Лабораторная работа № 3

Разработка моделей на основе диаграмм состояний (стейтчарты) в AnyLogic 6

Цель работы: научиться описывать изучаемые процессы и объекты через диаграммы состояний.

Приобретаемые навыки:

Теоретическая часть

Для теоретической части использованы материалы справочной системы Anylogic адресу http://www.xjtek.ru/anylogic/help/.

Если у активного объекта можно выделить несколько состояний, выполняющих различные действия при происхождении каких-то событий, или если у активного объекта есть несколько качественно различных поведений, последовательно сменяющих друг друга при происхождении определенных событий, то поведение такого объекта может быть описано в терминах диаграммы состояний. Диаграмма состояний позволяет графически задать пространство состояний алгоритма поведения объекта, а также события, которые являются причинами срабатывания переходов из одних состояний в другие, и действия, происходящие при смене состояний.

Для разработки и реализации таких моделей используются элементы панели инструментов (палитры) Диаграммы состояний:

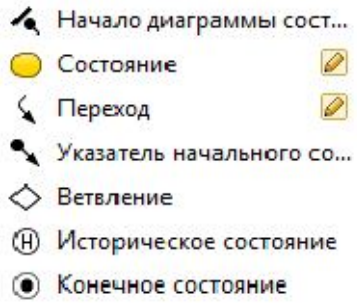

Рисунок 1 – Элементы диаграммы состояний

Для создания диаграммы нужно использовать три основных инструмента:

- «Начало диаграммы» - отмечает начальную точку обработки стейтчарта.
- «Состояние» - задает состояние диаграммы.
- «Переход» - используется для соединения состояний.
- «Указатель начального состояния» - служит для отметки состояния, с которого начинается обработка вложенной последовательности состояний.
- «Конечное состояние» - отмечает точку завершения обработки состояний.

Элементы диаграммы состояний добавляются на диаграмму путем перетаскивания соответствующих элементов из палитры **Диаграмма состояний**.

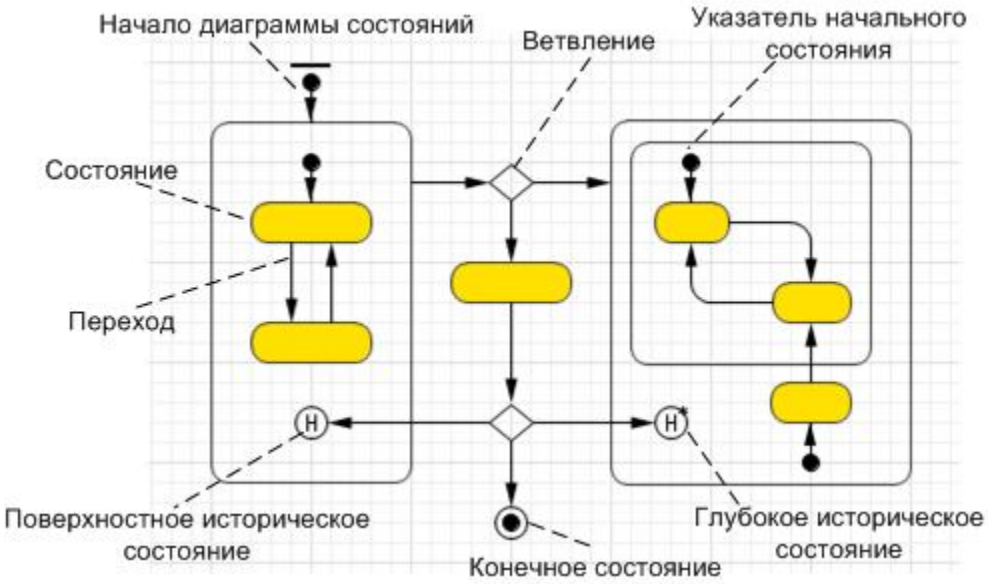

Рисунок 2 – Элементы диаграммы состояний

Наличие начала диаграммы обязательно.

Состояние

Состояние представляет собой местонахождение управления диаграммы состояний. Вы можете задать действия, которые должны быть выполнены при происхождении определенных событий и/или выполнении некоторых условий. Состояние может быть как простым, так и сложным (если оно содержит в себе другие состояния). Управление всегда принадлежит одному из простых состояний, а текущий набор действий включает в себя действия как текущего простого состояния, так и действия всех сложных состояний, содержащих это простое – то есть, может сработать переход, выходящий из любого из этих состояний.

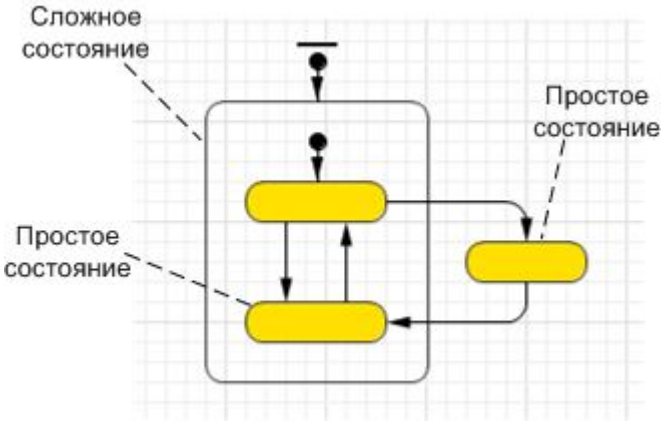

Рисунок 3 – Примеры типов состояний

Основные свойства:

Имя – Имя состояния. Имя используется для идентификации состояния и доступа к нему из кода.

Отображать имя – Если опция выбрана, то имя состояния будет отображаться в графическом редакторе.

Исключить – Если опция выбрана, то состояние будет исключено из модели.

На презентации – Если опция выбрана, то состояние будет отображаться на презентации во время выполнения модели.

Цвет заливки - Задает цвет заливки состояния. Щелкните мышью внутри элемента управления и выберите нужный цвет из списка наиболее часто используемых

цветов или же выберите любой другой цвет с помощью диалога **Цвета**. Если Вы не хотите, чтобы состояние было закрашено, выберите **Нет заливки**.

Действие при входе - Код, выполняемый, когда управление переходит в это состояние (состояние становится активным).

Действие при выходе - Код, выполняемый, когда управление покидает это состояние (состояние перестает быть активным).

Переход

Переход означает переключение управления диаграммы состояний, ее переход из одного состояния в другое. Переход означает, что если происходит заданное событие срабатывания перехода, и выполняется заданное дополнительное условие, то диаграмма состояний переключается из одного состояния в другое и выполняет заданные действия. Когда это происходит, мы говорим, что срабатывает переход.

Если переход пересекает состояние, но и начальная и конечная точки этого перехода лежат за пределами состояния, то считается, что это состояние не участвует в процессе смен состояний диаграммы состояний, и ни действие при входе, ни действие при выходе из этого состояния выполняться не будут.

Внутренние переходы

Есть специальный тип перехода, называемый внутренним переходом. *Внутренний переход* лежит внутри состояния, причем как начальная, так и конечная точки этого перехода лежат на границе этого состояния. Поскольку внутренний переход не покидает состояние, то не выполняются ни действия, которые должны выполняться при выходе из этого состояния, ни действия, выполняемые при входе в него. Более того, не изменяется и текущее простое состояние этого сложного состояния. Поэтому внутренний переход очень удобен для выполнения фоновых задач, которые не должны прерывать основную активность сложного состояния.

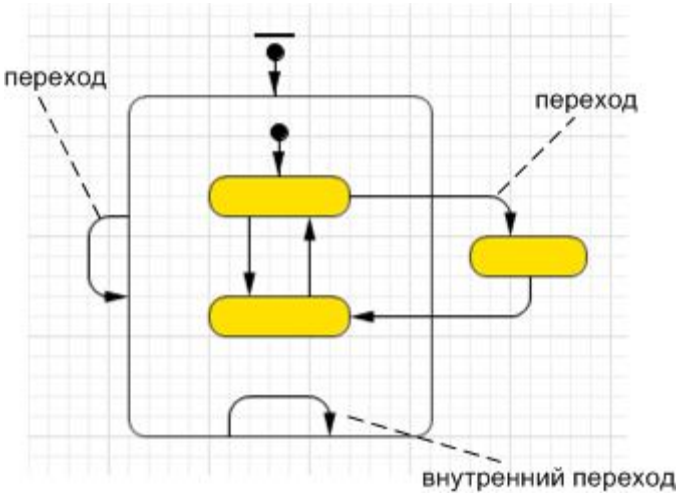

Рисунок 4 – Примеры типов переходов

Основные свойства:

Имя – Имя перехода. Имя используется для идентификации перехода.

Отображать имя – Если опция выбрана, то имя перехода будет отображаться в графическом редакторе.

Исключить – Если опция выбрана, то переход будет исключен из модели.

На презентации – Если опция выбрана, то переход будет отображаться на презентации во время выполнения модели.

Происходит – Выберите здесь тип события, при происхождении которого переход сработает:

- **По таймауту** – переход будет активирован, если истечет время заданного таймаута.

- **С заданной интенсивностью** - переход будет активироваться с заданной интенсивностью.
- **При выполнении условия** - переход будет активирован, когда будет выполнено заданное логическое условие.
- **При получении сообщения** - переход будет активирован по прибытии сообщения в соединенный с диаграммой состояний порт.
- **По прибытию агента** - переход будет активирован, когда агент (чье поведение задается этой диаграммой состояний) достигнет точки назначения.

Таймаут – [Только для перехода, происходящего по таймауту] Таймаут, по истечении которого сработает переход.

Интенсивность – [Только для перехода, происходящего с заданной интенсивностью] Интенсивность, с которой будет срабатывать данный переход. Переход активируется по таймауту, вычисленному согласно экспоненциальному распределению с параметром, равном заданной **Интенсивности** (таймаут отсчитывается от момента входа управления в состояние, из которого выходит данный переход). То есть, если интенсивность равна 5, то переход будет срабатывать в среднем 5 раз в единицу модельного времени.

Условие – [Только для перехода, происходящего при выполнении условия] Логическое условие, при выполнении которого будет активирован переход.

Тип сообщения – [Только для перехода, происходящего по прибытии сообщения] Здесь Вы выбираете тип сообщения, при получении которого сработает переход. Вы можете выбрать один из наиболее часто используемых типов (**int**, **double**, **boolean**, **String**), выбрав соответствующую опцию справа, либо же задать любой другой Java класс, выбрав опцию **Другой** и введя имя класса в поле **Имя класса**.

Осуществлять переход – [Только для перехода, происходящего по прибытии сообщения] Здесь Вы можете задать дополнительное условие, выполнение которого будет требоваться для срабатывания перехода:

- **Безусловно** – Выберите эту опцию, если Вы не хотите производить проверку типа сообщения.
- **Если сообщение равно** - Если опция выбрана, то переход будет срабатывать только по приходе сообщений, удовлетворяющих заданному в поле справа дескриптору.
- **Если выполняется условие (сообщение доступно как msg)** - Здесь Вы можете ввести код сложной проверки содержимого сообщения (только что полученное сообщение доступно здесь как локальная переменная msg).

Действие – Последовательность выражений Java, выполняемых при срабатывании перехода.

Доп. условие – Логическое выражение, разрешающее (если оно истинно, т.е. равно true) или запрещающее (если равно false) срабатавание перехода. Если условие не задано, то подразумевается true.

Свойства переходов, исходящих из **ветвлений**, отличаются от свойств обычных переходов:

Свойства переходов, ведущих из ветвления:

Условие – Если опция выбрана, то этот переход будет срабатывать, если заданное в поле справа логическое условие будет истинно.

По умолчанию (выбирается, если все остальные условия не выполняются) – Если опция выбрана, то этот переход будет выбираться в том случае, если условия всех остальных переходов, ведущих из состояния ветвления, не выполняются.

Конечное состояние

Конечное состояние является конечной точкой диаграммы состояния. Когда управление передается в конечное состояние, выполняется действие этого состояния,

и диаграмма состояния завершает свою работу. Из конечного состояния не могут выходить никакие переходы.

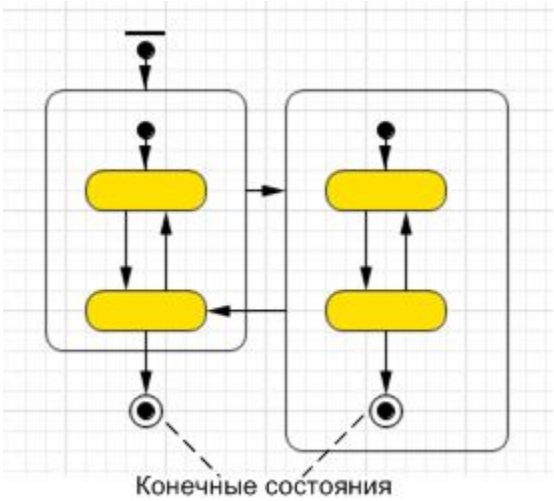

Рисунок 5 – Примеры конечных состояний

Ветвление

Ветвление представляет собой точку разветвления или соединения переходов. С помощью ветвлений Вы можете создать переход, имеющий более одного пункта назначения, или соединить несколько переходов, выполняющих вместе некое общее действие.

Когда управление проходит через состояние-ветвление, выполняется действие этого состояния, и вычисляются дополнительные условия переходов, исходящих из этого состояния. Сработает первый же найденный разрешенный переход – т.е., тот переход, дополнительное условие которого истинно.

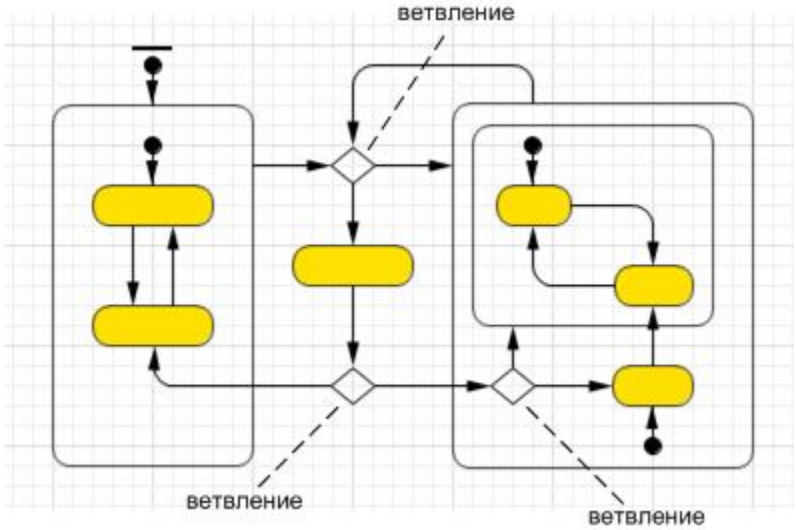

Рисунок 6 – Примеры ветвлений

Основные свойства

Имя – Имя состояния. Имя используется для идентификации состояния и доступа к нему из кода.

Отображать имя – Если опция выбрана, то имя состояния будет отображаться в графическом редакторе.

Исключить – Если опция выбрана, то состояние будет исключено из модели.

На презентации – Если опция выбрана, то состояние будет отображаться на презентации во время выполнения модели.

Действие - Код, выполняемый, когда управление переходит в это состояние.

Ветвление может иметь не более одного выходящего перехода, помеченного как выход из ветвления по умолчанию. Этот переход сработает в том случае, когда все остальные исходящие переходы будут закрыты.

Переходы, ведущие из состояний-ветвлений, имеют следующие свойства, несколько отличные от свойств обычных переходов:

Свойства переходов, ведущих из ветвления:

Условие – Если опция выбрана, то этот переход будет срабатывать, если заданное в поле справа логическое условие будет истинно.

По умолчанию (выбирается, если все остальные условия не выполняются) – Если опция выбрана, то этот переход будет выбираться в том случае, если условия всех остальных переходов, ведущих из состояния ветвления, не выполняются.

Практическая часть

Рассмотрим порядок построения модели на известном примере подготовки к экзамену.

Постановка задачи: Необходимо представить процесс подготовки к экзамену, в котором нужно выучить 17 тем. Среднее время на прочтение одной тему колеблется в диапазоне от 30 до 40 мин. Далее по памяти воспроизводится тема, если материал забывается (эмитировать этот процесс с помощью случайной величины так что, при степени воспроизведения выученного меньше чем на 20%, нужно перечитать материал полностью, если от 20 до 50 %, то прочитать половину материала, если от 50 до 80%, то прочитать треть, иначе перейти к следующей теме). Так повторяется процесс, пока не будут освоены 17 тем.

Порядок выполнения:

Создать новую модель в Anylogic с нуля.

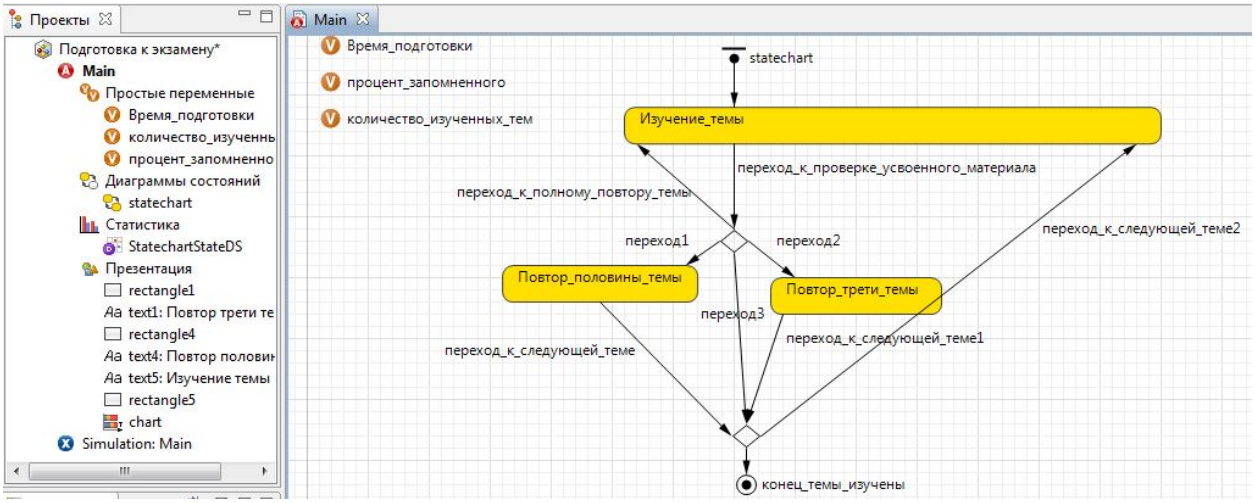

Рисунок 7 – Пример проекта модели

Добавим в проект три простых переменных для хранения времени подготовки к экзамену, количества освоенных тем и временного хранения процента запомненного материала текущей темы.

Таблица 1 – Настройка свойств для проекта на рисунке 7

Название элемента (устанавливается в свойстве *Имя* соответствующего элемента)	**Свойство**	**Значение**
Время_подготовки	Отображать имя	Отметить галочкой

(простая переменная)	Начальное значение	0
	Тип	int
Процент_запомненного (простая переменная)	Отображать имя	Отметить галочкой
	Начальное значение	0
	Тип	int
Количество_изученных_тем (простая переменная)	Отображать имя	Отметить галочкой
	Начальное значение	0
	Тип	int

Добавим элементы диаграммы состояний, соединяя их между собой, как показано на рисунке.

Таблица 2 – Настройка свойств для проекта на рисунке 7

Название элемента (устанавливается в свойстве *Имя* соответствующего элемента)	**Свойство**	**Значение**
Начало_работы (начало диаграммы состояний)	Отображать имя	Отметить галочкой
Изучение_темы (состояние)	Отображать имя	Отметить галочкой
	Действие при выходе	Время_подготовки=Время_подготовки + uniform_discr(30, 40)
переход_к_проверке_усвоенного_материала (переход)	Отображать имя	Отметить галочкой
	Происходит	По таймауту
	По таймауту	4
переход_к_полному_повтору_темы (переход)	Отображать имя	Отметить галочкой
	Происходит	При выполнении условия
	Условие	процент_запомненного<20
ветвление1 (ветвление)	Действие	процент_запомненного=uniform_discr (10, 100)
переход1 (переход)	Отображать имя	Отметить галочкой
	Происходит	При выполнении условия
	Условие	процент_запомненного>=20 & процент_запомненного<50
переход2 (переход)	Отображать имя	Отметить галочкой
	Происходит	При выполнении условия
	Условие	процент_запомненного>=50 & процент_запомненного<80
переход3 (переход)	Отображать имя	Отметить галочкой
	Происходит	При выполнении условия
	По умолчанию	
Повтор_половины_темы (событие)	Отображать имя	Отметить галочкой
	Действие при выходе	Время_подготовки=Время_подготовки + uniform_discr(30, 40)/2
Повтор_трети_темы	Отображать имя	Отметить галочкой

(событие)	Действие при выходе	`Время_подготовки=Время_подготовки + uniform_discr(30, 40)/3`
ветвление2 (ветвление)	Действие	`количество_изученных_тем=количество_изученных_тем+1`
переход_к_следующей_теме (переход)	Отображать имя	`Отметить галочкой`
	Происходит	`По таймауту`
	По таймауту	`2`
переход_к_следующей_теме1 (переход)	Отображать имя	`Отметить галочкой`
	Происходит	`По таймауту`
	По таймауту	`1.3`
переход_к_следующей_теме2 (переход)	Отображать имя	`Отметить галочкой`
	Происходит	`При выполнении условия`
	Условие	`количество_изученных_тем<17`
переход4 (переход)	Отображать имя	`Отметить галочкой`
	Происходит	`При выполнении условия`
	По умолчанию	
	Действие	`процент_запомненного=100`
конец_темы_изучены (конечное состояние)	Отображать имя	`Отметить галочкой`

Задание. Создайте по описанию модель. Проанализируйте действия, которые установлены у элементов Состояния и Переходы, соотнесите их с постановкой задачи.
Задание. Запустите модель на исполнение, проверьте работу. Оцените за несколько запусков модели, сколько в среднем студент будет тратить времени на подготовку экзамены из 17 тем.
Задание. Измените проект модели так, чтобы время, затрачиваемое на изучение темы было в диапазоне от 40 до 60 мин. Выполните несколько запусков нового варианта модели. Сравните общее среднее время подготовки с результатами работы исходной модели.
Задание. Измените проект модели так, чтобы на изучении было 10 тем. Выполните несколько запусков нового варианта модели. Сравните общее среднее время подготовки с результатами работы исходной модели.
Задание. Измените проект модели так, чтобы переход к изучению следующей темы после повтора половины материала происходил с повторной проверкой усвоения материала по тем же условиям, что и при первом ветвлении. Выполните несколько запусков нового варианта модели. Сравните общее среднее время подготовки с результатами работы исходной модели.

Визуализация хода подготовки в изучаемой модели

Анализируя работу модели понятно, что нет достаточной наглядности в процессе подготовки тем. Желательно видеть ход подготовки, а именно, как часто происходит возврат на повторение тем. Для этого используем элемент панели инструментов (палитры) **Статистика**.

Расположим в проекте модели элемент **Временная цветовая диаграмма** и **Набор данных**.

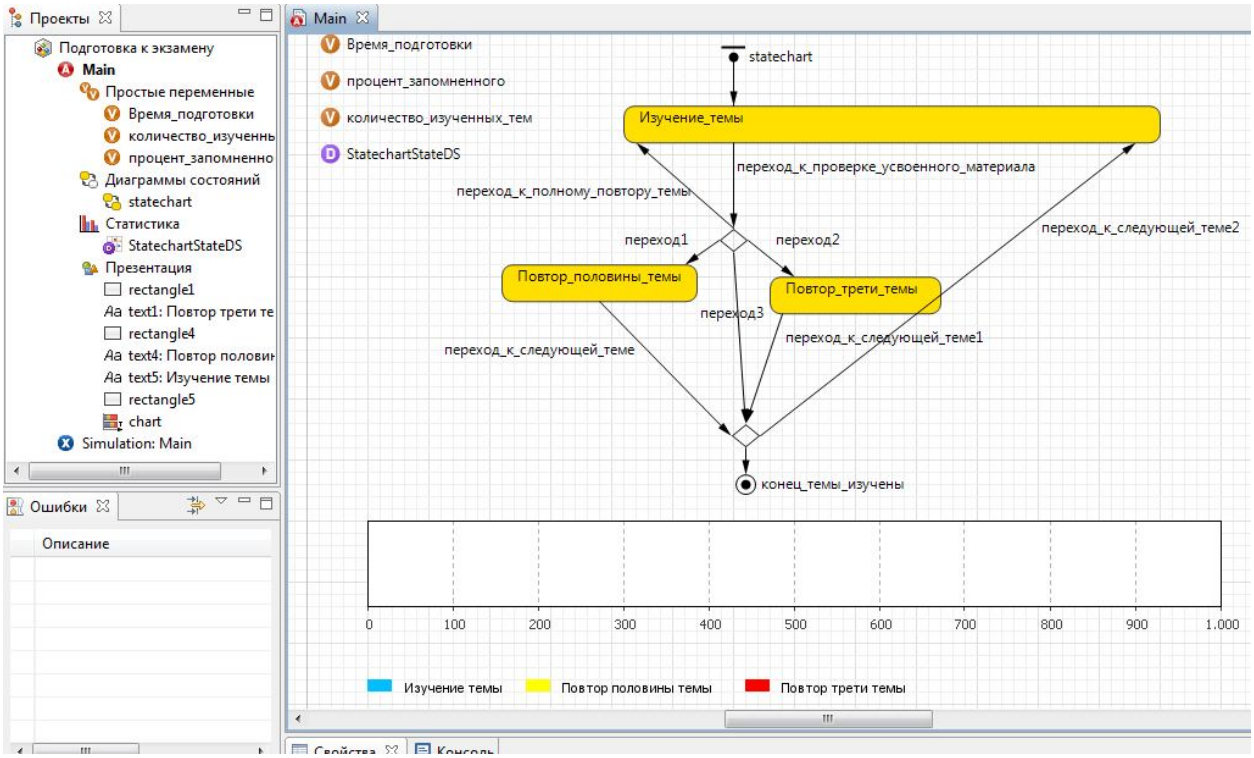

Рисунок 8 – Проект модели с временной диаграммой

Установим свойства согласно таблице 3. Также, **обратите внимание,** что будет изменено одно свойство окна **Main** (класса активного объекта), а также свойства **Событий** диаграммы состояний.

Таблица 3 – Настройка свойств для проекта на рисунке 8

Название элемента (устанавливается в свойстве *Имя* соответствующего элемента)	**Свойство**	**Значение**
chart (временная цветовая диаграмма)	Трижды нажать кнопку **«Добавить цветовое соответствие». В каждом из добавленных соответствий установить пары значений:**	
	Цветовое соответствие	deepSkyBlue (голубой)
	Выражение	value==Изучение_темы
	Цветовое соответствие	yellow (желтый)
	Выражение	value==Повтор_половины_темы
	Цветовое соответствие	red (красный)
	Выражение	value==Повтор_трети_темы
	Основные→Временной диапазон	1000
	Основные →Обновлять автоматически Период	1
StatechartStateDS	Основные	Начало_работы.getActiveSimpleState

(набор данных)	→Значение по оси Y	()
	Основные →Обновлять автоматически Период	1

Таблица 4 – Настройка свойств ранее созданных элементов для проекта на рисунке 8

Название элемента (устанавливается в свойстве *Имя* соответствующего элемента)	**Свойство**	**Значение**
Main (класс активного)	Основные →Действие при запуске	chart.addDataSet(StatechartStateDS);
Изучение_темы (состояние)	Действие при входе	StatechartStateDS.add(time(), Изучение_темы)
Повтор_половины_темы (событие)	Действие при входе	StatechartStateDS.add(time(), Повтор_половины_темы);
Повтор_трети_темы (событие)	Действие при входе	StatechartStateDS.add(time(), Повтор_трети_темы);

Для облегчения понимаемости диаграммы необходимо отобразить какие цвета какие состояния отражают. Для этого под диаграммой в ряд расположите 3 прямоугольника и 3 метки со следующими свойствами, описанными в таблице 5, и показанными на рисунке 8.

Таблица 5 – Настройка свойств ранее созданных элементов для проекта на рисунке 8

Название элемента (устанавливается в свойстве *Имя* соответствующего элемента)	**Свойство**	**Значение**
Rectangle1 (прямоугольник)	Цвет заливки	deepSkyBlue
Text1 (текст)	Текст	Изучение темы
Rectangle2 (прямоугольник)	Цвет заливки	yellow
Text2 (текст)	Текст	Повтор половины темы
Rectangle3 (прямоугольник)5	Цвет заливки	red
Text3 (текст)	Текст	Повтор трети темы

Задание. Измените по описанию модель. Проанализируйте действия, которые установлены у элементов **Временная диаграмма** и **Набор данных**, обратите внимание на то, как связаны элементы **Диаграммы состояний** с указанными элементами.

Задание. Запустите модель на исполнение, проверьте работу. Оцените работу диаграммы. Модифицируйте значения таймаутов у элементов **Переход_к_следующей_теме**, **Переход_к_следующей_теме1** и **Переход_к_проверке_усвоенного_материала**. Проверьте, как отобразятся эти изменения на диаграмме. Подумайте, почему?

Постановка задачи 2: Создать конечный автомат, который бы включал/отключал один из 4-х сигнальных кругов с заданным значением вероятности и с заданным условием остановки работы, когда будет выполнено 5 переключений.

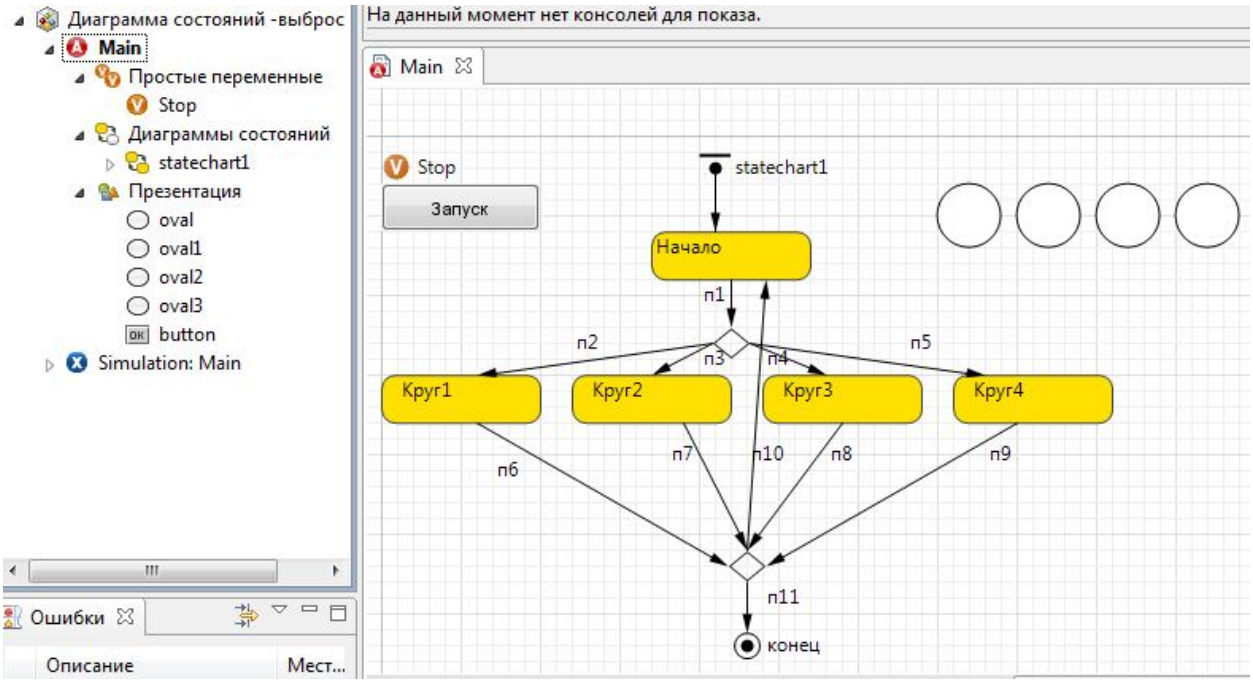

Рисунок 9 – Проект модели конечного автомата

Таблица 6 – Настройка свойств добавляемых элементов для проекта на рисунке 9

Название элемента (устанавливается в свойстве *Имя* соответствующего элемента)	**Свойство**	**Значение**
Oval (овал)	Имя	Oval
Oval1 (овал)	Имя	Oval1
Oval2 (овал)	Имя	Oval2
Oval3 (овал)	Имя	Oval3
Stop (простая переменная)	Тип	`int`
	Начальное значение	`0`
Button (кнопка)	Метка	`Запуск`
	Действие	`Stop=0;` `statechart1.start();`
statechart1 (начало диаграммы	Отображать имя	`Отметить галочкой`

состояний)		
Начало (состояние)	Отображать имя	Отметить галочкой

	Действие при входе	oval.setFillColor(new Color(0,255,0));
		oval1.setFillColor(new Color(0,255,0));
		oval2.setFillColor(new Color(0,255,0));
		oval3.setFillColor(new Color(0,255,0));
		oval.setVisible(true);
		oval1.setVisible(true);
		oval2.setVisible(true);
		oval3.setVisible(true);

п1 (переход)	Отображать имя	Отметить галочкой
	Происходит	По таймауту
	По таймауту	1
ветвление1 (ветвление)	Имя	ветвление1
п2 (переход)	Отображать имя	Отметить галочкой
	Условие	randomTrue(0.3)
П3 (переход)	Отображать имя	Отметить галочкой
	Условие	randomTrue(0.5)
П4 (переход)	Отображать имя	Отметить галочкой
	Условие	randomTrue(0.8)
П5 (переход)	Отображать имя	Отметить галочкой
	По умолчанию	
Круг1 (состояние)	Отображать имя	Отметить галочкой
	Действие при входе	oval.setFillColor(new Color(255,0,0));
Круг2 (состояние)	Отображать имя	Отметить галочкой
	Действие при входе	Oval1.setFillColor(new Color(255,0,0));
Круг3 (состояние)	Отображать имя	Отметить галочкой
	Действие при входе	Oval2.setFillColor(new Color(255,0,0));
Круг4 (состояние)	Отображать имя	Отметить галочкой
	Действие при входе	Oval3.setFillColor(new Color(255,0,0));
П6 (переход)	Отображать имя	Отметить галочкой
	Происходит	По таймауту
	По таймауту	0.1
	Действие	Stop=Stop+1
П7 (переход)	Отображать имя	Отметить галочкой
	Происходит	По таймауту
	По таймауту	0.1
	Действие	Stop=Stop+1
П8 (переход)	Отображать имя	Отметить галочкой
	Происходит	По таймауту
	По таймауту	0.1
	Действие	Stop=Stop+1
П9 (переход)	Отображать имя	Отметить галочкой
	Происходит	По таймауту
	По таймауту	0.1
	Действие	Stop=Stop+1
П10	Отображать имя	Отметить галочкой

(переход)	По умолчанию	
ветвление2 (ветвление)	Имя	`ветвление2`
П11 (переход)	Отображать имя	`Отметить галочкой`
	Условие	`Stop==5`
Конец (конечное состояние)	Отображать имя	`Отметить галочкой`

Задание: Модифицировать пример:

1) добавить гистограмму с четырьмя столбцами, каждый из которых будет накапливать статистику по исходам работы диаграммы (например, первый столбец будет показывать сколько раз работа закончилась окрашиванием в красный 1-го круга).
2) Запустить модель 100 раз, проанализировать результат статистики.
3) Проварьировать параметры переходов (в свойствах параметр Условие) к блокам Круг1…Круг4 так, чтобы добиться равномерного распределения статистики.

Порядок выполнения действий элементов диаграммы состояний

Очень важно точно знать, в каком именно порядке выполняются действия элементов диаграммы состояний. Для этого мы предлагаем Вам изучить приведенный ниже алгоритм.

При срабатывании перехода выполняются следующие действия (в указанном порядке):

1. Действия при выходе из состояния, начиная с текущего простого состояния, и дальше вверх по иерархии состояний, заканчивая тем сложным состоянием, на уровне иерархии которого и передается управление.
2. Действие перехода.
3. Действия при входе в состояние, начиная со сложного состояния, которое получает управление, и дальше, вниз по иерархии состояний, вплоть до простого состояния или псевдосостояния, в которое передается управление.
4. Если управление передается в псевдосостояние, то выполняется код действия псевдосостояния, а затем управление немедленно передается другому состоянию, и описанный выше алгоритм выполняется сначала.

Действия состояний и переходов выполняются за нулевое модельное время. Поэтому они не могут содержать синхронизационных операций и операций задержки и не могут вызывать методы, явно или неявно содержащие такие операции.

Пример

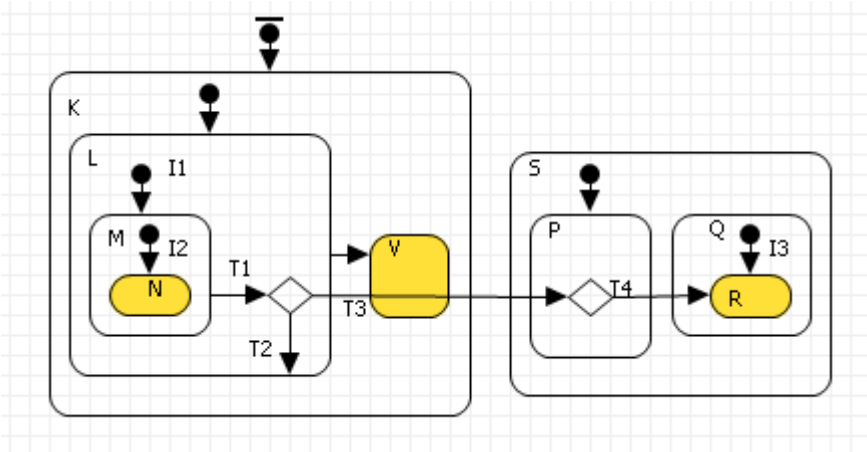

Рисунок А.1 - Пример порядка выполнения действий

Давайте рассмотрим пример, приведенный на рисунке. Предположим, что состояние N является текущим простым состоянием, и срабатывает переход T1. Тогда действия выполняются в следующем порядке:

1. Действие при выходе из состояния N
2. Действие при выходе из состояния M
3. Действие перехода T1
4. Действие состояния ветвления

Затем, в зависимости от дополнительных условий переходов, будет выбран переход T2 или T3. Если будет выбран переход T2, то выполнятся следующие действия:

5. Действие перехода T2
6. Действие указателя начального состояния I1 (действия при входе и выходе из состояния L не выполняются, поскольку управление остается в этом состоянии)
7. Действие при входе в состояние M
8. Действие указателя начального состояния I2
9. Действие при входе в состояние N

Если выбирается переход T3, то выполняются следующие действия:

10. Действие при выходе из состояния L
11. Действие при выходе из состояния K (действия состояния V не выполняются)

12. Действие перехода T3
13. Действие при входе в состояние S
14. Действие при входе в состояние P
15. Действие состояния-ветвления
16. Действие при выходе из состояния P
17. Действие перехода T4 (дополнительное условие перехода должно быть равно true, поскольку это единственный выход из ветвления)
18. Действие при входе в состояние Q
19. Действие указателя начального состояния I3
20. Действие при входе в состояние R