

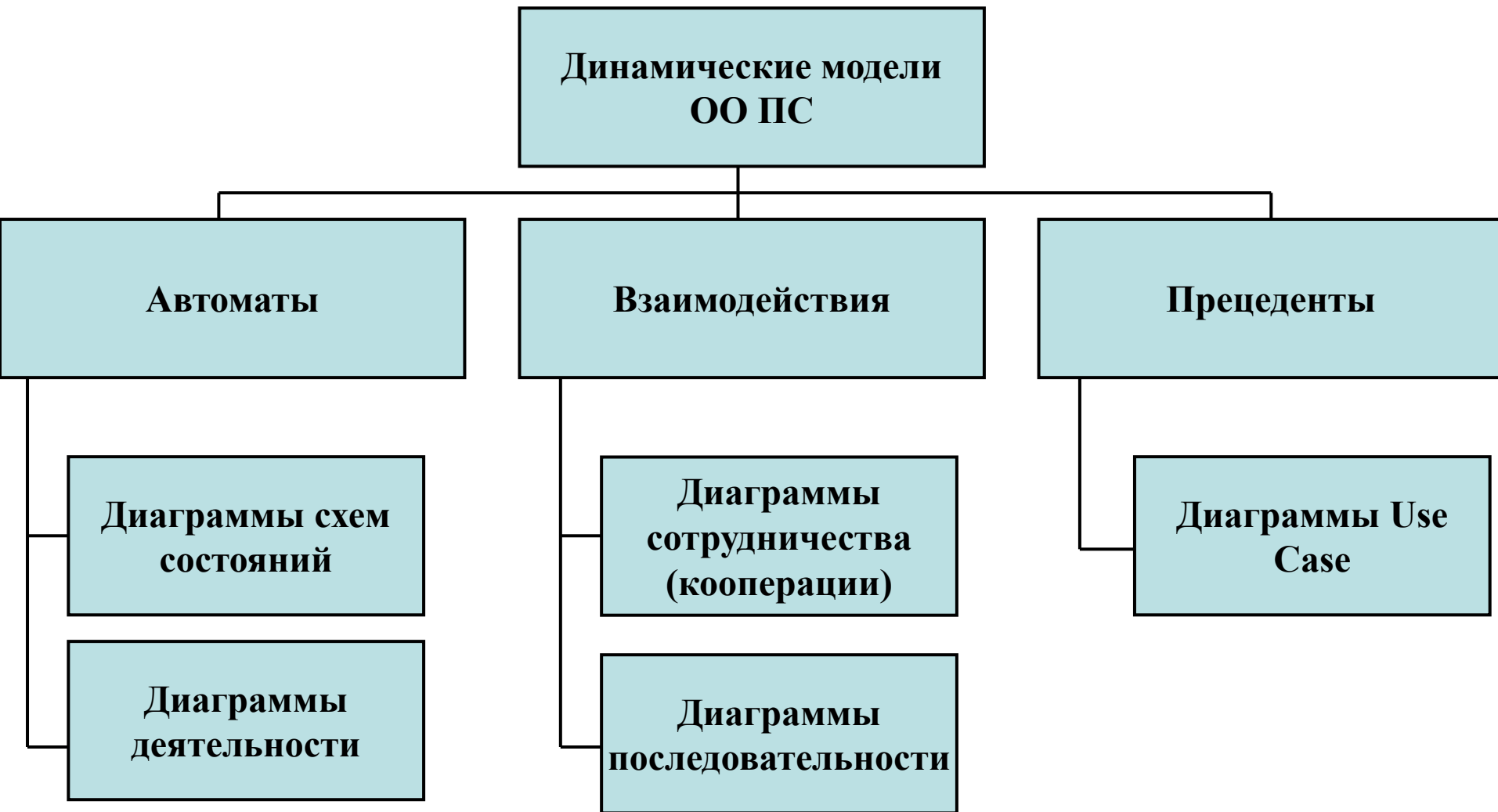
Лекция 6. Динамические модели объектно-ориентированных программных систем

Учебные вопросы:

1. Автоматы.
2. Диаграммы взаимодействия.
3. Диаграммы Use Case.

Литература: [4], [6], [10].

Классификация динамических моделей объектно-ориентированных ПС



Диаграммы схем состояний

Диаграмма схем состояний отображает *конечный автомат*, выделяя поток управления, следующий от состояния к состоянию.

Конечный автомат – это поведение, которое определяет последовательность состояний в ходе существования объекта. Эта последовательность рассматривается как ответ на события и включает реакции на эти события.

Диаграмма схем состояний показывает:

- набор состояний системы;
- события, которые вызывают переход из одного состояния в другое;
- действия, которые происходят в результате изменения состояния.

Элементы диаграммы схем состояний:

- состояния;
- переходы между состояниями.

Элементы диаграммы

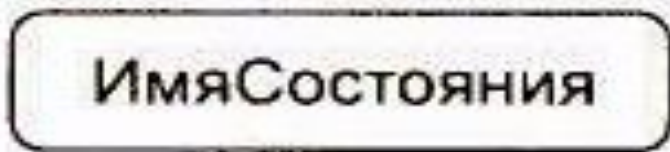


Рисунок 1.1 – Обозначение состояния

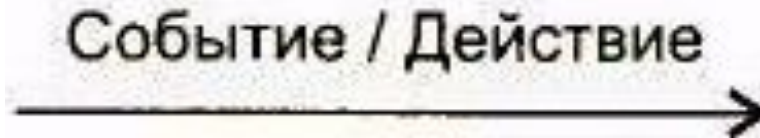


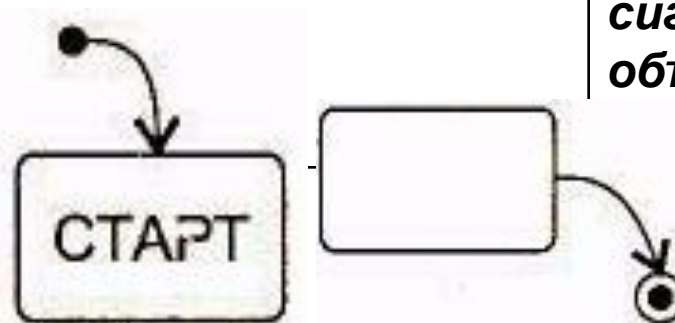
Рисунок 1.2 – Переходы между состояниями

Таблица 1.1 – Примеры событий

| | |
|---------------------|---|
| баланс < 0 | Изменение в состоянии |
| помехи | Сигнал (объект с именем) |
| уменьшить(Давление) | Вызов действия |
| after (5 seconds) | Истечение периода времени |
| when (time = 16:30) | Наступление абсолютного момента времени |

Таблица 1.2 – Примеры действий

| | |
|---|---|
| Кассир. прекратитьВыплаты () | <i>Вызов одной операции</i> |
| flt:= new(Фильтр); flt. убратьПомехи() | <i>Вызов двух операций</i> |
| send Ник. привет | <i>Посылка сигнала в объект</i> |



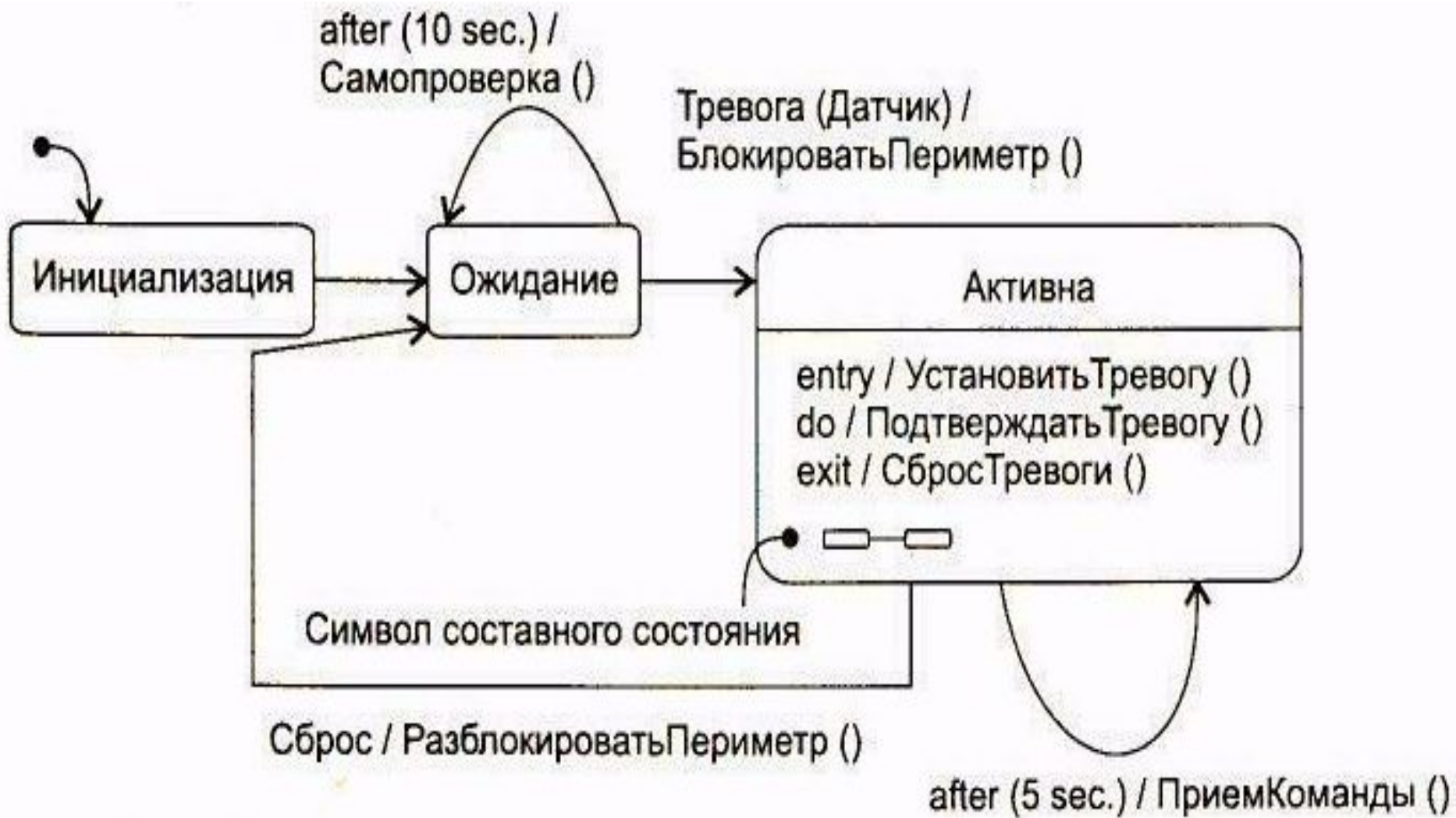


Рисунок 1.3 – Диаграмма схем состояний системы охранной сигнализации

Действия в состояниях

Для указания действий, выполняемых при входе в состояние и при выходе из состояния, используются метки **entry** и **exit** соответственно.

Действие, которое должно выполняться, когда система находится в данном состоянии, указывается после метки **do**.

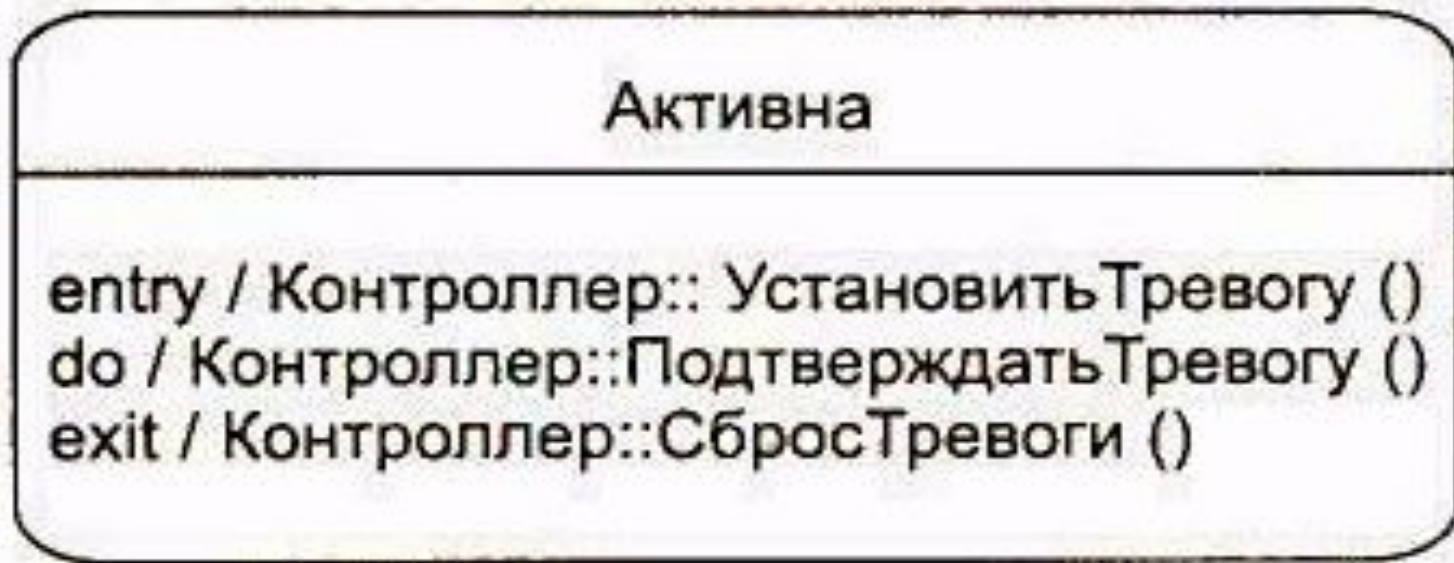


Рисунок 1.4 – Входные и выходные действия и деятельность в состоянии Активна

Условные переходы

Между состояниями возможны различные типы переходов:

- переход инициируемый событием;
- переходы без событий;
- условные или охраняемые переходы.

Событие [Условие] / Действие



Рисунок 1.5 – Обозначение условного перехода

Порядок выполнения условного перехода:

- происходит событие;
- вычисляется условие [УсловиеПерехода];
- при [УсловиеПерехода] = true запускается переход и активизируется действие, в противном случае переход не выполняется.

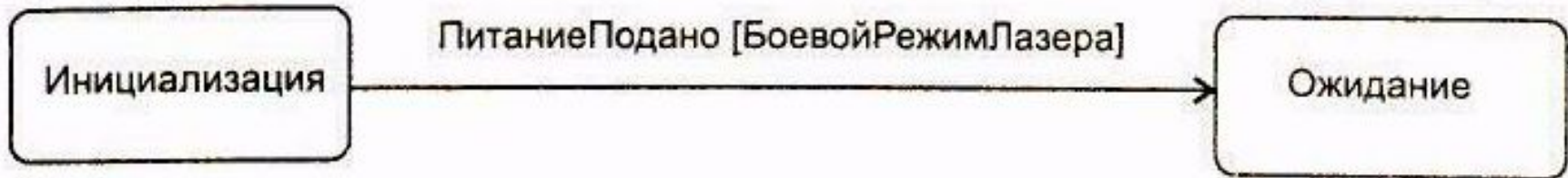


Рисунок 1.6 – Условный переход между состояниями

Вложенные состояния



Рисунок 1.7 – Обозначение подсостояний

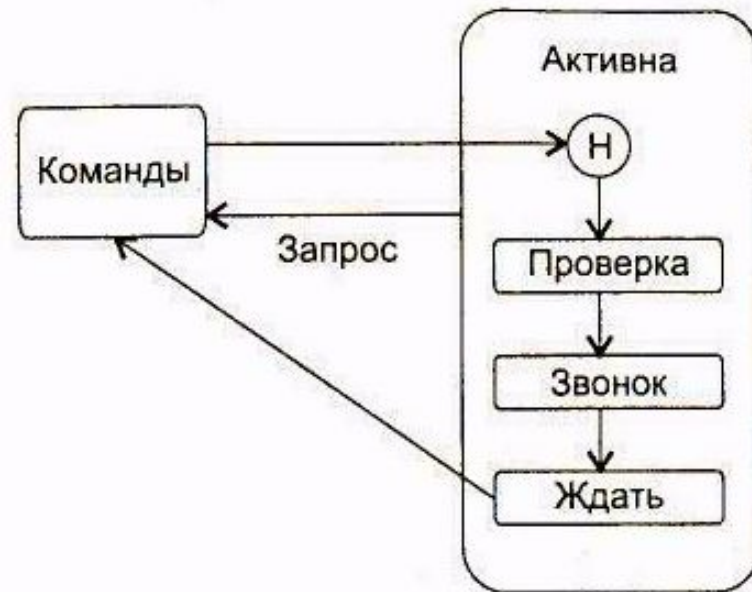


Рисунок 1.9 – Историческое состояние



Рисунок 1.8 – Переходы в состоянии Активна



Рисунок 1.10 – Символ состояния со скрытыми подсостояниями

Диаграммы деятельности

Диаграмма деятельности представляет особую форму конечного автомата, в которой показываются процесс вычислений и потоки работ. В ней выделяются не обычные состояния объекта, а состояния выполняемых вычислений – состояния действий. При этом полагается, что процесс вычислений не прерывается внешними событиями.

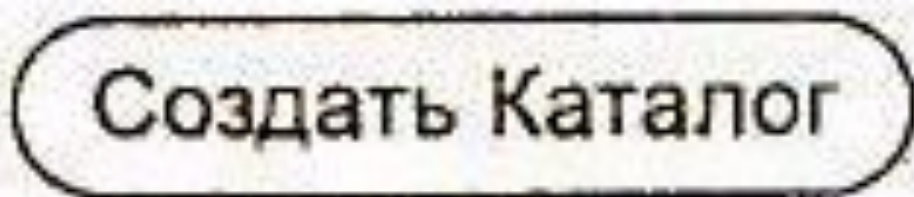


Рисунок 1.11 – Состояние действия

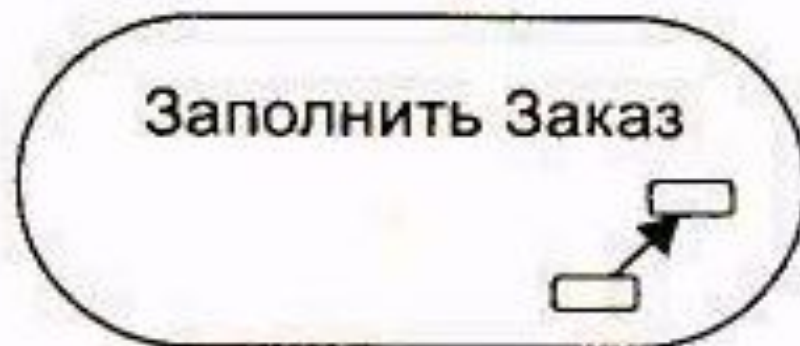


Рисунок 1.12 – Состояние под-деятельности

Переходы и вспомогательные вершины

Переходы между вершинами – состояниями действий – изображаются в виде стрелок.

В диаграммах деятельности используются *вспомогательные вершины*:

- *решение* (ромбик с одной входящей и несколькими исходящими стрелками);
- *объединение* (ромбик с несколькими входящими и одной исходящей стрелкой);
- *линейка синхронизации – разделение* (жирная горизонтальная линия с одной входящей и несколькими исходящими стрелками);
- *линейка синхронизации – слияние* (жирная горизонтальная линия с несколькими входящими и одной исходящей стрелкой);
- *начальное состояние* (черный кружок);
- *конечное состояние* (не закрашенный кружок, в котором размещен черный кружок меньшего размера).

Плавательная дорожка – это определенная область, которая имеет имя и фиксирует область деятельности конкретного лица, обозначая зону его ответственности.

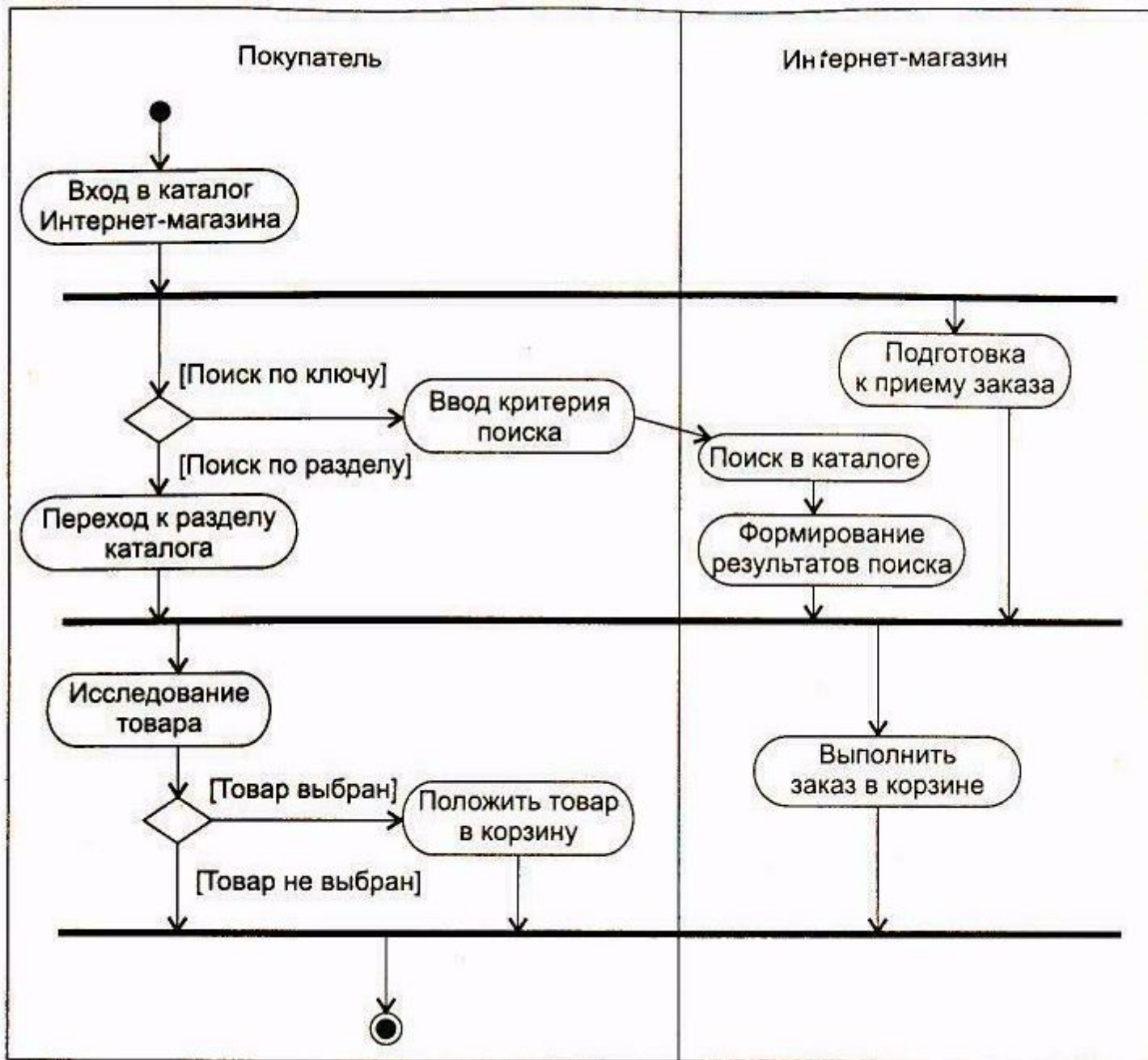


Рисунок 1.13 – Диаграмма деятельности покупателя в Интернет-магазине

Диаграммы сотрудничества

Диаграмма сотрудничества – это диаграмма взаимодействия, которая выделяет структурную организацию объектов, посылающих и принимающих сообщения.

Элементами диаграмм взаимодействия являются участники взаимодействия – **объекты, связи, сообщения.**

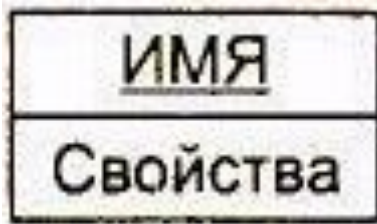


Рисунок 2.1 – Обозначение объекта

Имя : Тип = Значение

| | |
|-------------------------------|-----------------------|
| номер:Телефон = "7350-420" | Имя, тип, значение |
| активен = True | Имя и значение |

ИмяОбъекта : ИмяКласса

| | |
|---|---|
| <u>Адам :</u> <u>Человек</u> | Имя объекта и класса |
| <u>:</u> <u>Пользовате</u> <u>ль</u> | Только имя класса (анонимный объект) |
| <u>мойКомпьюте</u> <u>р</u> | Только имя объекта (подразумевается, что имя класса известно) |
| <u>агент :</u> | Объект – сирота (подразумевается, что имя класса неизвестно) |

Взаимодействие объектов

Стандартные стереотипы видимости

| | |
|--------------------|--|
| «global» | Объект-поставщик находится в глобальной области определения |
| «local» | Объект-поставщик находится в локальной области определения объекта-клиента |
| «parameter» | Объект-поставщик является параметром операции объекта-клиента |
| «self» | Один и тот же объект является и клиентом, и поставщиком |

Моделируемые разновидности действий

| | |
|-----------------------|--|
| Вызов | В объекте запускается операция |
| Возврат | Возврат значения в вызывающий объект |
| Посылка (Send) | В объект посылается сигнал |
| Создание | Создание объекта, выполняется по стандартному сообщению «create» |
| Уничтожение | Уничтожение объекта, выполняется по стандартному сообщению «destroy» |

Взаимодействие объектов

Для записи сообщений в языке UML принят следующий синтаксис:

ВозврВеличина := ИмяСообщения (Аргументы)

Примеры записи сообщений:

| | |
|--|--|
| Коорд := ТекущПоложение (самолетТ1) | Вызов операции, возврат значения |
| оповещение() | Посылка сигнала |
| УстановитьМаршрут(x) | Вызов операции с действительным параметром |
| «create» | Стандартное сообщение для создания объекта |



Рисунок 2.2 – Поток синхронных сообщений

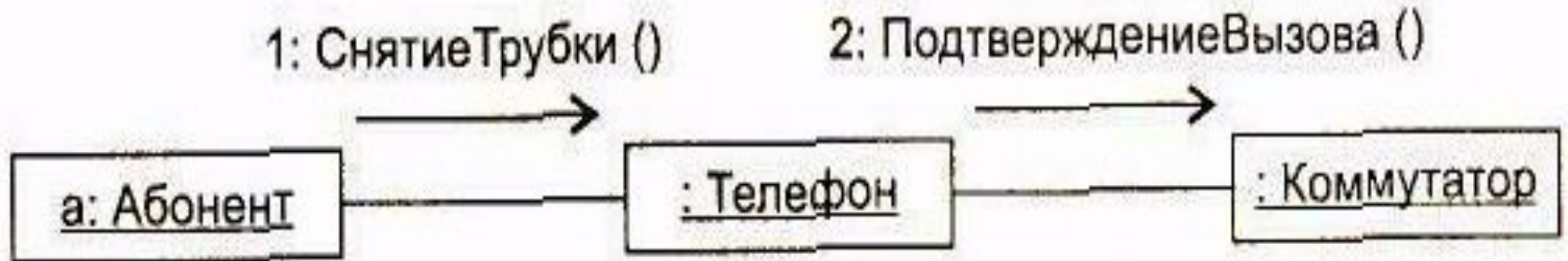


Рисунок 2.3 – Поток асинхронных сообщений

1*[i: = 1..4]: РисоватьСторонуПрямоугольника(i)

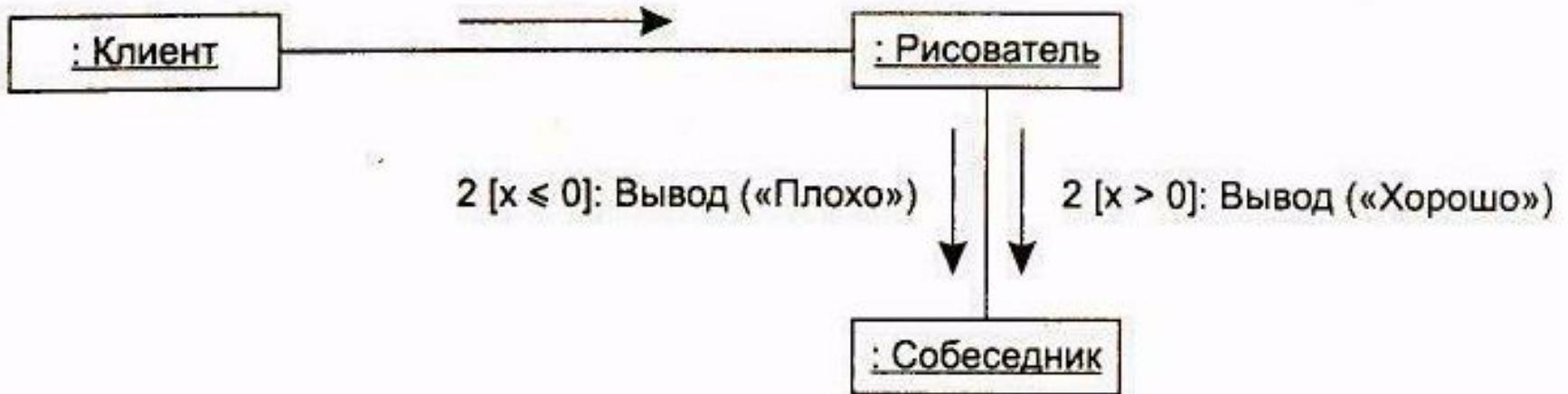


Рисунок 2.4 – Итерационный и разветвляющийся поток сообщений

Алгоритм формирования диаграммы сотрудничества:

- отображаются объекты, которые участвуют во взаимодействии;
- рисуются связи, соединяющие эти объекты;
- связи помечаются сообщениями, которые посылают и получают выделенные объекты.

Взаимодействие объектов

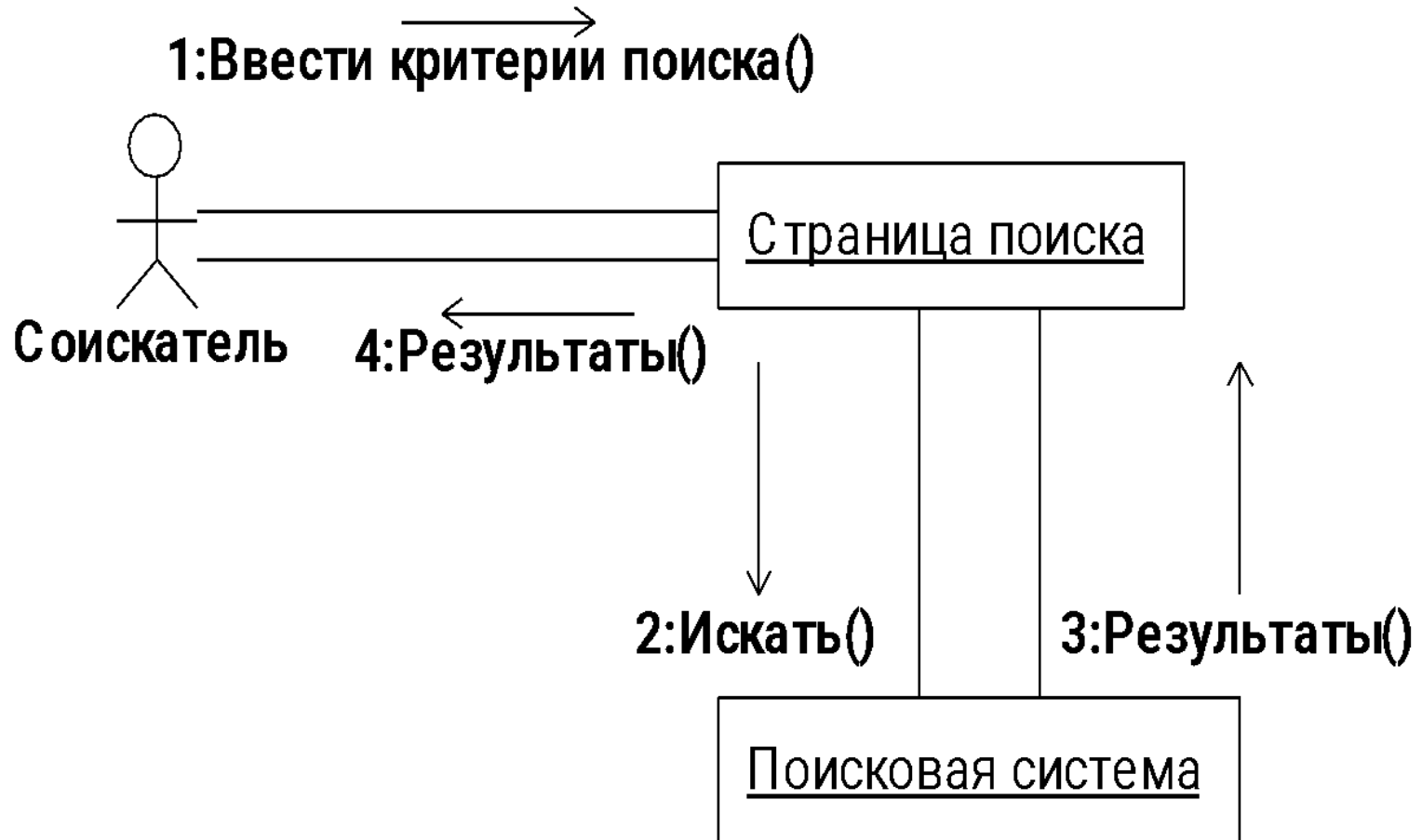
Стандартные стереотипы видимости

| | |
|--------------------|--|
| «global» | Объект-поставщик находится в глобальной области определения |
| «local» | Объект-поставщик находится в локальной области определения объекта-клиента |
| «parameter» | Объект-поставщик является параметром операции объекта-клиента |
| «self» | Один и тот же объект является и клиентом, и поставщиком |

Моделируемые разновидности действий

| | |
|-----------------------|--|
| Вызов | В объекте запускается операция |
| Возврат | Возврат значения в вызывающий объект |
| Посылка (Send) | В объект посылается сигнал |
| Создание | Создание объекта, выполняется по стандартному сообщению «create» |
| Уничтожение | Уничтожение объекта, выполняется по стандартному сообщению «destroy» |

Диаграммы сотрудничества



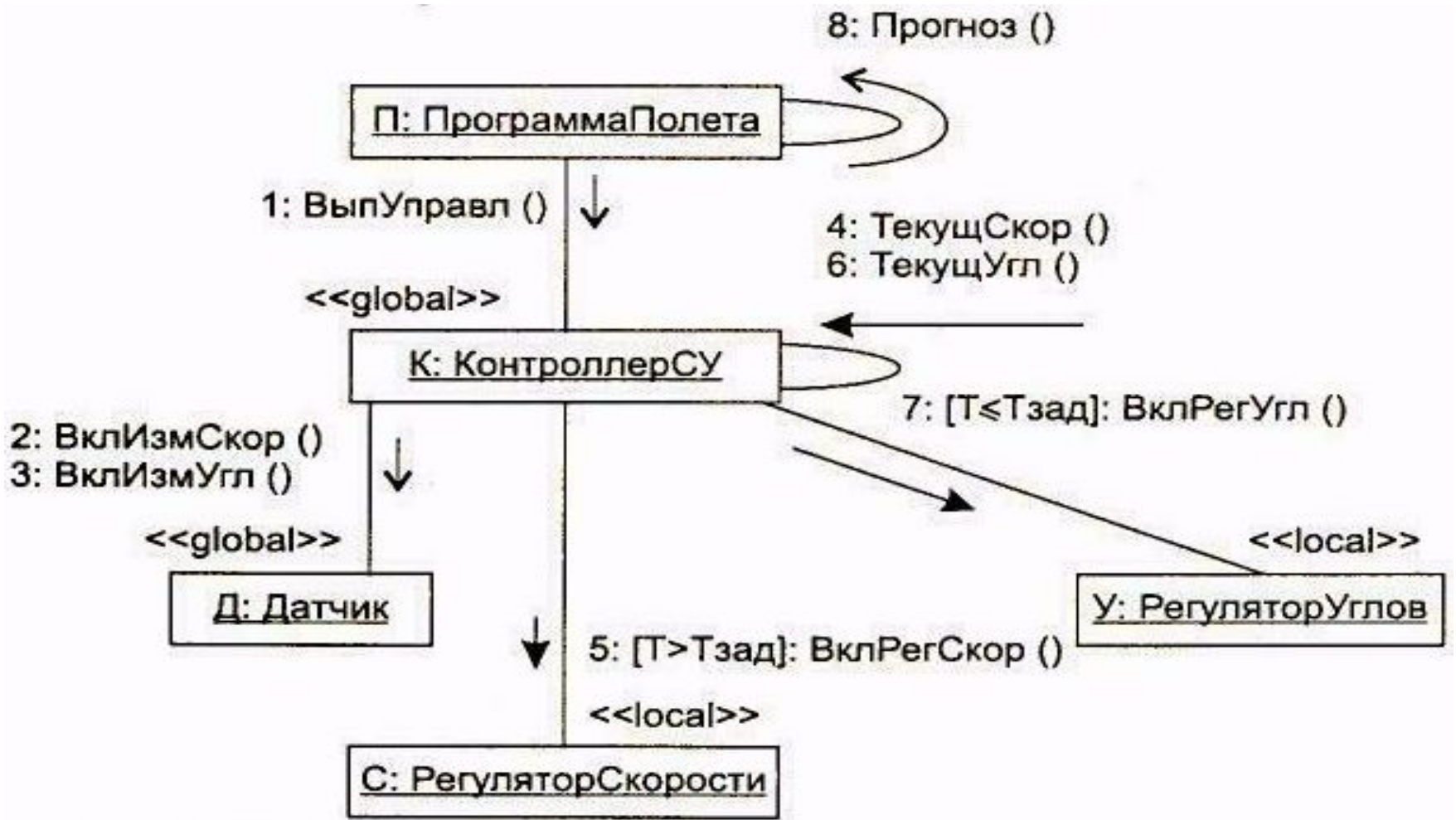


Диаграмма сотрудничества системы управления полетом

Диаграммы последовательности

Диаграмма последовательности, отражая сценарий поведения в системе, обеспечивает более наглядное представление порядка передачи сообщений по времени.

Фокус управления – это высокий тонкий прямоугольник, отображающий период времени, в течение которого объект выполняет действие (свою или подчиненную процедуру).

Линия жизни объекта – это вертикальная пунктирная линия, которая обозначает период существования объекта.



Диаграммы последовательности

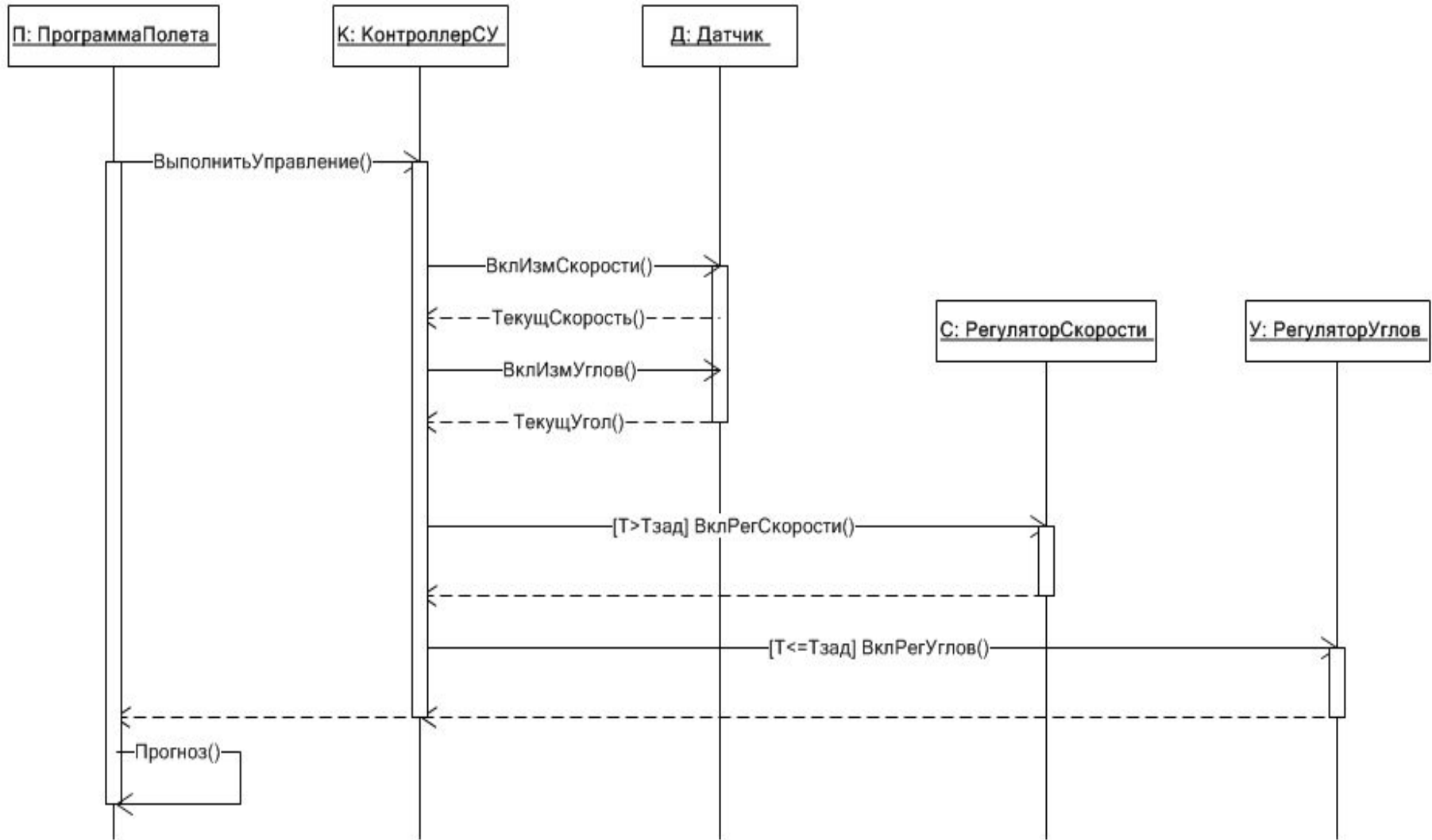
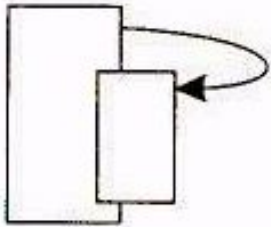


Рисунок 2.5 – Диаграмма последовательности системы управления полетом

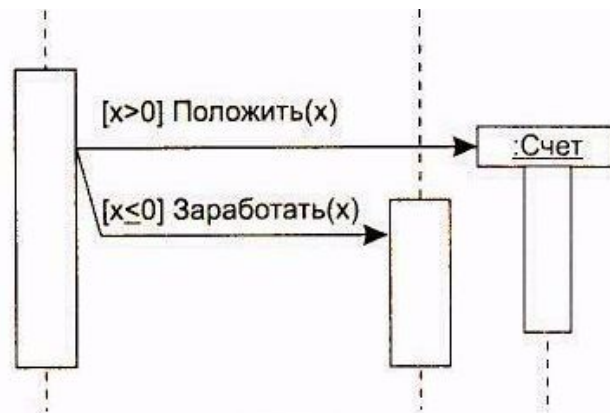
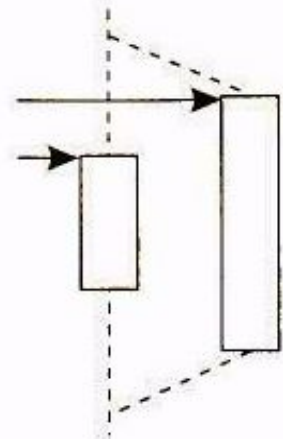
Фокус управления

Фокус управления – это высокий тонкий прямоугольник, отображающий период времени, в течение которого объект выполняет действие (свою или подчиненную процедуру). Вершина прямоугольника отмечает начало действия, а основание – его завершение. Момент завершения может маркироваться сообщением возврата, которое показывается пунктирной стрелкой. Можно показать вложение фокуса управления (например, рекурсивный вызов собственной операции).



**Рисунок 2.6 –
Вложение
фокусов
управления**

Для отображения «условности» линия жизни может быть разделена на несколько параллельных линий жизни. Каждая отдельная линия соответствует условному ветвлению во взаимодействии (рис. 2.7)



Ветвление показывается множеством стрелок, идущих из одной точки. Каждая стрелка отмечается сторожевым условием (рис. 2.8)

Диаграммы USE CASE

Диаграмма Use Case определяет поведение системы с точки зрения пользователя. Она рассматривается как главное средство для первичного моделирования динамики системы, используется для выяснения требований к разрабатываемой системе, фиксации этих требований в форме, которая позволит проводить дальнейшую разработку. Диаграммы Use Case часто называют диаграммами прецедентов или вариантов использования.

В состав диаграмм Use Case входят элементы Use Case, актеры, отношения зависимости, обобщения и ассоциации, примечания и ограничения и пакеты.

Актер – это роль объекта вне системы, который прямо взаимодействует с ее частью – конкретным элементом (элементом Use Case).

Элемент Use Case – это описание последовательности действий (или нескольких последовательностей), которые выполняются системой и производят для отдельного актера видимый результат.

Отношения в диаграммах Use Case

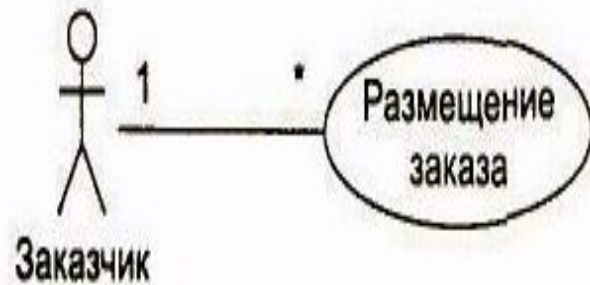


Рисунок 2.9– Отношение ассоциации между актером и элементом Use Case



Рисунок 2.10– Отношение обобщения между актерами

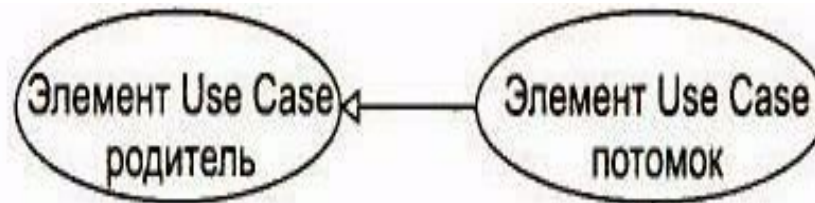


Рисунок 2.11– Отношение обобщения между элементами Use Case

Диаграммы USE CASE



Рисунок 2.12 – Пример диаграммы Use Case