

## **ЗАДАНИЯ 5 – 6**

# **СТЕЙТЧАРТЫ: МОДЕЛЬ ПЕШЕХОДНОГО ПЕРЕХОДА**

### **ЦЕЛИ ЗАНЯТИЯ**

В результате построения этой модели будут рассмотрены следующие новые вопросы:

- построение стейтчартов;
- действия при входе и выходе из состояния, иерархические состояния;
- переход по исчерпанию таймаута;
- переход при получении сообщения;

### **ФОРМА ОРГАНИЗАЦИИ ЗАНЯТИЯ**

Фронтальная.

### **СТУДЕНТ ДОЛЖЕН ЗНАТЬ**

- понятия: сообщение, таймаут, переход,
- состояния в стейтчартах и их виды,
- основы алгоритмического языка Java,
- интерфейс программы AnyLogic.

### **СТУДЕНТ ДОЛЖЕН УМЕТЬ**

- выполнять лабораторно-практическое задание №4,
- создавать модели в программе AnyLogic,
- использовать стейтчарты.

### **ОБЕСПЕЧЕННОСТЬ**

- компьютер с установленной программой AnyLogic версии 6,
- настоящий курс лабораторно-практических работ.

## **ПРАКТИЧЕСКОЕ ЗАДАНИЕ**

Построим модель регулируемого пешеходного перехода со светофором, разрешающим или запрещающим движение транспорта.

### **5.1. ОПИСАНИЕ ПРОБЛЕМЫ**

Светофор, регулирующий движение автомобилей на пешеходном переходе, может находиться в следующих состояниях: движение транспорта разрешено (зеленый), приготовиться к запрещающему сигналу (мигающий зеленый), приготовиться к остановке (желтый), движение запрещено (красный) и приготовиться к движению (красный и желтый).

Светофор работает в автоматическом режиме. В каждом состоянии светофор находится определенный постоянный период времени.

## 5.2. ПОСТРОЕНИЕ МОДЕЛИ

Создайте новый проект под названием Svetofor и назовите класс корневого активного объекта Model.

Наша модель будет иметь только один активный объект, представляющий светофор, поэтому корневой объект Model будет единственным активным объектом нашей модели. На диаграмму класса активного объекта Model поместите **Начало диаграммы состояний**

из панели **Диаграмма состояний** и назовите ее Для\_автомобилей, заметьте, что AnyLogic может работать с элементами, набранными кириллицей. Перетащите мышью элемент **Состояние** под стрелочку начала диаграммы как показано на рис. 5.1.

Рис. 5.1

Имя состояния, как и все другие его параметры, можно редактировать в окне его свойств.

Для того чтобы построить стейтchart, следует использовать элементы из палитры **Диаграмма состояний**, рис. 5.2.

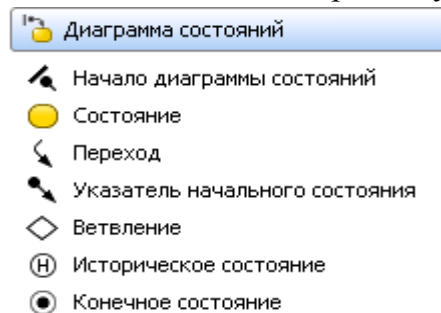


Рис. 5.2

1. определяет начальное состояние всего стейтчарта,
2. с помощью кнопки рисуются состояния (как простые, так и гиперсостояния),
3. используется для рисования переходов между состояниями,
4. определяет начальное состояние внутри сложного состояния,
5. используется для рисования состояния, являющегося "финальным" в поведении активного объекта.

Заметьте, что для любого выделенного объекта внизу появляется панель его свойств, в котором можно изменить параметры и, в частности, имя объекта, если это необходимо. Структурные ошибки при рисовании стейтчарта - повисшие переходы, дублированные указатели начального состояния и т. п. - выделяются в панели **Проекты** значком X красного цвета и записью в панели **Ошибки**. Выделенные переходы должны иметь на концах зеленые точки, если точки белые, это значит, что переход не соединен с состоянием – висит.

В соответствии с алгоритмом работы светофора помимо начального состояния в модель нужно ввести дополнительные состояния (рис. 5.3). Начальное состояние назовите движение – движение автомобилям разрешено (зеленый свет), затем светофор переходит в состояние внимание – внимание (мигающий зеленый), медленно – приготовиться к остановке (желтый свет), остановка транспорта stop – запрет движения (красный свет) и приготовиться – приготовиться к движению (красный и желтый свет горят одновременно).

Состояние внимание представим гиперсостоянием с парой переключающихся элементарных состояний: в одном из них зеленый горит (состояние А), в другом – нет (состояние В). Для построения гиперсостояния сначала создайте обычное состояние, увеличьте его (растянув мышью) и поместите внутрь другое состояние. Постройте эти

состояния и соедините их переходами, как показано на рис. 5.3.

Зададим условия срабатывания переходов. Переходы в нашем автоматическом светофоре выполняются по таймауту, т. е. по истечении интервала времени, который прошел с момента прихода системы в данное состояние.



1. В состоянии движение светофор находится 10 секунд,
2. затем 7 секунд зеленый сигнал мигает,
3. в состоянии медленно 4 секунды горит желтый,
4. в течение 10 секунд движение запрещено и
5. 4 секунды светофор находится в состоянии приготовиться.

В нашей модели единица модельного времени соответствует 1 секунде реального времени.

Для задания условий срабатывания переходов, выделите переход t1, и в поле **Происходит** оставьте без изменения вариант **По таймауту**, а в поле **По таймауту** введите 10 (рис. 5.4).

Аналогично задайте условия срабатывания других переходов. Между состояниями А и В переходы должны срабатывать через 1 секунду.

Рис. 5.3

Запустите модель. Активное в данный момент состояние подсвечивается красным. Переход, ожидающий истечения таймута подсвечивается синим. Проведите эксперименты с моделью при различных масштабах времени.

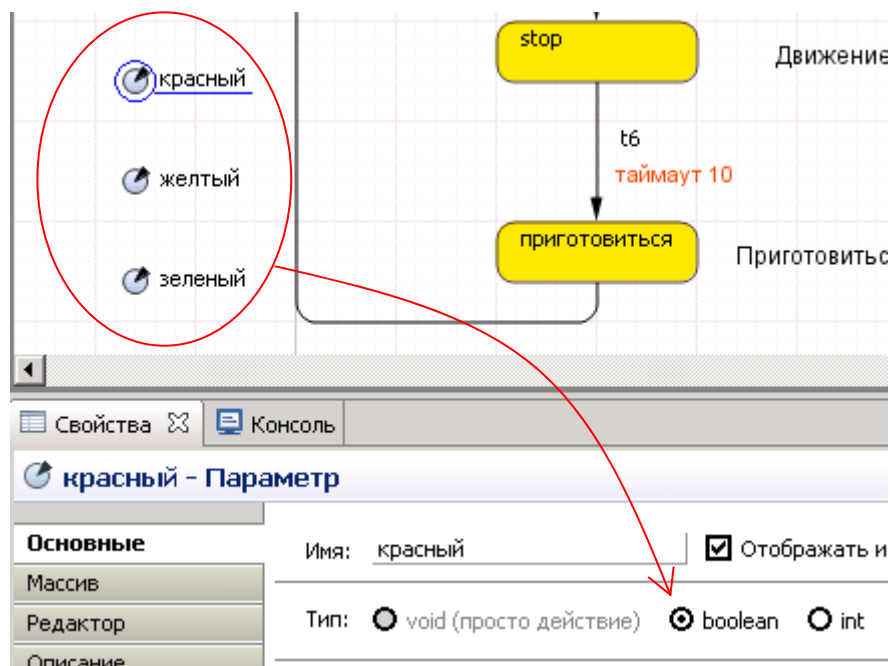
t1 - Переход	
Основные	Имя: t1 <input checked="" type="checkbox"/> Отображать имя
Описание	Происходит: По таймауту
	По таймауту: 10

Рис. 5.4

В каждом состоянии светофора должен гореть определенный сигнал: в состоянии движение должен гореть зеленый, в состоянии приготовиться должны гореть красный и желтый одновременно и т. п. Перейдите на диаграмму класса активного объекта Model. Создайте три параметра логического типа: красный, желтый и зеленый, которые будут

принимать истинное значение тогда, когда у светофора горит соответствующий сигнал: красный, желтый или зеленый (рис. 5.5). Начальные значения этих булевых параметров

можно не задавать: по умолчанию они будут равны false.



Мы создали стейтchart именно для управления значениями этих параметров, каждое состояние отвечает за зажигание своего света или их комбинации. Например, в состоянии медленно должен гореть желтый, в состоянии stop должен загореться красный свет, а в состоянии приготовиться должны гореть красный и желтый одновременно.

Запрограммируем эти действия.

Рис. 5.5

1. В свойствах состояния движение в поле **Действие при входе** запишите `зеленый=true;`, а в поле **Действие при выходе** запишите `зеленый = false;` (рис. 5.6).
2. То же самое запишите для состояния В гиперсостояния внимание, а у состояния А эти поля нужно оставить пустыми – когда светофор находится в этом состоянии, он не горит.

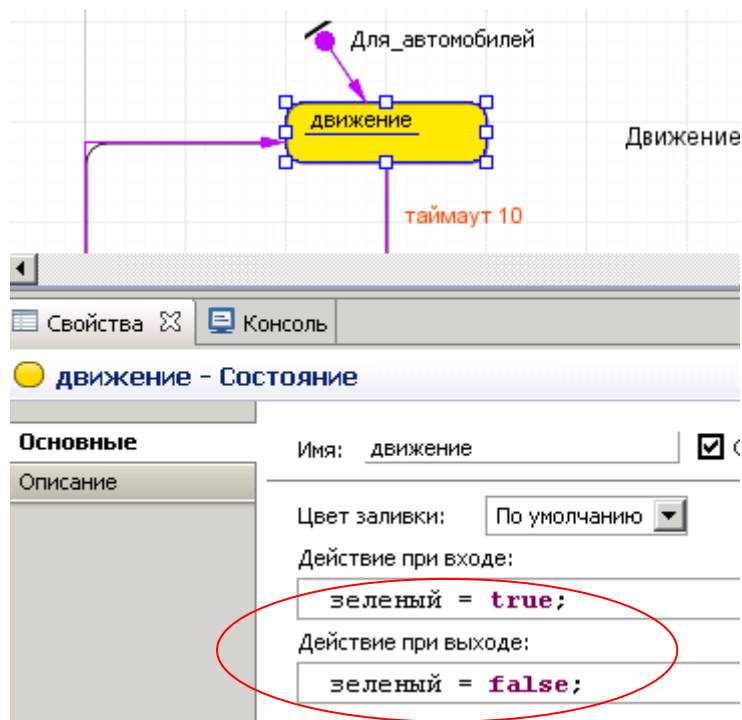


Рис. 5.6

3. Аналогично, в состоянии медленно нужно включить желтый сигнал, т. е. при входе в это состояние установить параметр желтый в true, а при выходе из этого состояния установить его в false.
4. Для состояния stop аналогично опишите состояния параметра красный,
5. а для состояния приготовиться оба параметра красный и желтый нужно установить в true при входе, и в false при выходе.

Запустите модель на выполнение. Вы увидите, что параметры зеленый, желтый и

красный будут переключаться между значениями истина и ложь в соответствии с алгоритмом переключения светофора.

### 5.3. ПРЕЗЕНТАЦИЯ МОДЕЛИ

Презентация модели рисуется в той же диаграмме (в графическом редакторе), в которой задается и диаграмма моделируемого процесса, (рис. 5.7). Графические объекты цвета сигналов светофора в презентации имеют динамические параметры, все остальные – статические. Проезжую часть удобно нарисовать с помощью графических примитивов типа **Прямоугольник**, а светофор с помощью примитива - **Ломаная**. Сигнальные элементы светофора строятся из трех овалов, повернутых на 45 градусов (поле **Поворот** вкладки **Дополнительные** окна свойств овала).

Установим динамическое значение цвета верхнего сигнала светофора: если переменная красный истинна, то цвет должен быть red (красный), в противном случае его цвет нужно установить gray (серый). Это записывается следующим условным выражением на языке Java:

```
красный? red: gray
```

Цвет среднего и нижнего овалов, следует установить в поле их динамических значений соответственно так:

```
желтый? yellow: gray
```

```
зеленый? green: gray
```

red, yellow, green и gray – предопределенные константы, обозначающие стандартные цвета.

Запустите модель и проверьте ее работу.

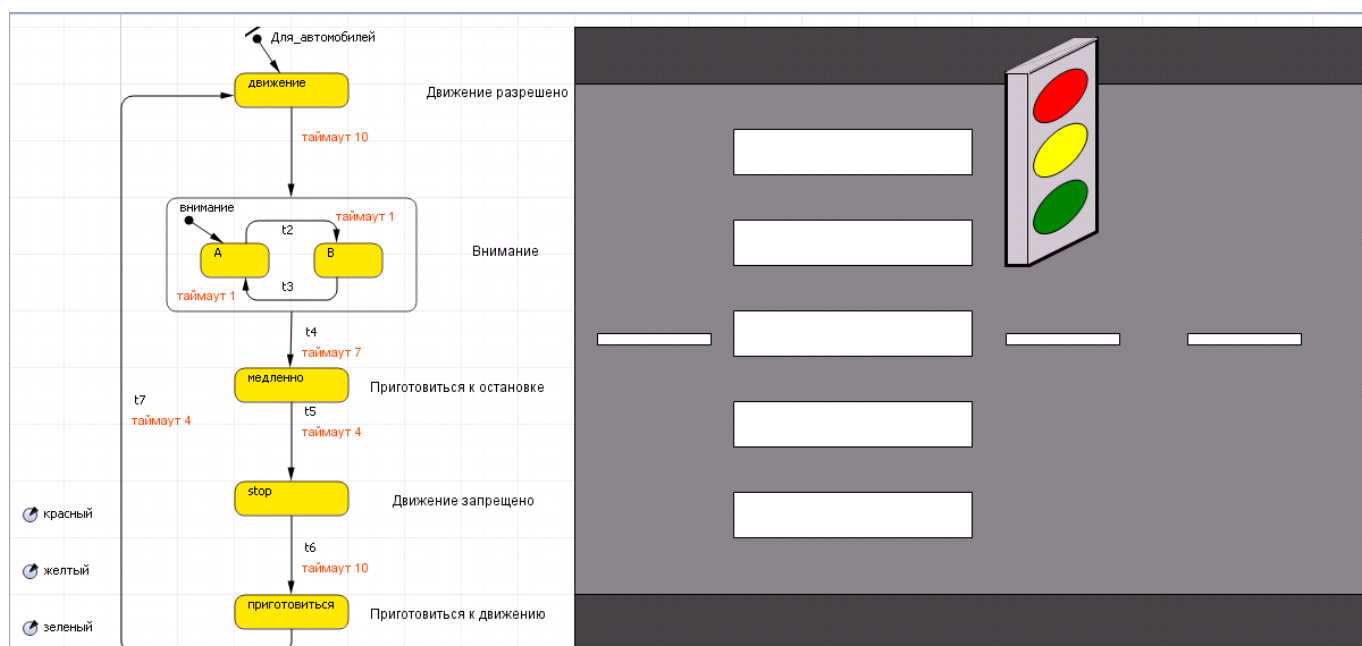


Рис. 5.7

## 5.4. СРАБАТЫВАНИЕ ПЕРЕХОДА ПО СИГНАЛУ

Добавим к нашей модели второй светофор, для пешеходов. Он будет иметь два сигнала, зеленый и красный, и три состояния: идите (зеленый), внимание (мигающий зеленый) и стойте (красный). Добавим в модель два булевских параметра стойте и идите, их значениями будет управлять второй стейтchart – для пешеходов. Создадим этот стейтchart на той же диаграмме класса Model, назвав его Для\_пешеходов, (рис. 5.8).

Поскольку управление светофором пешеходов похоже на управление светофором автомобилей, новый стейтchart можно построить копированием и изменением уже построенного стейтcharta Для\_автомобилей. Выделите 3 состояния стейтcharta Для\_автомобилей и вставьте их в другое место диаграммы. Эти элементы скопируются, и им автоматически будут даны уникальные имена, чтобы не было конфликта имен. Переименуйте состояния стейтcharta Для\_пешеходов, дорисуйте недостающий переход t12 и перенесите начало диаграммы на состояние Стойте, как показано на рис. 5.8.

Измените параметры в полях Действие при входе и Действие при выходе в свойствах состояний стейтcharta Для\_пешеходов. Теперь наш стейтchart должен управлять параметрами идите и стойте, которые, в свою очередь, будут управлять зажиганием света именно пешеходного светофора.

Рис. 5.8

Настроим условия срабатывания переходов стейтchartов между состояниями. Для обеспечения безопасной работы пешеходного перехода необходимо синхронизировать срабатывания стейтchartов так, чтобы всегда, когда светофор пешеходов находится в состояниях Идите или Мигает, светофор автомобилей обязательно находился бы в состоянии stop. Для этого можно подобрать подходящие таймауты срабатывания переходов стейтcharta, но при каждом изменении модели, придется эти таймауты подбирать снова и снова. Более разумно синхронизировать стейтchartы, посылая специальные разрешающие сигналы из одного стейтcharta в другой.

В нашей модели стейтchartы будут обмениваться следующими сигналами: АВТОМОБИЛИ и ПЕШЕХОДЫ. В стейтcharte Для\_пешеходов переход t12 будет срабатывать когда получен сигнал ПЕШЕХОДЫ, который будет генерироваться в стейтcharte Для\_автомобилей при переходе t5 в состояние stop. В свою очередь, в стейтcharte Для\_автомобилей переход t6 будет срабатывать когда получен сигнал АВТОМОБИЛИ, который генерируется в стейтcharte Для\_пешеходов при переходе t11 в состояние Стойте, рис. 5.9.

В AnyLogic есть несколько способов передачи сообщения в диаграмму состояний. В нашей модели мы будем использовать метод `fireEvent( )`, который должен вызываться в том стейтcharte, которому предназначено сообщение. То есть, если из некоего объекта мы хотим послать сообщение стейтchartу, то нужно в этом объекте написать команду:



стейтchart.fireEvent(сообщение)

Поэтому, в поле Действие перехода t5 стейтchartа Для\_автомобилей нужно вставить команду:

Для\_пешеходов.fireEvent("ПЕШЕХОДЫ"),

в такое же поле перехода t11 стейтchartа Для\_пешеходов вставьте команду:

Для\_автомобилей.fireEvent("АВТОМОБИЛИ")

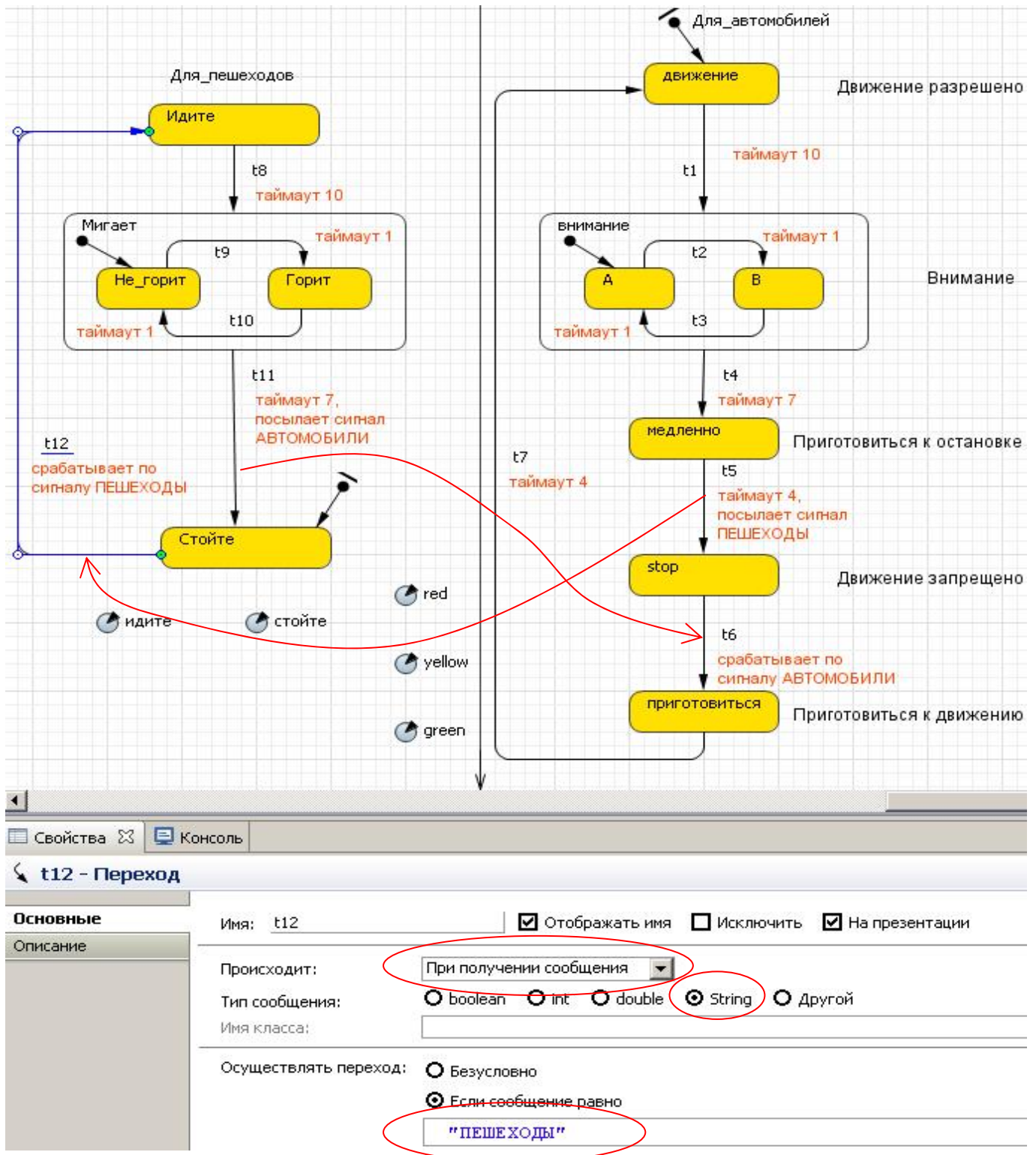


Рис. 5.9

Таким образом, каждый из светофоров будет информировать другого о своем переходе в состояние запрещения движения, как пешеходов, так и автомобилей.

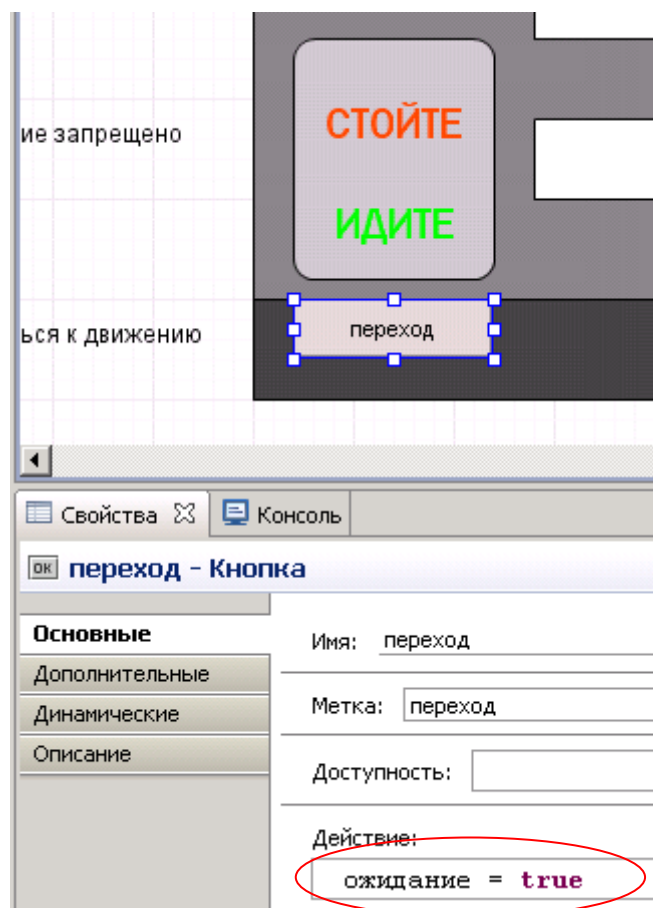
Для срабатывания перехода стейтчарта при получении нужного сообщения, в стейтчарте Для\_пешеходов в поле **Происходит** окна свойств перехода t12 выберите вариант **При получении сообщения**, укажите тип сообщения String, а в поле **Осуществлять переход** выберите Если сообщение равно и введите "ПЕШЕХОДЫ", рис. 5.9.

Аналогично, для срабатывания перехода автомобильного стейтчарта по сигналу от пешеходного стейтчарта в стейтчарте Для\_автомобилей в поле **Происходит** окна свойств перехода t6 выберите вариант **При получении сообщения**, укажите тип сообщения String, а в поле **Осуществлять переход** выберите Если сообщение равно и введите "АВТОМОБИЛИ".

Остальные переходы этих стейтчартов будут срабатывать по таймаутам, как и прежде. Проверьте по рисунку 5.9 установленные параметры переходов стейтчартов. Запустите модель на выполнение.

На презентации модели, в дополнение к светофору для автомобилей, следует нарисовать светофор для пешеходов с двумя сигналами – красной надписью СТОЙТЕ и зеленой ИДИТЕ. Динамикой цвета этих надписей будут управлять логические параметры стоите и идите, которые нужно создать на диаграмме по аналогии с параметрами красный, желтый и зеленый. Продемонстрируйте построенную модель преподавателю.

## 5.5. СРАБАТЫВАНИЕ ПЕРЕХОДА ПО УСЛОВИЮ



Если пешеходы переходят улицу достаточно редко, то автоматическое переключение светофора будет неоправданно часто прерывать движение автомобилей. В этом случае разумно установить специальную кнопку, при нажатии которой пешеходом, светофор автомобилей остановит на некоторое время движение автомашин. Для этого перетащите мышью элемент **Кнопка** с палитры **Элементы управления** на диаграмму класса активного объекта Model рядом с пешеходным светофором, рис. 5.10. Назовите кнопку и ее метку: **переход**.

Определите действие при нажатии на кнопку так, чтобы устанавливался в истину булевый параметр ожидание. Для этого нужно записать в поле **Действие** окна свойств этой кнопки команду: `ожидание = true`.

Рис. 5.10

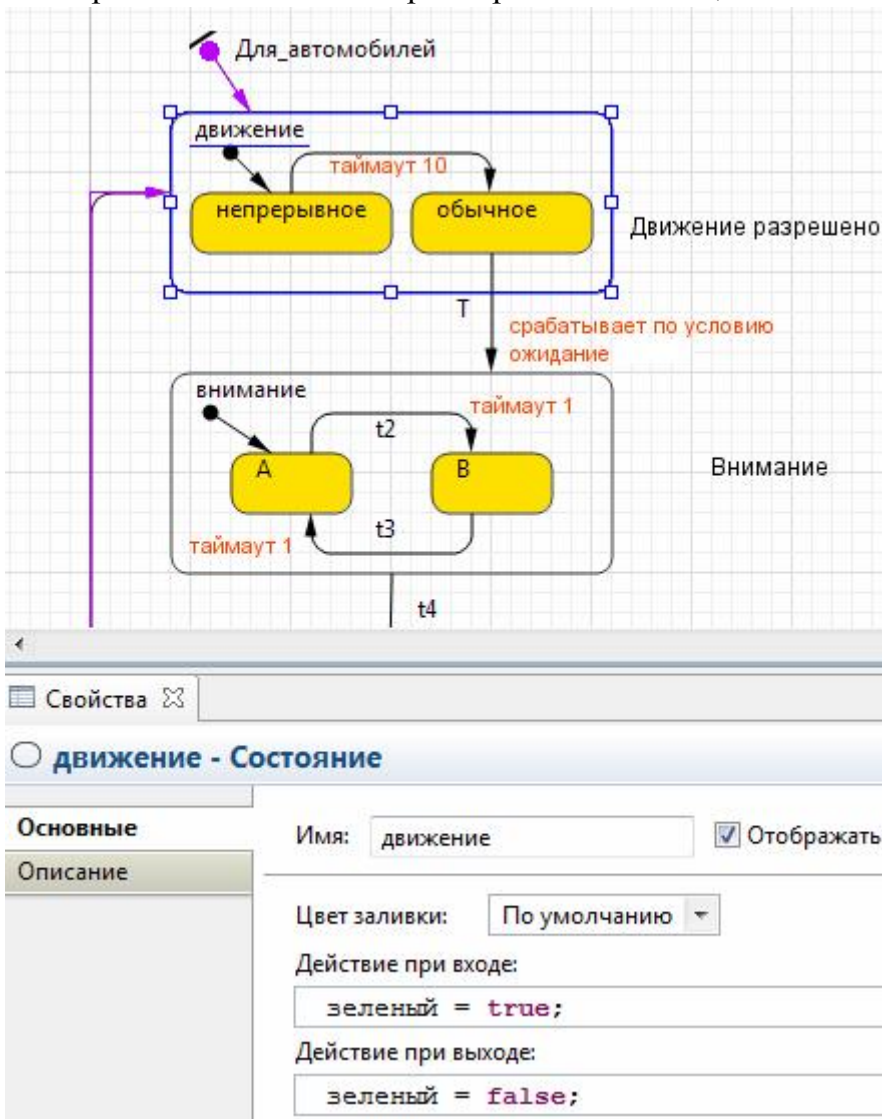


Логический параметр `ожидание`, конечно же, нужно ввести на диаграмме активного объекта `Model` с начальным значением `false`. Этот параметр будет определять, собирается ли пешеход перейти дорогу. Значение этого параметра будем переводить в `false` каждый раз, как только пешеходный светофор перейдет в состояние `Мигает`. Для этого нужно записать в поле **Действие при входе** окна свойств состояния `Мигает` стейтчарта `Для_пешеходов` команду: `ожидание=false`.

Таким образом, при нажатии кнопки **переход**, стейтchart автомобилей «узнает» об ожидающих на переходе пешеходах и переключится из состояния `Движение` в состояние `Внимание` и затем в состояние запрещения движения автомобилей.

В случае, если некий злоумышленник будет постоянно нажимать кнопку **переход**, это может полностью парализовать движение автомобилей. Для того чтобы этого не произошло, в стейтchartе `Для_автомобилей` сделайте иерархическим состояние `Движение`, с двумя простыми состояниями непрерывное и обычное, как показано на рис. 5.11.

В состоянии `непрерывное` автомобили будут двигаться до истечения таймута 10 секунд, невзирая на состояние параметра `ожидание`, т.е. автомобилям будет гарантированно





предоставлено некоторое время для движения, даже если кнопка **переход** будет всегда нажата. После истечения таймута 10 секунд светофор перейдет в обычное состояние движения, которое может прерываться. Переход `T` из состояния `обычное` сработает, когда будет нажата кнопка **переход**, т.е. параметр `ожидание` будет истинен.

Если кнопка **переход** не нажата (т. е. параметр `ожидание` имеет значение `false`), автомобили будут продолжать движение до нажатия этой кнопки. Если нажать кнопку один или более раз, то параметр `ожидание` станет истинным и автомобильный светофор из состояния `обычное` перейдет в состояние `внимание` и затем остановит движение автомобилей.

Рис. 5.11

Проверьте работу модели и продемонстрируйте ее преподавателю.

## 5.6. КОНТРОЛЬНЫЕ ЗАДАНИЯ

1. Измените модель светофора с кнопкой таким образом, чтобы моделировалось автоматическое нажатие кнопки **ПЕРЕХОД** в случайные моменты времени в интервале от 5 секунд до 1 минуты. (5+)
2. Измените модель автоматического светофора таким образом, чтобы по кнопке **НОЧЬ** включался режим мигания желтого света, а по кнопке **ДЕНЬ** светофор переходил в нормальный режим работы. (5+)
3. Измените модель светофора с кнопкой таким образом, чтобы по кнопке **ВЫКЛ** светофор выключался (все серое), а по кнопке **ВКЛ** светофор переходил в нормальный режим работы. (5)
4. Измените презентацию модели таким образом, чтобы рядом со светофором высвечивалось время в секундах, оставшееся до смены сигнала. (5)
5. Измените презентацию модели таким образом, чтобы рядом со светофором высвечивалось время в секундах, прошедшее со смены сигнала. (5)
6. Измените модель автоматического светофора таким образом, чтобы по кнопке **СТОП** включался красный свет у всех светофоров, а по кнопке **ДВИЖЕНИЕ** светофор переходил в нормальный режим работы. (5)
7. Измените модель таким образом, чтобы регулировалось движение двух пересекающихся дорожных потоков в автоматическом режиме (без кнопки), т.е. модель двух трехцветных светофоров со следующей последовательностью сигналов: зеленый – зеленый мигающий – желтый – красный – (красный + желтый) – зеленый. (4)
8. Измените модель светофора с кнопкой таким образом, чтобы пешеход видел, нажата уже кнопка или нет, например, с помощью надписи. (4)
9. Измените модель таким образом, чтобы моделировать железнодорожный переезд: по кнопке **ПОЕЗД** включается сигнал и через 5 сек. начинает мигать сдвоенный красный светофор (попеременно правый – левый), еще через 5 сек. опускается шлагбаум. По истечении случайного промежутка времени в диапазоне от 30 до 60 сек. (прохождение поезда), переезд переводится в исходное состояние. (4)
10. Измените модель таким образом, чтобы 3-х цветный светофор переключался в полуавтоматическом режиме по нажатию кнопок регулировщиком: по кнопке **КРАСНЫЙ** начинает мигать зеленый свет, затем зажигается желтый и через несколько сек. – красный; по кнопке **ЗЕЛЕНый** зажигается одновременно красный и желтый, затем горит зеленый. (4)
11. Измените модель таким образом, чтобы частота мигания зеленого света регулировалась с помощью слайдера. (3)
12. Измените модель автоматического светофора автомобилей и пешеходов (без кнопки), удалив переходы по сигналу и подберите подходящие таймауты для согласованной работы светофоров. (3)
13. Измените модель светофора с кнопкой таким образом, чтобы с помощью слайдера можно было регулировать время, через которое повторное нажатие кнопки **ПЕРЕХОД** приведет к переключению светофора автомобилей в режим ожидания. (3)
14. Как передать сигнал от стейтчарта СТ\_1 стейтчарту СТ\_2 ? (3)
15. Измените модель таким образом, чтобы время, в течение которого разрешено движение автомобилей, было 15 секунд и перед окончанием запрещающего сигнала автомобилям мигал красный свет. (3)
16. В чем отличие  от  ? (3)
17. Измените модель таким образом, чтобы время, в течение которого разрешено

движение пешеходов, было 10 секунд и перед окончанием запрещающего сигнала пешеходам мигал красный свет. (3)

18. Как в модели реализована защита от чрезмерно-частого нажатия кнопки **ПЕРЕХОД**? (3)

19. Измените модель таким образом, чтобы зеленый сигнал светофора не мигал. (3)