

UML - Universal Modelling Language –общие сведения

UML - Universal Modelling Language – универсальный язык моделирования – предназначен для создания унифицированных описаний моделей системных объектов. Его появление во второй половине 90-х годов прошлого века обусловлено сформировавшейся необходимостью оптимизации взаимодействия заказчиков и разработчиков при разработке сложных программных систем (уровня ERP), стандартизация описания реализаций этих систем. Очень быстро UML вышел за рамки сферы программных разработок и стал применяться при решении иных, зачастую не связанных с технологиями задач, например в реинжиниринге бизнес-процессов, превращаясь в индустриальный стандарт, универсальное средство общения специалистов из различных предметных областей. Концептуально UML отражает парадигму объектно-ориентированного проектирования. Математические основы UML – теория множеств, теория графов. В практике математического моделирования радиотехнических систем UML может быть эффективно использован как для собственно формирования моделей радиосистем, так и при разработке программных реализаций математических моделей.

Основные этапы развития :

1994 г. - разработка в рамках Rational Software Corporation языка объектно-ориентированного моделирования Unified Method 0.8, впоследствии превратившегося в UML.

1995 г. – поддержка UML консорциумом OMG (Object Management Group) - объединение ведущих производителей ПО (более 800 компаний и организаций).

1997 г. - подготовка спецификации UML 1.0, создание консорциума UML для поддержки спецификации (MS, Oracle, HP, IBM и др.).

1999 г. – публикация OMG спецификации UM 1.3.

Общая структура языка:

UML состоит из 2-х взаимодействующих частей:

- 1) Семантика UML – правила определяющие синтаксис и содержание понятий для моделей систем.
- 2) Нотация UML – набор графических представлений для визуального представления семантики UML.

В рамках UML представления о модели сложной системы фиксируются в виде специальных графических конструкций – диаграмм, канонический перечень которых включает следующие позиции:

- A) Диаграмма вариантов использования (Use Case Diagram)
- B) Диаграмма классов (ClassDiagram)
- C) Диаграммы поведения (Behavior Diagrams)
 - a) Диаграмма состояний (Statechart Diagram)
 - b) Диаграмма деятельности (Activity Diagram)
 - c) Диаграммы взаимодействия (Interaction Diagrams)
 - i) Диаграмма последовательности (Sequence Diagram)
 - ii) Диаграмма сотрудничества (Collaboration Diagram)
- D) Диаграммы реализации (Implementation Diagrams)
 - a) Диаграмма компонентов (Component Diagram)
 - b) Диаграмма развертывания (Deployment Diagram)

На диаграммах UML используются три типа визуальных обозначений:

- графические символы (значки или пиктограммы);

- связи;
- текст.

При составлении диаграмм придерживаются следующих основных рекомендаций:

- каждая диаграмма должна быть законченным представлением соответствующего фрагмента оригинала;
- все сущности на диаграмме должны быть одного концептуального уровня;
- вся информация о сущностях должна быть явно представлена на диаграмме;
- диаграмма не должна содержать противоречивой информации;
- диаграмма должна быть самодостаточной для однозначной интерпретации всех элементов;
- количество диаграмм для конкретной модели не является строго фиксированным.

Диаграмма вариантов использования (Use Case Diagram, UCD)

Цели разработки:

- определить границы модели на начальном этапе;
- сформулировать общие требования к функциональности модели;
- разработать исходную концептуальную модель;
- разработать исходную документацию для взаимодействия разработчиков и заказчиков.

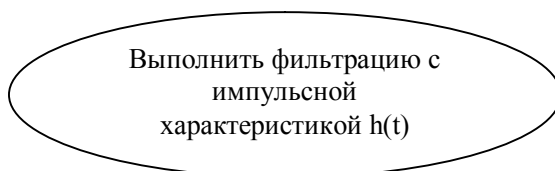
Элементы UCD:

- вариант использования;
- актер;
- интерфейс;
- отношение;
- примечание.

Элемент «вариант использования»

- применяется для спецификации общих особенностей поведения оригинала без рассмотрения его внутренней структуры. Вариант использования определяет действие, выполняемое моделируемой системой при взаимодействии с внешними сущностями.

Отдельный вариант использования обозначается эллипсом, внутри которого содержится его краткое название в форме глагола повелительного наклонения с пояснительными словами:



Каждый вариант использования соответствует сервису, представляющему моделируемый оригинал по запросу извне. Сервис, который инициализируется по запросу, представляет законченную последовательность действий – т.е. оригинал, после выполнения обработки запроса должен возвратиться в первоначальное состояние.

Элемент «актер»

- любая внешняя по отношению к оригиналу сущность, которая взаимодействует с системой и использует ее функциональные возможности для достижения определенных целей или решения частных задач.

Обозначается условной человеческой фигурой над которой записывается его имя:



В качестве актеров могут выступать пользователи, другие системы, подсистемы оригинала, устройства.

Актер находится вне оригинала, его внутренняя структура никак не определяется. Актеры взаимодействуют с оригиналами путем передачи и приема сообщений от вариантов использования.

Элемент «интерфейс»

- служит для спецификации параметров модели, которые видны извне без указания их внутренней структуры.

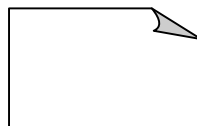
Изображаются в виде круга, рядом с которым записывается его имя:



Интерфейсы определяют операции, которые обеспечивают функциональность оригинала для актеров.

Элемент «примечание»

- служат для включения в диаграмму произвольной текстовой информации – комментарии разработчика, ограничения, параметры и т.д.



Элемент «отношение»

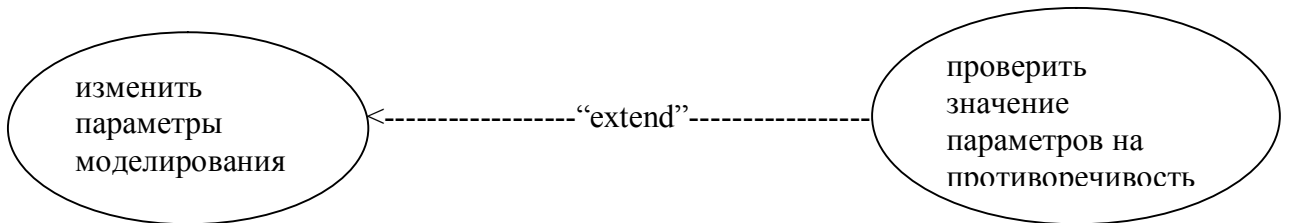
- описывает взаимодействия между актерами и вариантами использования.

В UML определены следующие типы отношений:

а) отношение «ассоциации» - базовый тип, служит для обозначения специфической роли актера при взаимодействии с вариантом использования. Параметром отношения ассоциации является «кратность» компонента, участвующего в данной ассоциации, которая характеризует количество экземпляров элемента, участвующего в ассоциации.



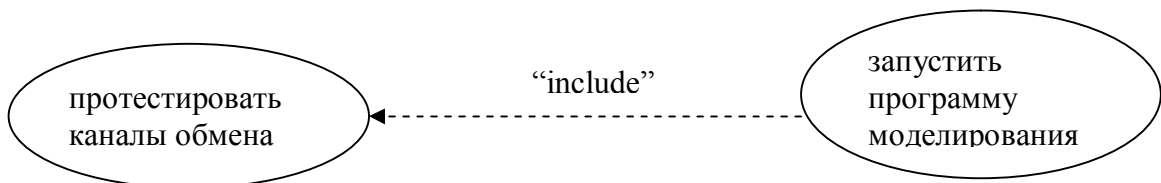
б) отношение «расширения» - определяет взаимосвязь экземпляра отдельного варианта расширения с более общим вариантом



в) отношение «обобщения» - указывает на тот факт, что некоторый вариант А использования может быть обобщен до варианта использования В



г) – отношение «включения» - указывает на то, что некоторое заданное поведение одного варианта использования включается в качестве составного компонента в последовательность поведения другого варианта использования



Пример UCD - модель программной реализации радиотехнической системы:

Модель программной реализации радиотехнической системы (РТС)

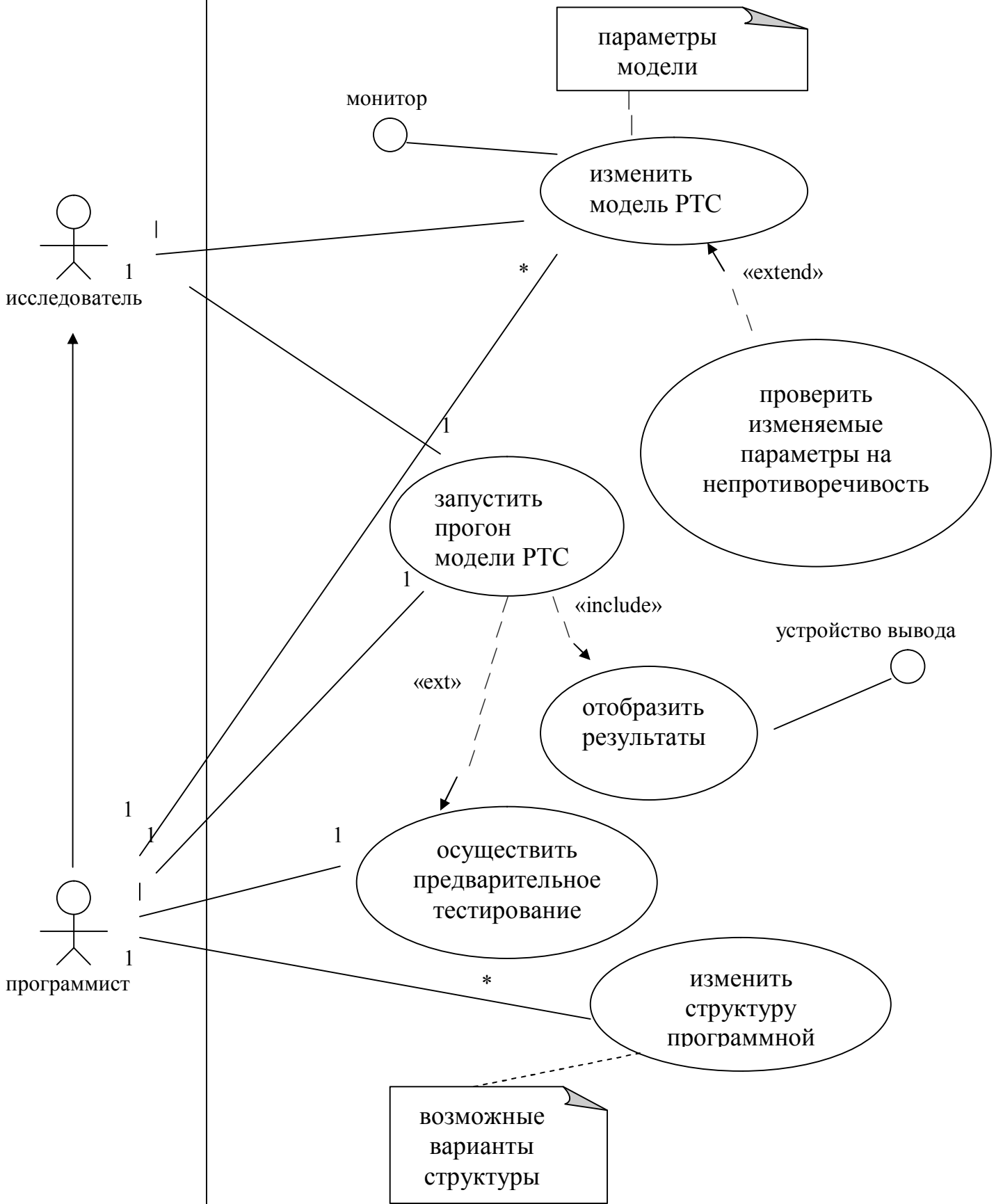


Диаграмма состояний (Statechart Diagram, SCD)

Цели разработки:

- сформулировать представления о поведении системы во времени и в зависимости от реакции на внешнее воздействия;
- описать возможные последствия состояний и переходов.

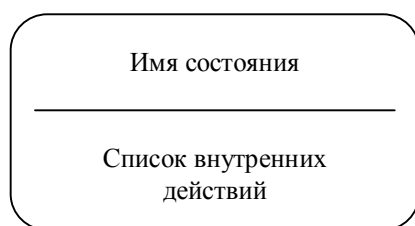
Диаграмма состояний является графом специального вида, вершины которого соответствуют состояниям, а дуги – переходам между состояниями. Диаграмма состояний могут быть вложены друг в друга.

Элементы SCD:

- состояние;
- переход.

Элемент «состояние»

- абстрактная сущность используемая для моделирования отдельной ситуации, в течение которой имеет место выполнение некоторого действия.



Имя состояния всегда записывается с заглавной буквы, должно раскрывать содержательный смысл этого состояния. Рекомендуется в качестве имени использовать глаголы 3-го лица настоящего времени. Имя может отсутствовать (анонимное состояние).

Список внутренних действий содержит перечень действий, выполняемых при нахождении в заданном состоянии и имеет формат:

<метка действия / выражение действия>

Метка указывает на обстоятельства или условия, при которых будут выполняться действия; метка имеет фиксированные значения:

- entry- действие в момент входа в состояние;
- exit- действие в момент выхода из состояния;
- do- специфицирует деятельность в состоянии;
- include- используется для спецификаций действий подсостояний.

Во всех остальных случаях метка идентифицирует событие, которое запускает соответствующее выражение действия.

Состояния могут быть:

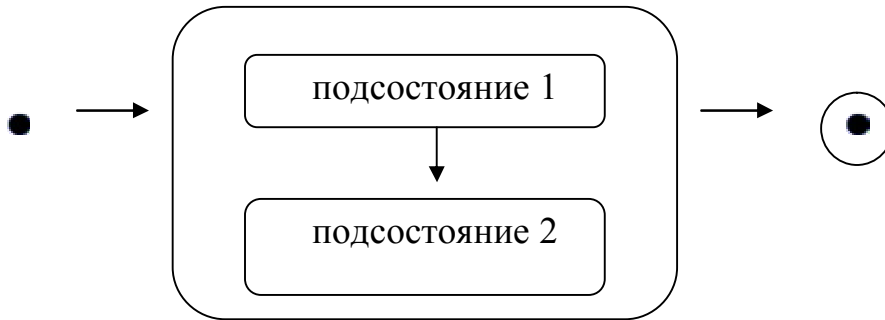
- псевдосостояниями, которые не содержат внутренних действий:
 - начальное



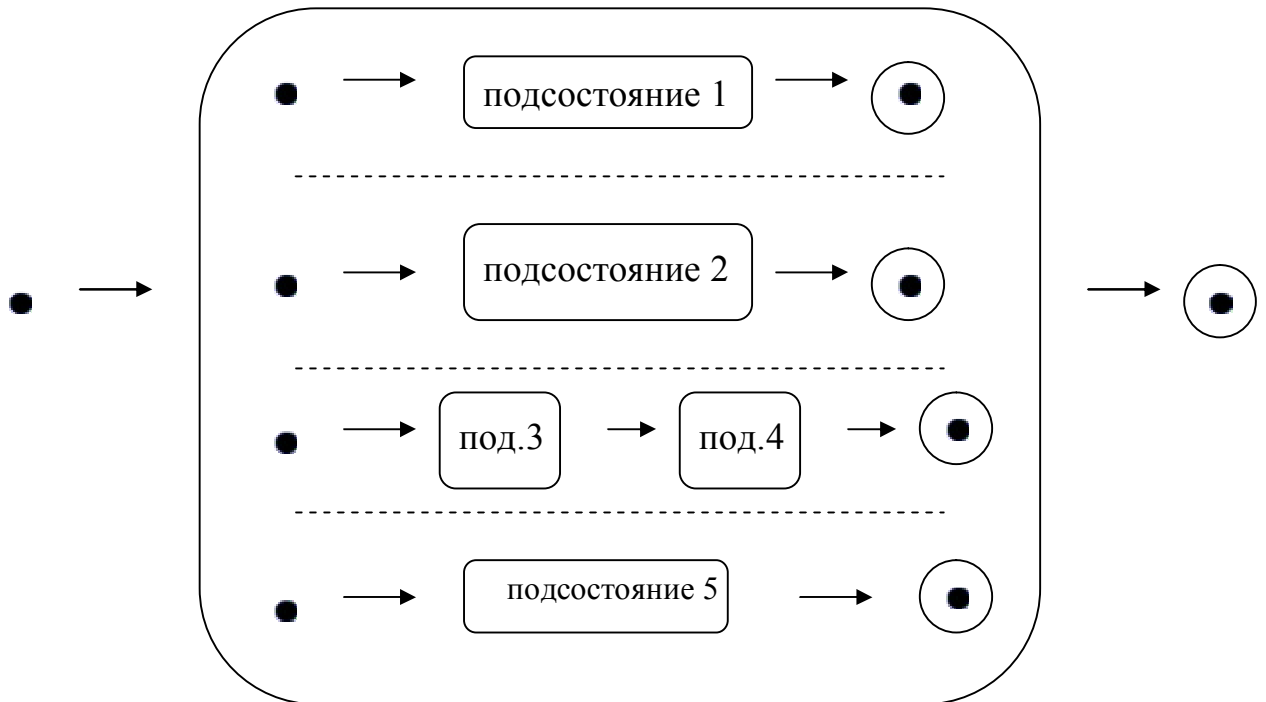
○ конечное



- составными, включающими вложенные подсостояния



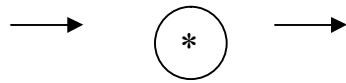
- последовательными, не пересекающимися во времени;
- параллельными, внутреннего действия которых могут выполняться одновременно;



- историческими, содержащими «память» о пребывании в каких-то состояниях;



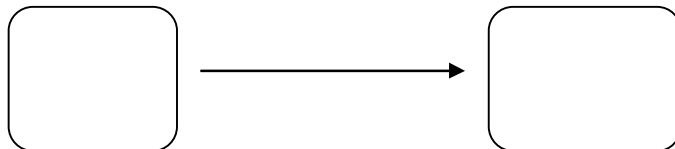
- ролирующими – псевдосостояния, используемые для синхронизации наступления отдельных событий



Элемент «переход»

- представляет отношение между состояниями, которое указывает на факт смены одного состояния другим.

Переход осуществляется при наступлении некоторого события, имя которого указывается на переходе в формате <имя события> (<список параметров события через запяты>). На диаграмме состояний переход изображается сплошной линией со стрелкой, которая направлена на целевое состояние.



Переход может быть помечен строкой текста, следующего формата:

<событие> [<сторожевое условие >] / <выражение действия>

Сторожевое условие – выражение, определяющее срабатывание перехода.

Выражение действия – вычисляется в момент срабатывания перехода.

Диаграмма классов (Class Diagram, CD)

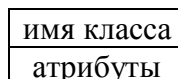
CD служит для представления статистической структуры в терминологии классов объектно-ориентированного проектирования. Отражает взаимосвязи между объектами и подсистемами, описывает их внутреннюю структуру и типы отношений вне зависимости от времени.

Элементы диаграммы классов:

- класс,
- отношения,
- интерфейсы.

Элемент класс

- обозначает множество объектов, которые обладают одинаковой структурой, поведением и отношениями с объектами из других классов.



класса
операции класса

Имя класса – существительное(ые) без пробелов с заглавной буквы. Если указывается имя надкласса, то оно записывается перед именем класса с двойным двоеточием: “Система_навигации::Спутниковая_подсистема.”

Атрибуты класса – его свойства, записываются в формате:
 <квантор видимости><имя атрибута>[кратность]:<тип атрибута>=<исходное значение>{свойство}

Квантор видимости:

+ - общедоступный (public),

- защитный (protected) – недоступен из других классов, кроме подклассов,

- - закрытый (private) – недоступен для всех классов.

Кратность – количество значений, которые может иметь данный атрибут [0..10, 14..*]

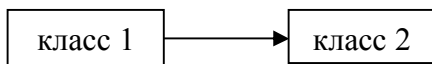
Операция класса – сервис, предоставляемый объектами данного класса, записывается в виде:

<квантор видимости><имя операции>(параметры):<тип возвращаемого значения>{свойства}

Элемент “отношение”

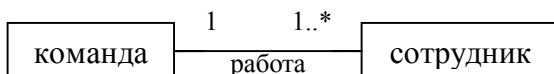
– специфицирует связи между классами.

А) Отношение зависимости указывает на связь, которая не является другим типом отношения:

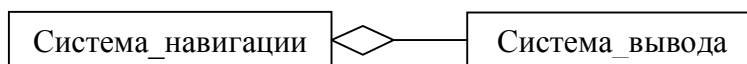


Стрелка – от класса-клиента к классу-источнику.

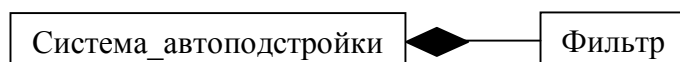
Б) Отношения ассоциации – указывает на связь в общем случае, обозначается сплошной линией:



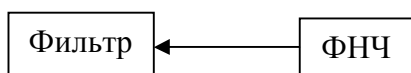
В) Отношение агрегации связывает целое и частное:



Г) Отношение композиции – частный случай отношения агрегации:



Д) Отношение обобщения – таксономическая связь между более общим элементом и специальным элементом:



Элемент “интерфейс”

служит для спецификации элементов модели, которые видимы извне без указания их внутренней структуры:

“interface” датчик 1
Значения

Диаграмма деятельности (Activity Diagram, AD)

AD служит для моделирования процесса выполнения операций. Во многом аналогична блок-схеме алгоритма работы. Является частным случаем диаграммы состояний, отличие заключается в семантике состояний, которые используются для представления не деятельности, а действий и в отсутствии на переходах сигнатуры событий.

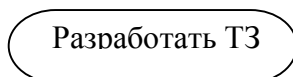
Основное применение диаграммы деятельности – визуализация операций классов, когда необходимо представить алгоритмы их выполнения, при этом каждое состояние AD может соответствовать операции некоторого класса, либо ее части.

Элементы AD:

- состояния действия,
- объекты,
- переходы.

Элемент “состояние действия”:

- не может иметь внутренних переходов, является элементарным. Выход из состояния действия означает его завершение. Обычно соответствует одному шагу алгоритма;
- изображается в виде вытянутого овала с выражением действия (глагол с пояснительными словами):

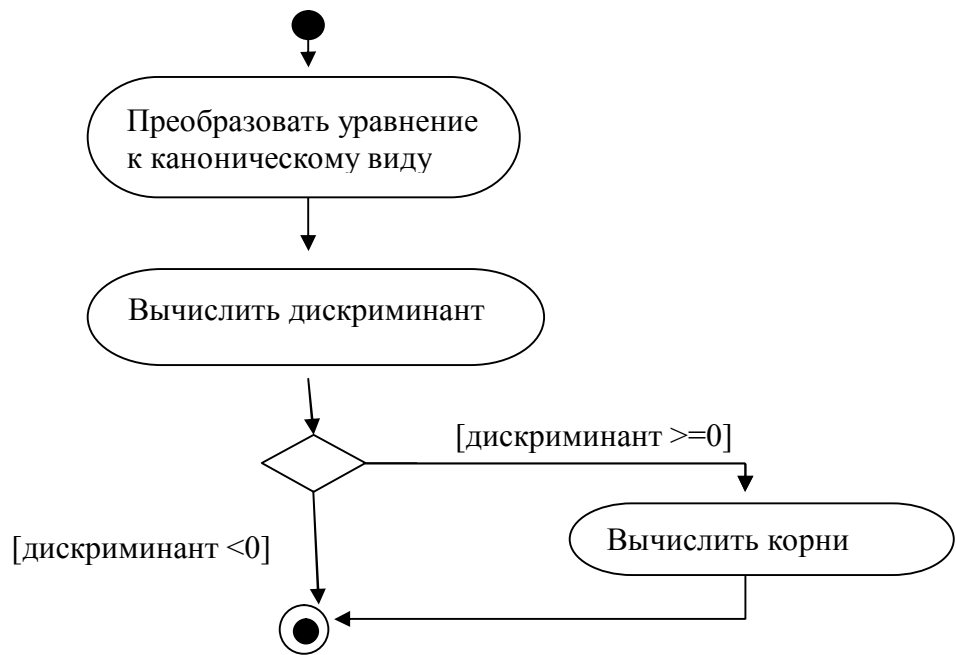


AD должна иметь единственное начальное и единственное конечное состояния.

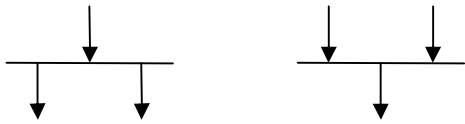
Элемент “переход”.

В AD используются только нетриггерные переходы, т.е. переходы, которые срабатывают сразу после завершения действия состояния. Изображается в виде сплошной линии со стрелкой. Если из состояния возможно несколько переходов, то срабатывать может только один из них. В этом случае каждый переход маркируется сторожевым условием (условием срабатывания). Ветвления переходов на AD обозначаются ромбом.

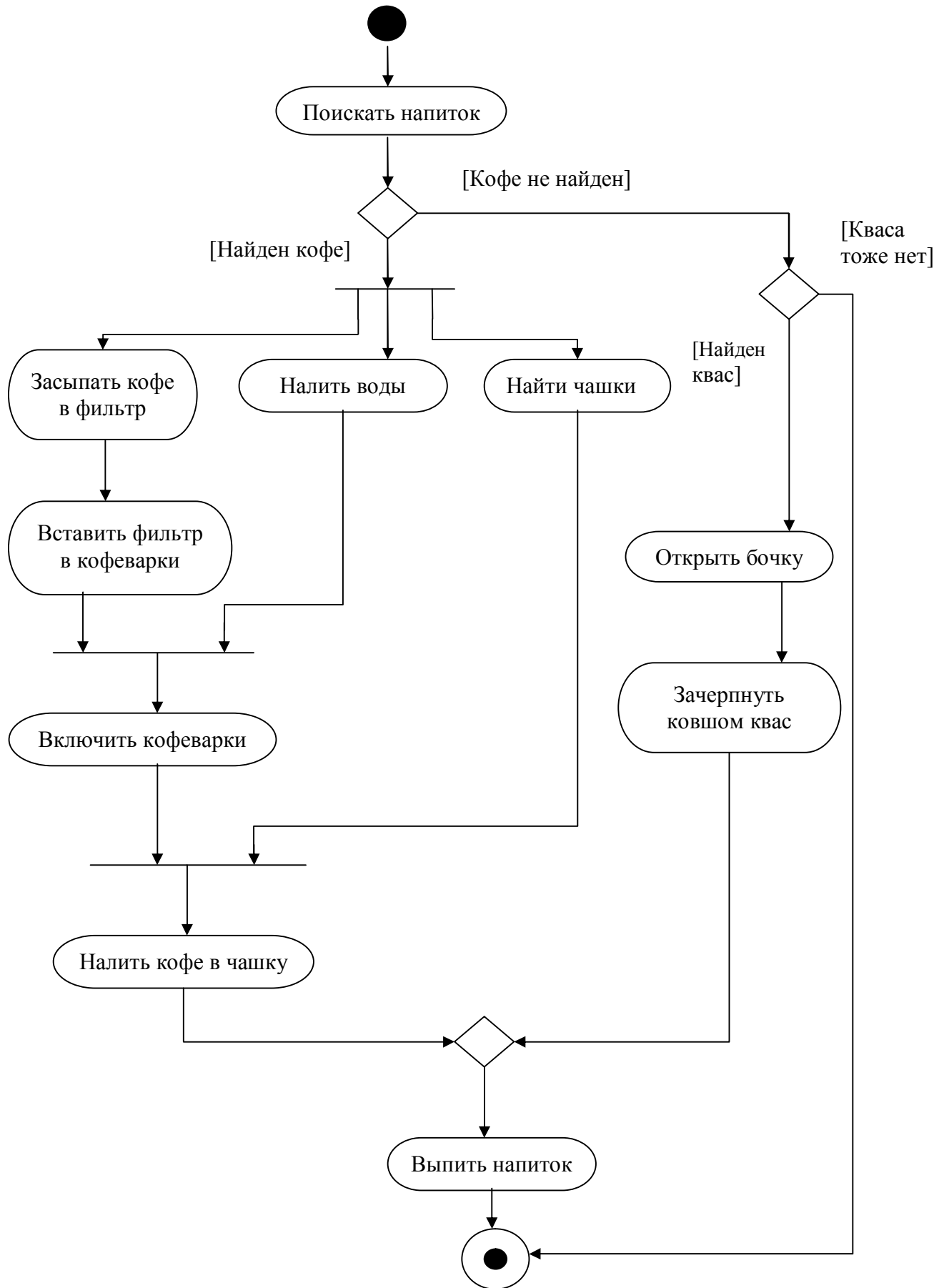
Пример АД для процедуры решения квадратного уравнения:



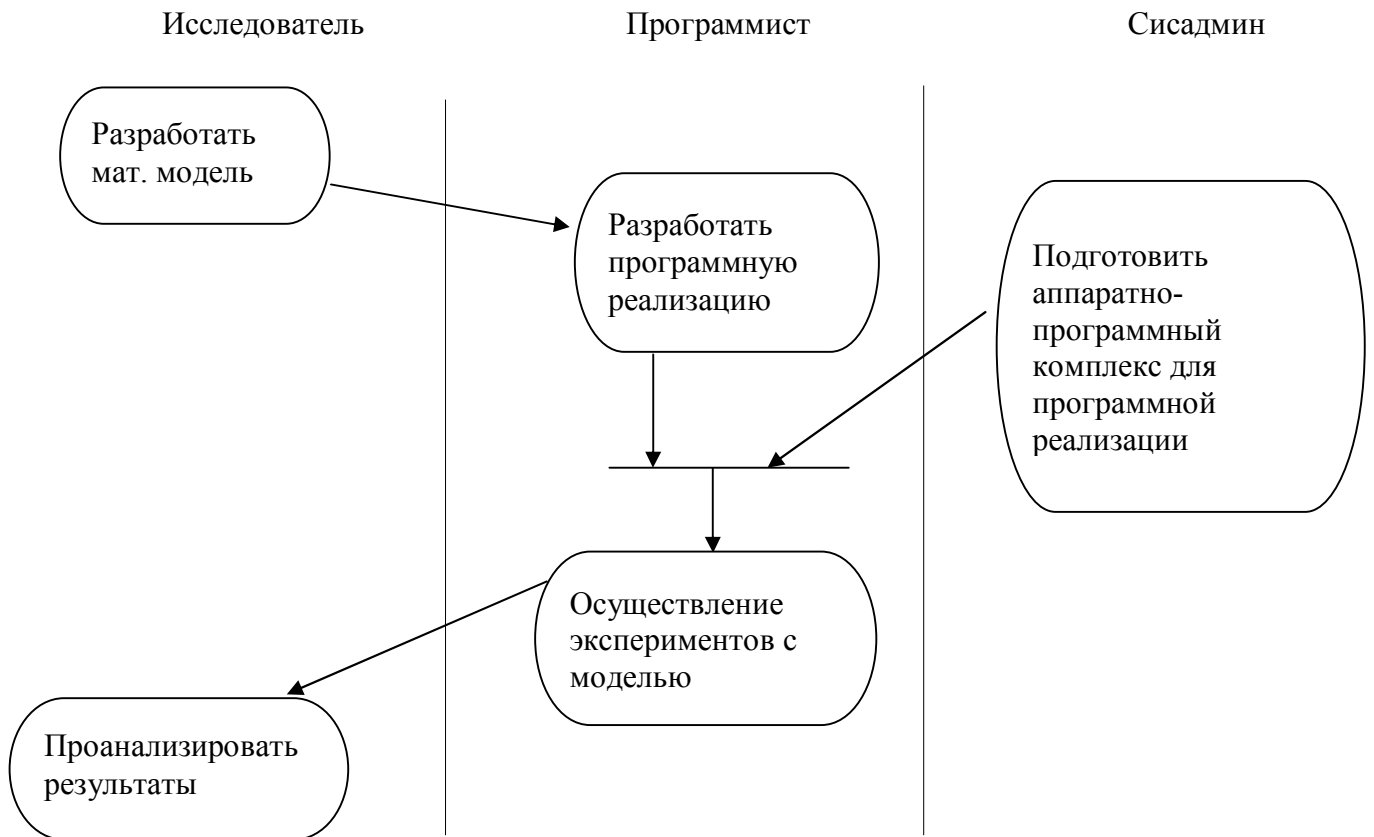
Для изображения разделения и слияния параллельных потоков действий используются символы:



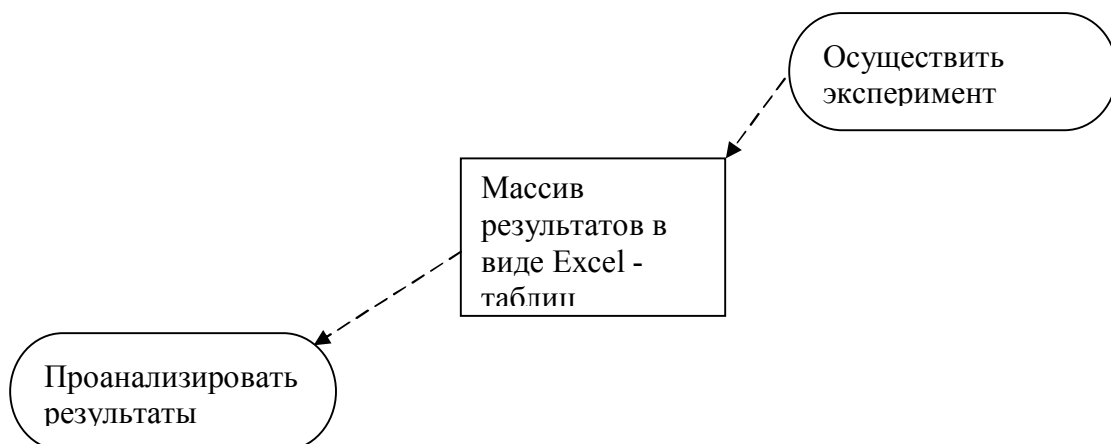
Пример АД для процедуры «выпить чег-нибудь».



Для отображения связи действий, выполняемых разными участниками моделируемого процесса, используются так называемые «дорожки» (swimlanes)

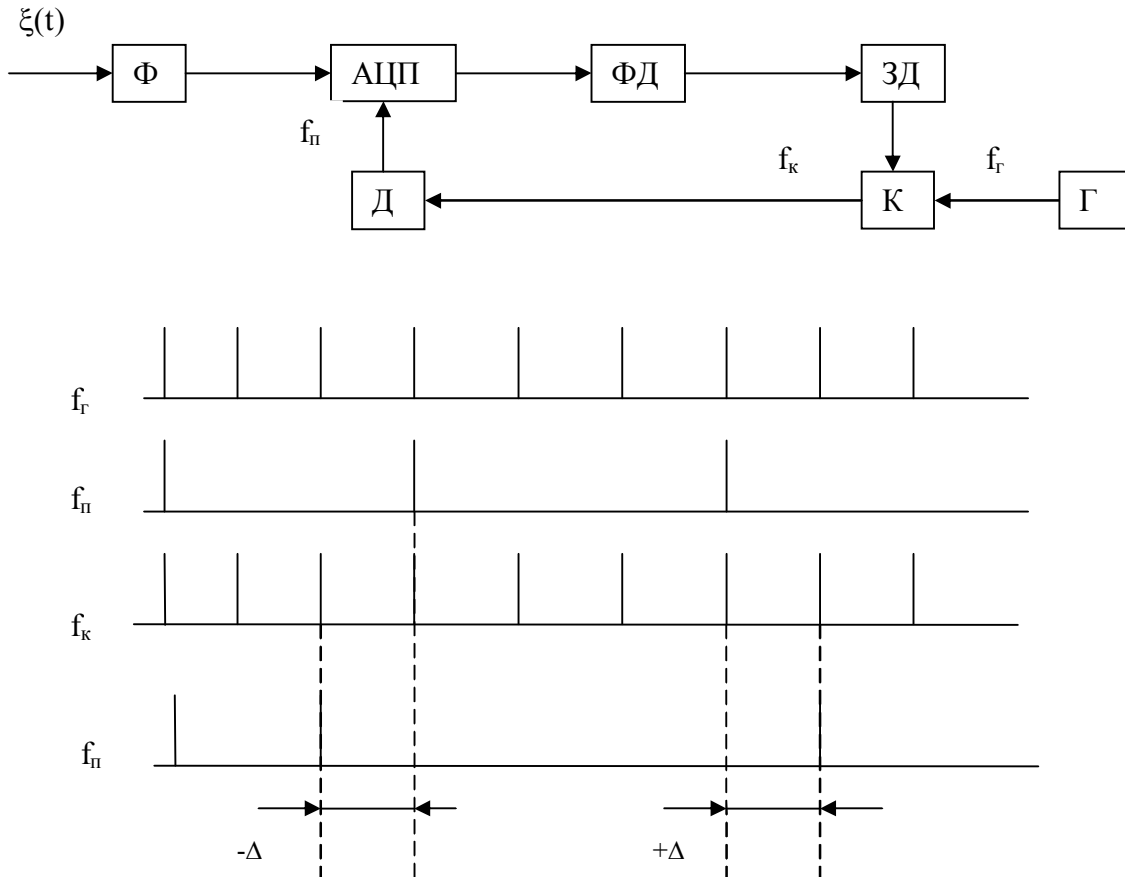


Элемент “объект” – является результатом действия и предметом действий. Обозначаются прямоугольником класса, имя объекта подчеркиваются. Связи объекта с состояниями обозначаются пунктирными линиями со стрелками:



ПР1. Формирование диаграммы состояний в нотации UML для системы цифровой фазовой автоподстройки (ЦФАП).

1) Структурная схема и описание работы ЦФАП



На вход ЦФАП действует смесь

$$\xi(t) = S(t) + U(t)$$

где $S(t)$ – полезный сигнал (пусть для простоты $S(t)$ – прямоугольной формы) с амплитудой A_s и периодом T_s .

$U(t)$ – нормальный белый шум с $M\langle U(t) \rangle = 0$ и спектральной плотностью N_u .

Φ – узкополосный фильтр с полосой пропускания ΔF .

АЦП – аналого-цифровой преобразователь с частотой дискретизации $f_{\Pi} = 2\Delta F$.

ФД – фазовый детектор, который определяет выборки, соответствующие моментам смены знака $S(t)$. Ошибка синхронизации при этом равна интервалу τ между переходом $S(t)$ и моментом взятия выборки. Наличие $U(t)$ определяет существование дополнительной ошибки синхронизации, величина которой зависит от соотношения сигнал/шум ($S(t)/U(t)$). Чтобы уменьшить влияние шума на точность синхронизации, коррекция положения выборок фазовых переходов сигнала производится по m выборкам за время

$$T = mT_s.$$

ЗД- знаковый детектор, сигнал на выходе которого появляется один раз за время T и равен знаку суммы m выборок переходов.

K – корректор, добавляет или исключает импульсы в последовательности f_r в зависимости от величины знака и знака сигнала ЗД.

D – делитель частоты с коэффициентом деления $R = f_r / 2\Delta F$.

G – генератор с частотой f_r .

Добавления или исключения в последовательности f_r изменяет в нужную сторону фазу последовательности импульсов, используемых для АЦП, т.е. изменяет положение выборок из $\xi(t)$ в нужную сторону на величину Δ - период f_r .

2) список состояний ЦФАП

Имя состояния	Список внутренних действий
Фильтрация	do/ выделение полосы ΔF в спектре входного сигнала
Генерация	do/ формирование опорной частоты f_r
Деление частоты	do/ формирование частоты $f_n = f_r/R$; $R = f_r/2\Delta F$
Детектирование фазы	do/ выделение выборок, соответствующих фазовым переходам входного сигнала
Знаковое детектирование	entry/ очистить память do/ фиксировать величину и знак выборок в моменты фазовых переходов в течение m циклов exit/ просуммировать накопленные данные
Коррекция фазы частоты АЦП	Сигнал на выходе знакового детектора положительный - добавить импульс в f_r Сигнал на выходе знакового детектора отрицательный - удалить импульс из f_r
Преобразование аналог-цифра	do/ брать отсчеты сигнала на входе с частотой f_n

3) Диаграмма состояний ЦФАП

