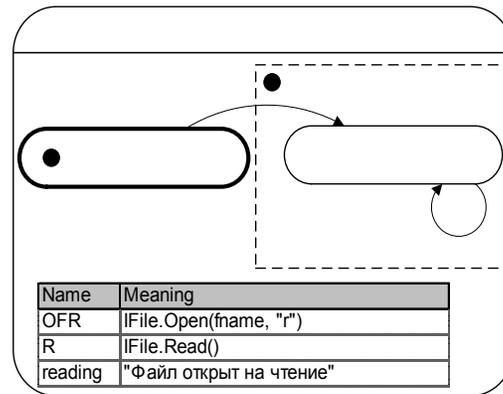
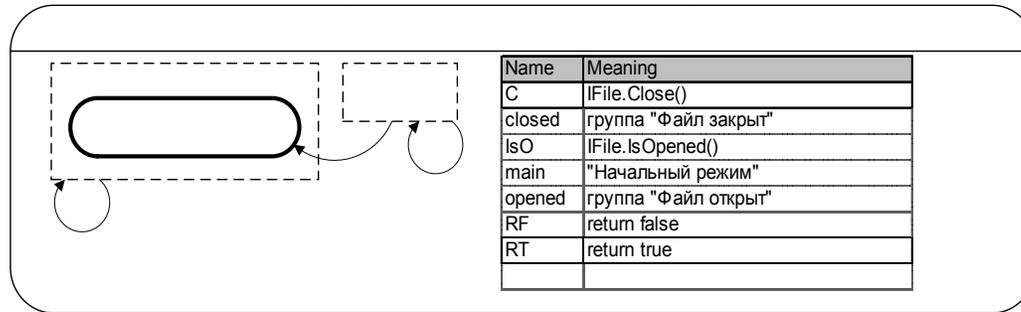
- Автоматные классы с предыдущего слайда можно организовать в следующую иерархию:



Дост

Графическая нотация: пример

Используя подход, предложенный в работе, диаграммы поведения могут быть существенно упрощены:



opened

C

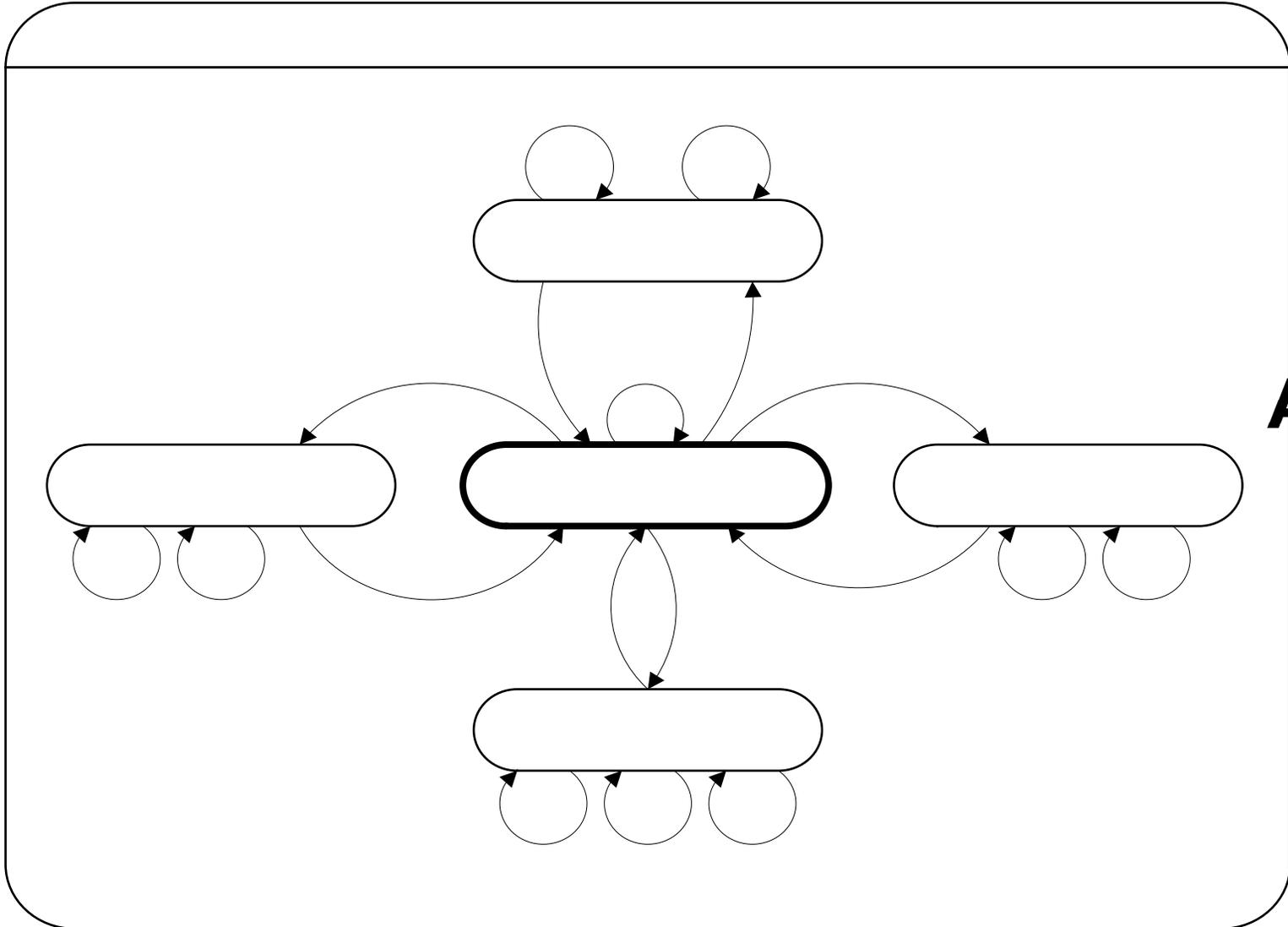
IsO / R

а) Абстрактный

ReadFile (IFile) : File

Графическая нотация: пример

- Предположим, что требуется добавить еще один автоматный класс:



Appen

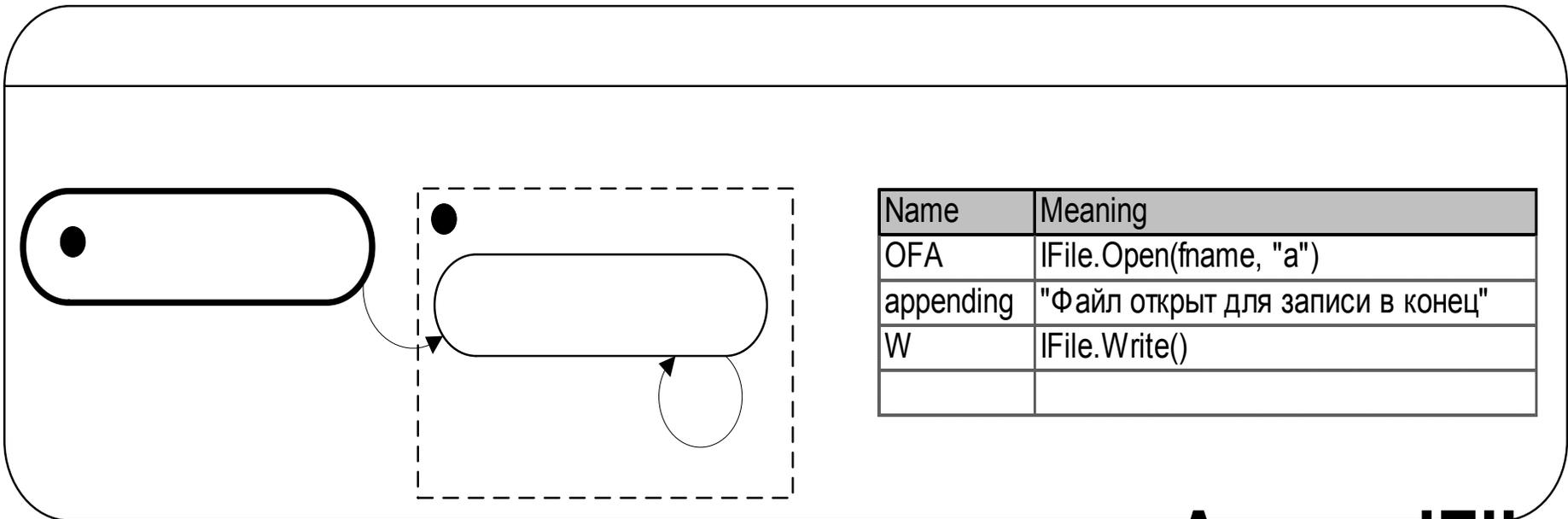
ISO /

ap

22

Графическая нотация: пример

- Используя подход, предлагаемый в работе, диаграмма может существенно упрощена:



AppendFile (

Графическая нотация

■ Количественные характеристики

	Без использования наследования и структурирования логики	С использованием наследования и структурирования логики
Количество состояний	13	5
Количество перегруженных состояний	–	4
Количество групп состояний	–	2
Количество перегруженных групп состояний	–	4
Количество переходов	42	12

Реализация автоматных объектов

- Методы на основе:
 - виртуальных методов;
 - виртуальных вложенных классов.
- Позволяют изоморфно отобразить предложенную выше графическую нотацию при реализации автоматных объектов.
- Соответствуют основным принципам ООП:
 - инкапсуляция;
 - наследование;
 - полиморфизм.

Заключение

■ Достоинства

- простота и интуитивность
- инкрементальное наследование

■ Недостатки

- дополнительные примитивы
- иерархии автоматных классов достаточно редки

Публикации

1. **Шопырин Д.Г., Шальто А.А.** Применение класса "STATE" в объектно-ориентированном программировании с явным выделением состояний //Труды X Всероссийской научно-методической конференции "Телематика-2003". СПб.: СПбГИТМО (ТУ). 2003. Т. 1, с. 284–285.
2. **Шальто А. А., Шопырин Д. Г.** Объектно-ориентированный подход к автоматному программированию //Информационно-управляющие системы. 2003, № 5, с. 29–39.
3. **Шальто А.А., Шопырин Д.Г.** Синхронное программирование //Информационно-управляющие системы. 2004, № 3, с. 35–42.
4. **Шопырин Д.Г.** Разработка промежуточного языка представления конечных автоматов //Труды XI Всероссийской научно-методической конференции "Телематика-2004". СПб.: СПбГИТМО (ТУ). 2004. Т. 1, с. 195–197.
5. **Шопырин Д.Г.** Метод проектирования и реализации конечных автоматов на основе виртуальных вложенных классов //Информационные технологии моделирования и управления. 2005, № 1(19), с. 87–97.
6. **Шопырин Д.Г.** Объектно-ориентированная реализация конечных автоматов на основе виртуальных методов // Информационно-управляющие системы. 2005, № 3, с. 36–40.
7. **Шопырин Д.Г.** Программирование с явным выделением состояний на платформе .Net //Труды XII Всероссийской научно-методической конференции "Телематика-2005". СПб.: СПбГИТМО (ТУ). 2005. Т. 1, с. 86–87.

Спасибо за внимание!