

Тема. Диаграмма состояний

Понятия автомата, состояния, перехода

Автомат (state machine):

- описывает поведение объекта в терминах последовательности состояний, через которые проходит объект в течение своей жизни, отвечая на события, а также его реакций на эти события.

Пример автомата

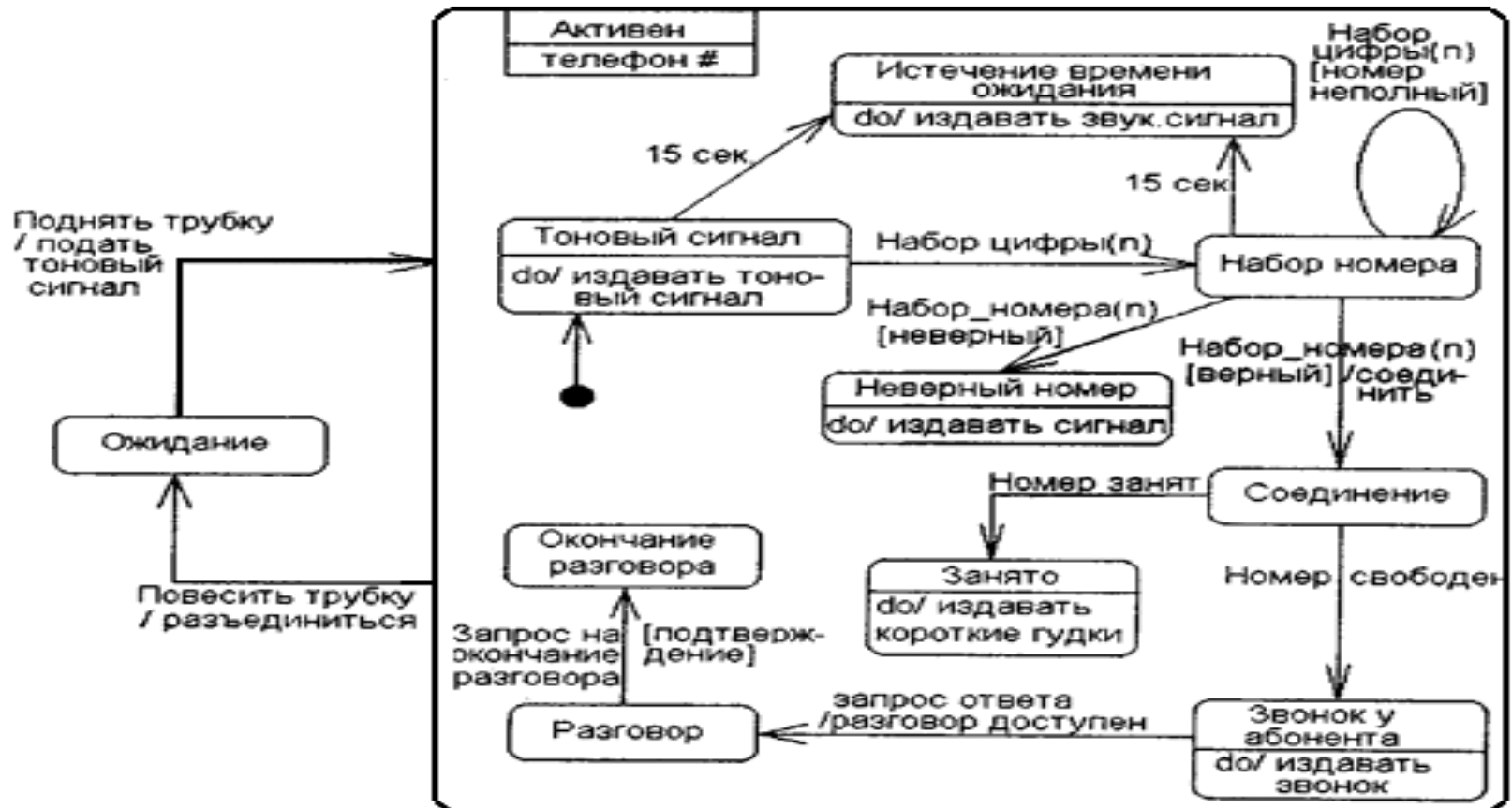
Простейший пример автомата - ситуация с исправностью некоторого технического устройства: два самых общих состояния: "исправен" и "неисправен" и два перехода: "выход из строя" и "ремонт".



Обязательные условия:

1. «А» не запоминает историю перемещения из состояния в состояние.
2. В каждый момент времени «А» может находиться в одном и только в одном из своих состояний.
3. Концепция времени не входит в формализм «А».
4. Кол-во состояний «А» должно быть конечным, и все они должны быть явно специфицированы.
5. «А» не должен содержать изолированных состояний и переходов.
6. «А» не должен содержать конфликтующих переходов .

Пример моделирования автомата



Основные элементы диаграммы состояний

1. **Состояние** (State) объекта - ситуация в его жизни, на протяжении которой он удовлетворяет некоторому условию, осуществляет определенную деятельность или ожидает какого-то события.
2. **Событие** (Event) - существенный факт, некий стимул, инициирующий переход из одного состояния в другое.
3. **Переход** (Transition) – отношения между двумя состояниями, показывающее, что объект должен перейти из одного состояния в другое.
4. **Деятельность** (Activity) - занимающее некоторое время неатомарное вычисление внутри автомата.
5. Результатом деятельности является некоторое **действие** (Action), составленное из атомарных вычислений, которое приводит к изменению состояния модели или возврату значения.

Состояние

- Элементы состояния:

1. имя
2. действия при входе/выходе
3. внутренние переходы
4. подсостояния
5. отложенные события



Имя состояния

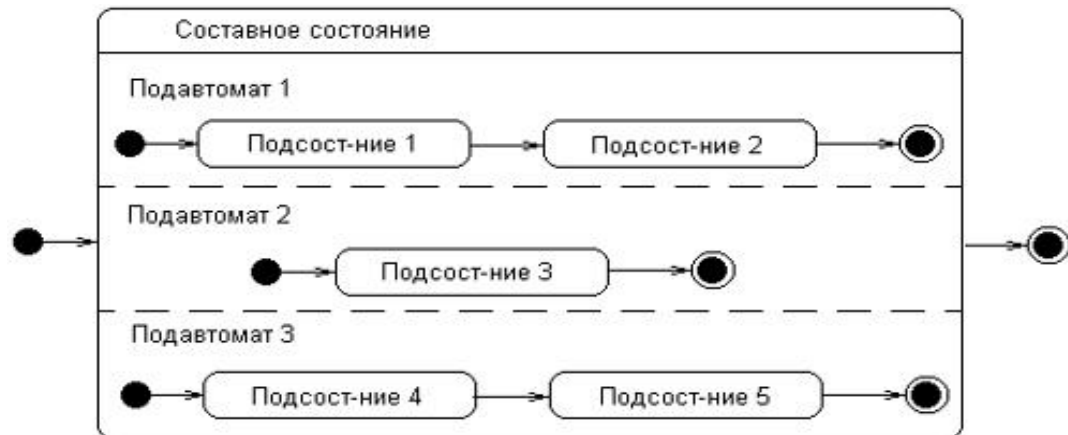
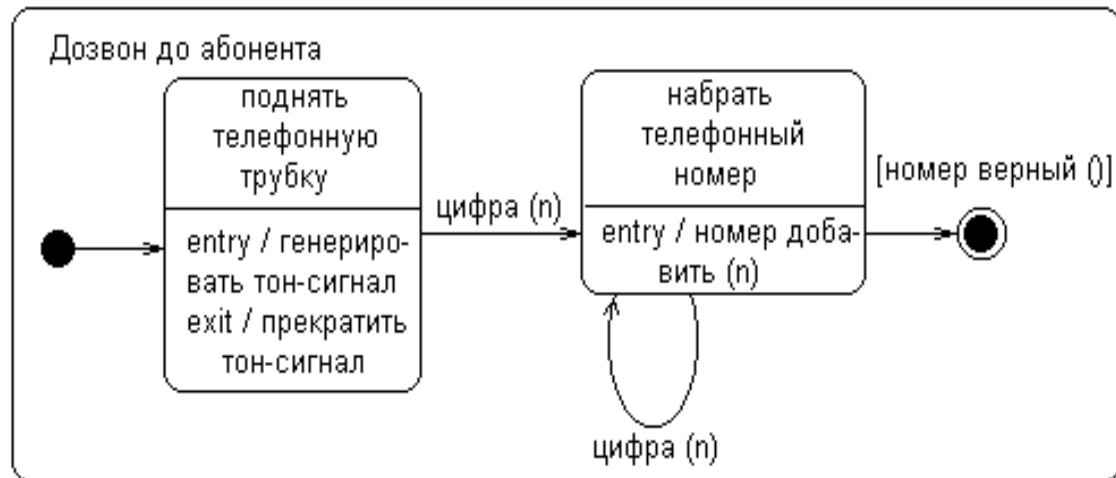


Имя состояния

Список внутренних
действий в данном
состоянии

- состояние изображается прямоугольником с закругленными углами
- состояния могут быть вложенными, последовательными и параллельными

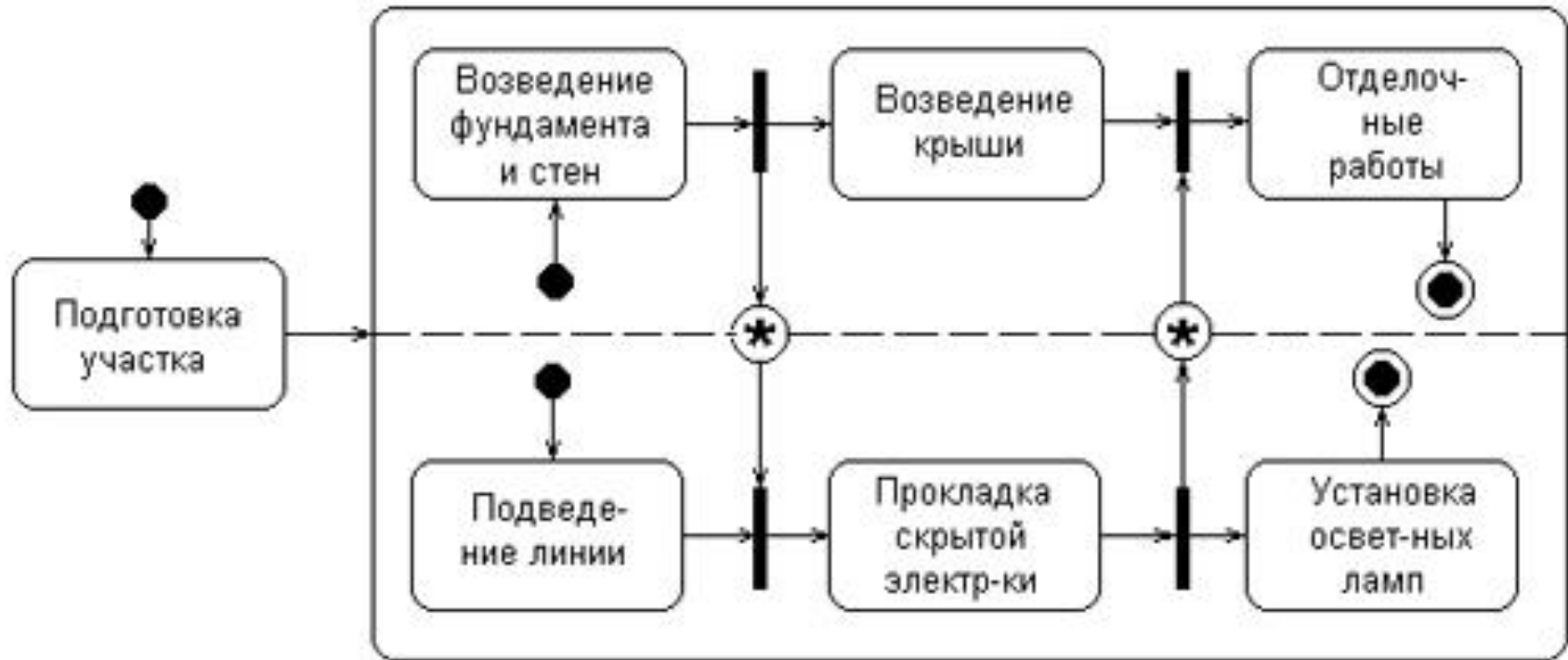
Примеры



Имя состояния

- строка текста, которая раскрывает содержательный смысл состояния
- всегда записывается с заглавной буквы
- рекомендуется использовать глаголы в настоящем времени (например, для принтера - **Печатает**, **Ожидает**) или соответствующие причастия (для сотрудника - **Занят**, **Свободен**)
- может отсутствовать

Синхронизирующие состояния



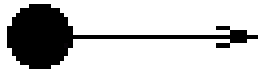
Действия

- Записываются в виде отдельной строки и имеют следующий формат: **метка-действия/выражение-действия**
- Метка указывает на условия, при которых будет выполняться деятельность, определенная выражением действия
- Перечень меток фиксирован в языке UML:
 1. **Entry** (входное действие)
 2. **Exit** (выходное действие)
 3. **Do** (выполняющуюся деятельность)
 4. **Include** (обращения к подавтомату)
- Пример:

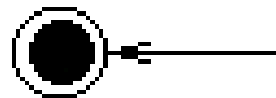
Ввод пароля
entry / установить символы невидимыми exit / установить символы видимыми символ / получить символ помощь / показать помощь

Начальное и конечное состояния

- представляют собой частные случаи состояния, которое также не содержит никаких внутренних действий (псевдосостояния)
- Начальное состояние - служит для указания места, от которого начинается процесс изменения состояний
- Конечное состояние - служит для указания места, в котором завершается жизненный цикл данного объекта



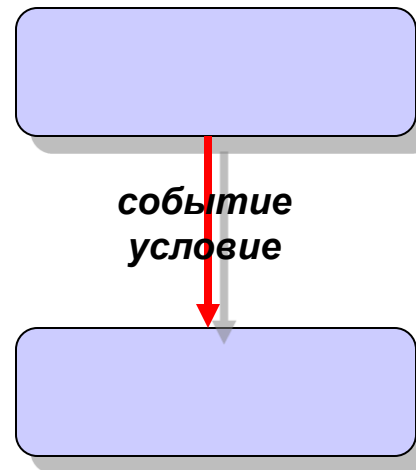
начальное состояние



конечное состояние

Переход

- **Переход** - это отношение между двумя состояниями, показывающее, что объект, находящийся в первом состоянии, должен перейти во второе состояние
- Элементы перехода:
 1. исходное состояние
 2. событие-триггер
 3. сторожевое условие
 4. действие
 5. целевое состояние
- переходы изображаются в виде линии со стрелкой, направленной от исходного к целевому состоянию



Переход

- Переход может быть направлен в то же состояние, из которого он выходит
- Каждый переход может помечен строкой текста, которая имеет следующий формат:

сигнатура события[сторожевое условие]/выражение действия

Переход

- *Сигнатура события* описывает некоторое событие с необходимыми аргументами:

имя события(список параметров)

- *Сторожевое условие* представляет собой некоторое булевское выражение и может отсутствовать. Если оно принимает значение «истина», то переход срабатывает
- *Выражение действия* выполняется в том и только в том случае, когда переход срабатывает. Это атомарная операция, выполняемая сразу после срабатывания соответствующего перехода до начала каких бы то ни было действий в целевом состоянии.

Переход (пример)

Пример спецификации перехода:



Содержание диаграмм состояний и их свойства

Диаграммы состояний

1. Один из пяти видов диаграмм, используемых для моделирования динамических аспектов системы
2. Показывает автомат
3. Используется для моделирования поведения реактивных объектов
4. Обычно диаграмма состояний включает в себя:
 1. простые и составные состояния
 2. переходы вместе с ассоциированными событиями и действиями

Диаграмма с параллельными подсостояниями

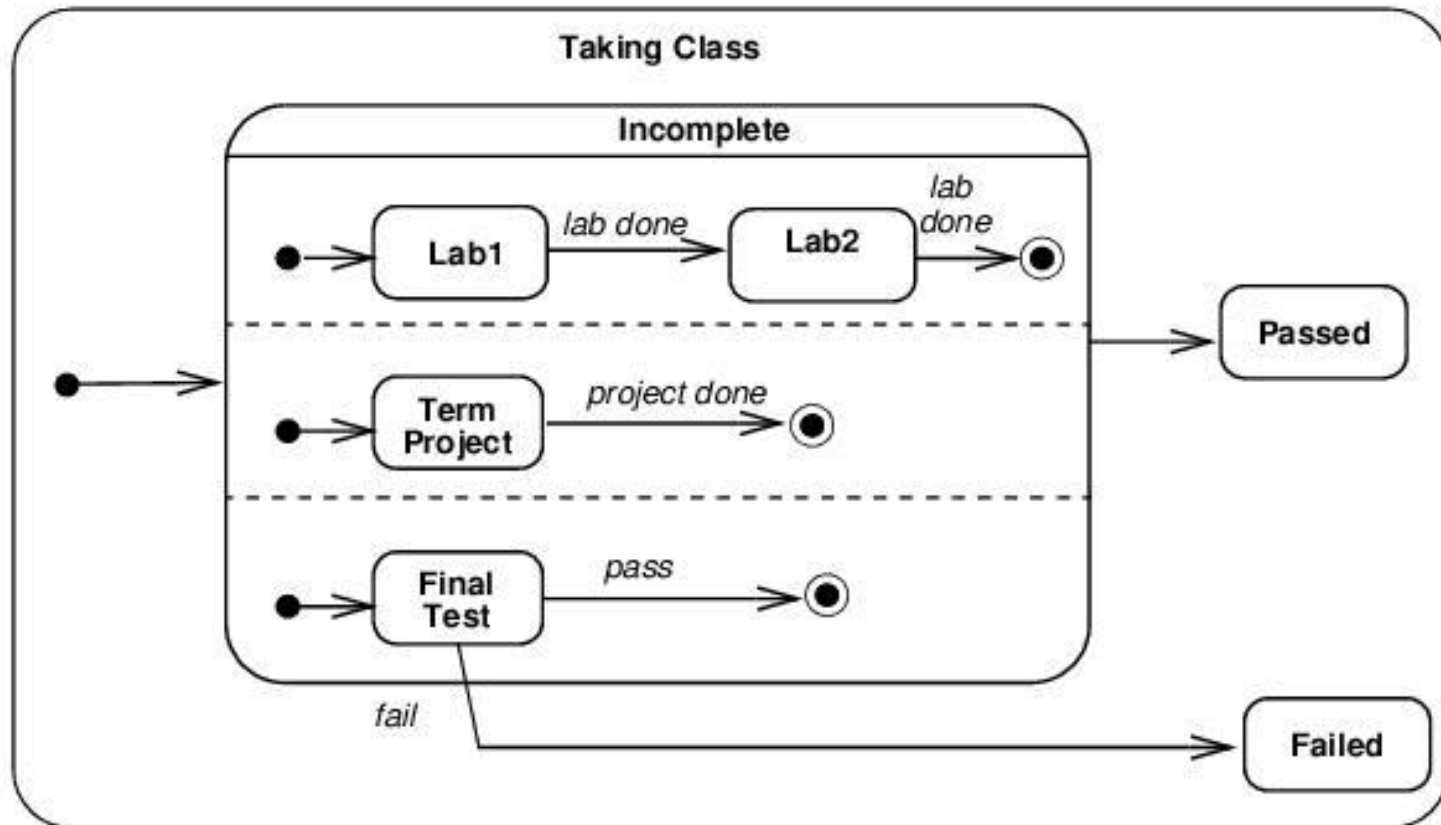
1. Параллельные подсостояния позволяют специфицировать несколько подавтоматов, которые могут выполняться *параллельно* внутри состояния
2. Каждый из подавтоматов занимает некоторую область внутри составного состояния, которая отделяется от остальных горизонтальной пунктирной линией.
3. Объект может одновременно находиться в каждом из параллельных подсостояний.

Диаграмма с параллельными подсостояниями

- На следующем слайде представлен конкретный пример диаграммы с параллельными подсостояниями для задачи прохождения курса студентом:

Для того, чтобы пройти курс, студент должен выполнить некоторое количество лабораторных работ, выполнить курсовой проект и пройти финальный тест.

Пример диаграммы с параллельными подсостояниями

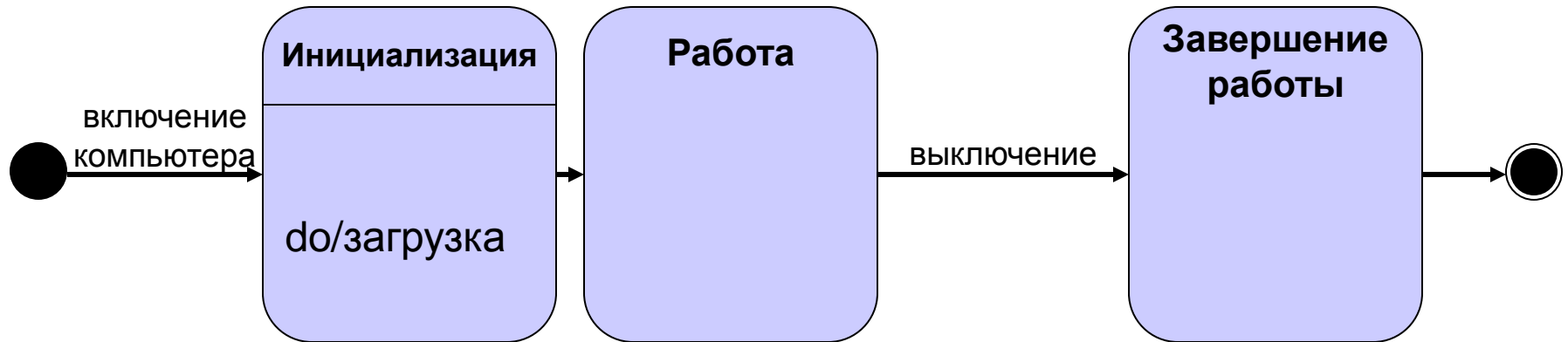


Развернутый пример диаграммы состояний

В качестве примера рассмотрим графический пользовательский интерфейс (GUI) ОС.

- GUI инициализируется при включении компьютера и завершает его работу при его выключении.
- можно выделить по крайней мере три устойчивых состояния GUI:
 - Инициализация
 - Работа
 - Завершение работы

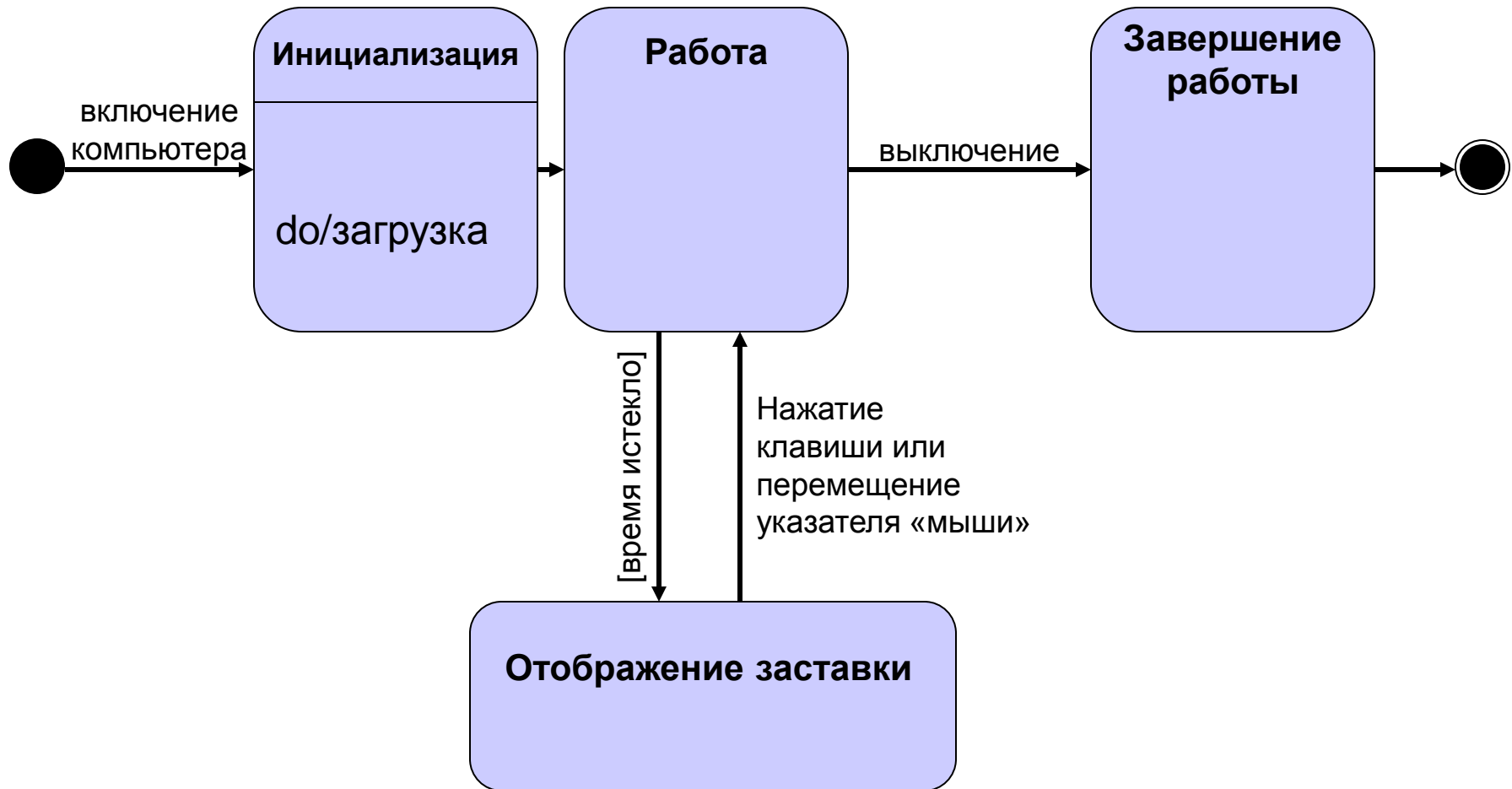
Развернутый пример – 1 приближение



Развернутый пример – продолжение

- Если не наблюдается никакой активности пользователя, активизируется хранитель экрана (заставка)
- В состояние «Отображение заставки» GUI переходит при условии истечения заданного промежутка времени и возвращается в состояние «Работа» при нажатии клавиши или перемещении указателя «МЫШИ»

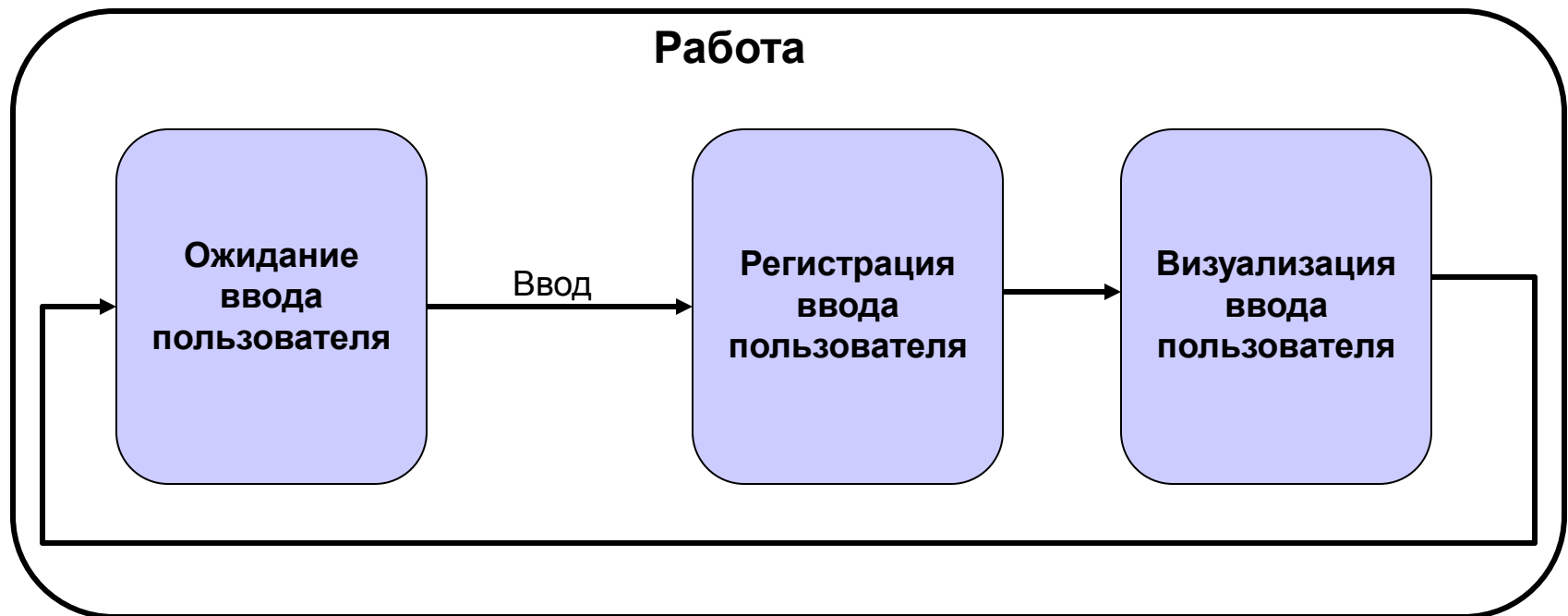
Развернутый пример – 2 приближение



Развернутый пример – продолжение

- Состояние «Работа» является более сложным по сравнению с остальными. В этом состоянии:
 - GUI постоянно ожидает действий пользователя
 - нажатия клавиш
 - перемещения указателя мыши
 - «щелчки» по кнопкам мыши
 - при выполнении таких действий, GUI регистрирует событие и
 - Изменяет содержимое экрана, чтобы отобразить действия пользователя
- Таким образом, состояние «Работа» имеет такие *последовательные подсостояния*:
 - Ожидание ввода пользователя
 - Регистрация ввода пользователя
 - Визуализация ввода пользователя

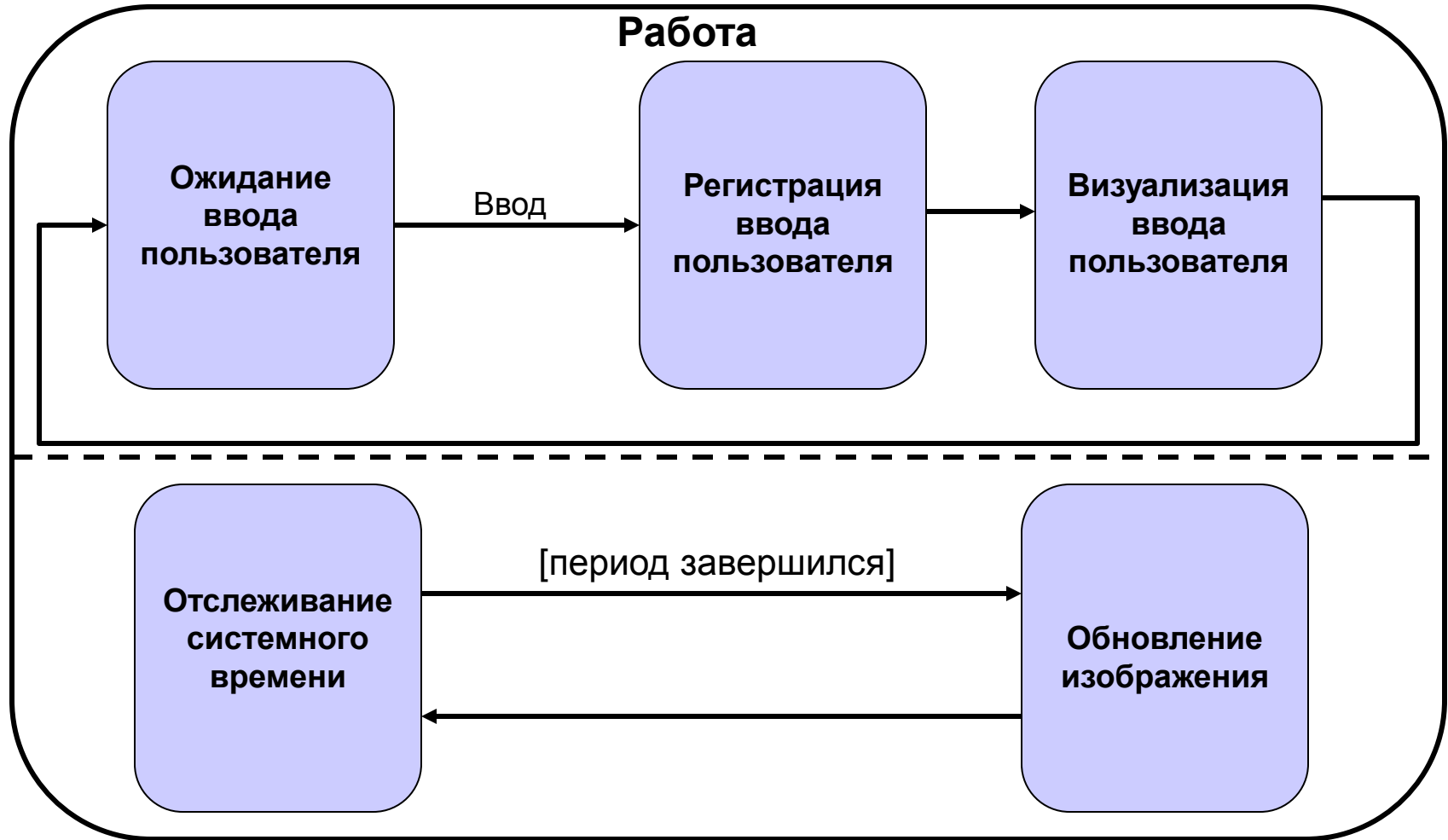
Развернутый пример – 3 приближение



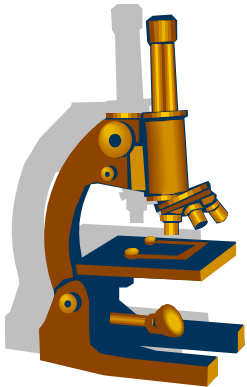
Развернутый пример – продолжение

- В состоянии «Работа» GUI не только ожидает действий пользователя и т.д.
- Также:
 - отслеживается системное время и
 - возможно, обновляется экран по истечении заданного промежутка времени
- Эта последовательность из двух подсостояний выполняется параллельно с ранее рассмотренными подсостояниями

Развернутый пример – 4 приближение



Развернутый пример – продолжение



Данный пример можно детализировать больше и больше...

Примечания

1. Диаграмма состояний - не обязательна и "присоединяется" к тому элементу, который, ведет себя нетривиально.
2. Из каждого состояния не может быть самопроизвольного перехода другое состояние.
3. Объект в каждый момент должен находиться в единственном состоянии.
4. Никакие два перехода из одного состояния не могут сработать одновременно.

Хорошо структурированная диаграмма состояний:

1. Сосредоточена на описании одного аспекта динамики системы.
2. Содержит только существенные для понимания этого аспекта элементы.
3. Описывает лишь соответствующие своему уровню абстракции детали.

Алгоритм построения диаграммы состояний

1. Выбрать контекст для автомата (класс, прецедент, систему).
2. Выбрать начальное и конечное состояния объекта.
3. Определить устойчивые состояния объекта.
4. Разумно/частично упорядочить устойчивые состояния.
5. Какие события могут инициировать переходы между состояниями?
6. Присоединить действия к переходам и/или к состояниям
7. как можно упростить автомат?
8. Проверить достижимость любого из состояний.
9. Убедиться в отсутствии тупиковых состояний.
10. Трассировать автомат.

Общие рекомендации по построению

1. Дайте диаграмме имя, соответствующее ее назначению.
2. Начинайте с моделирования устойчивых состояний объекта, затем переходите к допустимым переходам состояний.
3. Располагайте элементы так, чтобы число пересечений линий было минимальным.

Итоги

1. Диаграммы состояний - это один из пяти видов диаграмм в языке UML, используемых для моделирования динамических аспектов системы.
2. Диаграмма состояний показывает автомат.
3. Прямоугольники представляют состояния, стрелки представляют переходы от одного состояния к другому.
4. Состояние - это ситуация в жизни объекта, на протяжении которой он удовлетворяет некоторому условию, выполняет определенную деятельность или ожидает какого-то события.
5. Переход представляет собой отношение между двумя последовательными состояниями, которое указывает на факт смены одного состояния другим.