ЗАДАНИЕ 4

ДИСКРЕТНО-СОБЫТИЙНАЯ МОДЕЛЬ СЧЕТЧИКА

ЦЕЛИ ЗАНЯТИЯ

В результате построения этой модели будут рассмотрены следующие новые вопросы:

- Создание нового класса активного объекта.
- События.
- Значки активного объекта.
- Порты и сообщения;
- Действия при получении сообщений.

ФОРМА ОРГАНИЗАЦИИ ЗАНЯТИЯ

Фронтальная.

СТУДЕНТ ДОЛЖЕН ЗНАТЬ

- понятия: проект, активный объект, переменная, параметр, презентация, эксперимент,
- основы алгоритмического языка Java,
- интерфейс программы AnyLogic.

СТУДЕНТ ДОЛЖЕН УМЕТЬ

- выполнять лабораторно-практическое задание №3,
- создавать модели в программе AnyLogic,

ОБЕСПЕЧЕННОСТЬ

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

ПРАКТИЧЕСКОЕ ЗАДАНИЕ

В данной работе мы построим дискретно-событийную модель средствами AnyLogic.

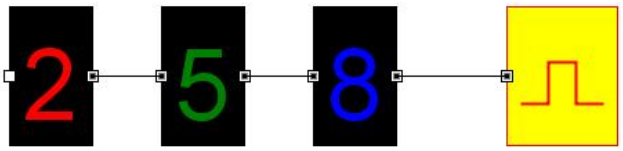
Системы называются дискретно-событийными, если изменения переменных состояния в них происходят только в явно определенные моменты времени или под влиянием явно определенных событий. Находясь в некотором состоянии, дискретная система сохраняет его до наступления очередного события, под воздействием которого переменные системы и, следовательно, ее состояние изменяются скачком. Например, при построении модели банка состояние системы может быть представлено количеством клиентов в помещении банка и числом занятых кассиров. Состояние системы изменяется, если новый клиент входит в банк или освобождается кассир, а это условно можно считать мгновенными событиями.

4.1. ДИСКРЕТНАЯ МОДЕЛЬ СЧЕТЧИКА

В счетчике генератор посылает устройству отображения какой-то сигнал ("тик"). Каждый разряд десятичного счетчика считает число пришедших на его вход "тиков" по модулю 10 и передает на свой выход сигнал переполнения после прихода на его вход каждого десятого сигнала.

При построении модели такого счетчика нужно использовать средства, характерные для моделей дискретно-событийных систем. Здесь нам достаточно трех таких средств: события, порта и передаваемых через порт сообщений.

4.2. ПОСТАНОВКА ЗАДАЧИ



Нужно построить модель трехразрядного десятичного счетчика, работающего от импульсного генератора. На рис. 4.1 представлен счетчик, насчитавший 258 импульсов от генератора.

Рис. 4.1

4.3. МОДЕЛЬ

Для реализации имитационной модели нужно построить генератор "тиков", работающий с заданной частотой, и три одинаковых десятичных разряда счетчика. Следовательно, модель должна содержать три класса активных объектов: генератор "тиков", разряд счетчика по модулю 10 и, кроме того, корневой активный объект, который будет включать в себя один экземпляр генератора и три одинаковых экземпляра одноразрядного счетчика, связанные подходящим образом.

Создайте в своей рабочей директории новый проект под названием `DCounter`. Корневой объект назовите `Model`. Модель будет состоять из нескольких подсистем, связанных между собой.

4.3.2. ПРЕДСТАВЛЕНИЕ СИГНАЛА КАК СООБЩЕНИЯ

Генератор посылает сигналы с определенной частотой разряду счетчика. Для этого предназначен специальный пакет данных – сообщение. Сообщения принимаются и посылаются через специальные элементы активных объектов – порты. Обмен сообщениями возможен только между портами, соединенными соединителями – элементами, играющими роль путей движения сообщений.

В нашей модели счетчика важен только сам факт передачи сообщений, а не содержимое. Прием сообщения будет вызывать увеличение значения разряда счетчика.

4.3.1. ГЕНЕРАТОР ТИКОВ

Постройте новый активный объект `Gen`. Для этого в панели **Проекты** щелкните правой кнопкой мыши по имени проекта `DCounter` и в появившемся контекстном меню выберите команду **Создать/Класс активного объекта**. Назовите новый класс `Gen`.

Класс активного объекта `Gen` должен посылать сообщения первому разряду счетчика с заданной частотой. Для генерации таких сообщений создадим **Событие**, для этого перетащите элемент **Событие** из палитры **Основная** на диаграмму класса активного объекта `Gen`. В окне свойств этого события нужно оставить **Тип события** `По таймауту` без изменений, а **Режим** установить `Циклический`. Циклический режим события позволяет пользователю выполнять некоторые действия с требуемой периодичностью, например, каждое утро или ежегодно. Период срабатывания обратно-пропорционален частоте, поэтому в поле **Период** запишите `1/frequency`. В нашей модели `frequency` – это частота следования импульсов генератора, ее нужно объявить как параметр модели. Создайте в классе `Gen` параметр `frequency` со значением по умолчанию равным `2`.

4.3.1.1. ПРЕЗЕНТАЦИЯ ГЕНЕРАТОРА

Внешний вид генератора, а точнее объекта `Gen` представим прямоугольником, содержащим изображение импульса, как показано в правой части рис. 4.1. Откройте палитру **Презентация** и перетащите на диаграмму класса `Gen` прямоугольник, в свойствах укажите ширину этого прямоугольника – 80 единиц, а высоту – 100. Залейте его желтым цветом, цвет линии границы сделайте красным, а толщину линии равной 1.

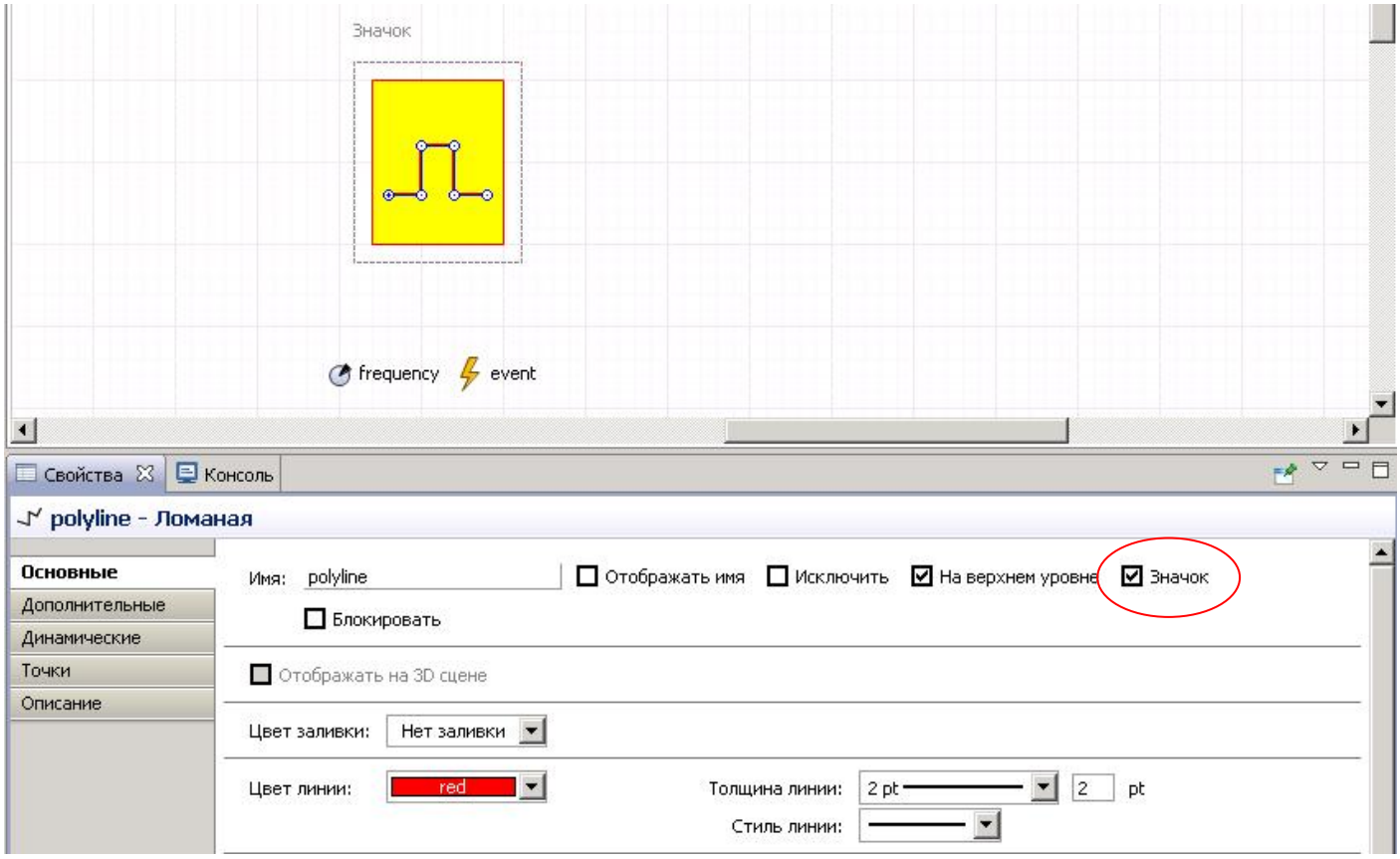
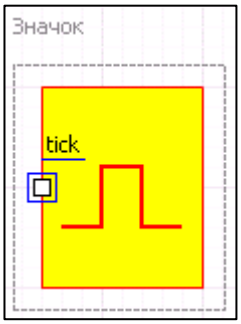


Рис. 4.2

Сделайте двойной щелчок мышью по элементу **Ломаная** в палитре (при этом его значок должен поменяться на этот:). Теперь Вы можете рисовать ломаную точка за точкой, последовательно щелкая мышью в тех точках диаграммы, куда Вы хотите поместить вершины ломаной. Чтобы завершить рисование, добавьте последнюю точку ломаной двойным щелчком мыши. Нарисуйте в поле прямоугольника импульс в соответствии с рис. 4.2.

Для того чтобы вложенные объекты – экземпляры класса (Gen) отображались на структурной диаграмме верхнего уровня (Model) в графическом виде и имели интерфейсные элементы – порты, нужно нарисовать для вложенного объекта значок. Мы уже нарисовали прямоугольник с импульсом внутри, для того чтобы сделать их значком выделите последовательно каждую из фигур и на вкладке **Основные** панели **Свойства** установите флажок **Значок**, как показано на рис. 4.2. Вокруг прямоугольника появится пунктирная линия и надпись Значок.



Как уже говорилось, порты играют центральную роль в механизме передачи сообщений. Сообщения посылаются и получаются портами. Порты являются двунаправленными, через них сообщения могут и посылаться, и приниматься. Перетащите элемент **Порт** из палитры **Основная** на диаграмму класса Gen, поместите его на левую сторону желтого прямоугольника и назовите этот порт tick, рис. 4.3.

Рис. 4.3

Через этот порт наш экземпляр класса активного объекта Gen может посылать и принимать сообщения от других активных объектов.

4.3.1.2. ГЕНЕРАЦИЯ ИМПУЛЬСОВ

Генератор нашей модели должен периодически с периодом 1/frequency посылать сообщения через порт tick. Ранее мы создали событие event, циклически срабатывающее с частотой frequency. При каждом срабатывании генератор должен выполнить действие - послать новое сообщение типа Object через выходной порт с именем tick. Для этого введите в поле **Действие** окна свойств события следующую строку:

```
tick.send ( new Object( ) );
```

4.3.2. РАЗРЯД СЧЕТЧИКА

Введите еще один класс активного объекта в проект – Counter. Этот класс будет моделировать разряд десятичного счетчика. Он должен иметь одну целую переменную – n, в которой будет храниться значение данного разряда, и два порта: tick и overflow. Создайте простую переменную n в классе Counter с начальным значением = 0.

4.3.2.1. ПРЕЗЕНТАЦИЯ РАЗРЯДА СЧЕТЧИКА

Разряд счетчика представим в виде прямоугольника с цифрой разряда внутри. На диаграмме класса Counter нарисуйте прямоугольник 60x100. Залейте его и линию рамки черным цветом. Внутрь прямоугольника вставьте цифру в виде текстового символа. Для этого поместите на прямоугольник текст (кнопка Aa на палитре **Презентация**) со значением 0 в поле **Текст**, вкладки **Основные**, свойств вставленного текста. Установите параметры текста: шрифт **SansSerif**, размер 72, цвет желтый. Во вкладку **Динамические** в поле **Текст** поместите имя переменной n, значение которой требуется отображать. Отметьте галочкой выбор **Значок** в свойствах текста и свойствах черного прямоугольника, рис. 4.4.

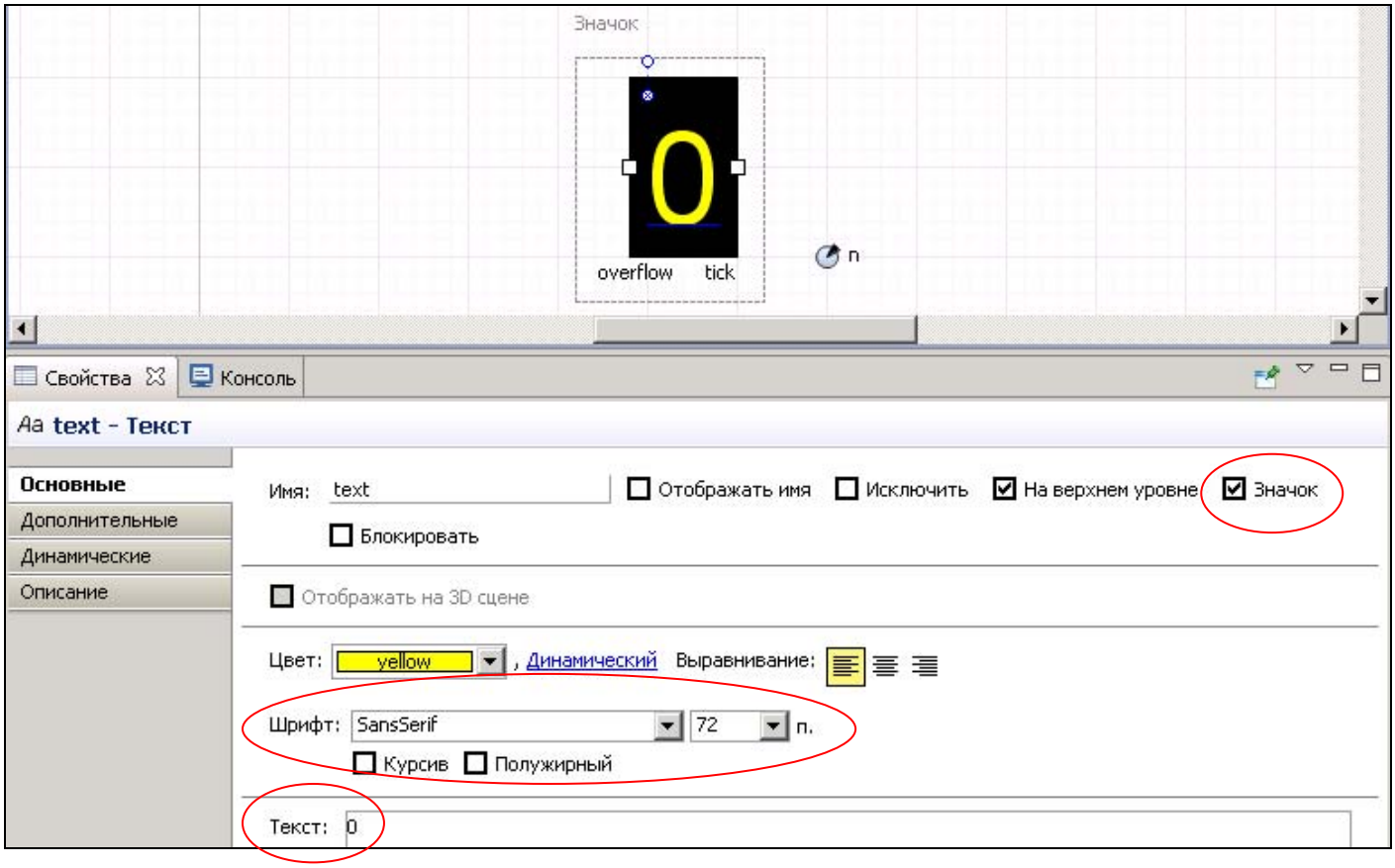


Рис. 4.4

4.3.2.2. СЧЕТ ИМПУЛЬСОВ

Создайте два порта: `tick` и `overflow` и поместите их на диаграмму класса `Counter`, как показано на рис. 4.4.

Каждый раз, когда приходит сообщение в порт `tick`, параметр `n` должен увеличиваться на единицу по модулю 10 с извещением по порту `overflow` о переполнении, если оно наступило. Это значит, что разряд счетчика увеличивает значение своего параметра `n` на 1 при приеме каждого сообщения, пришедшего через порт `tick`, кроме случая, когда `n` равно 9. В этом случае параметр `n` должен принять значение 0, а через выходной порт `overflow` будет послано сообщение. Именно это следует записать в поле **Действие при получении** в свойствах порта `tick` (рис. 4.5).

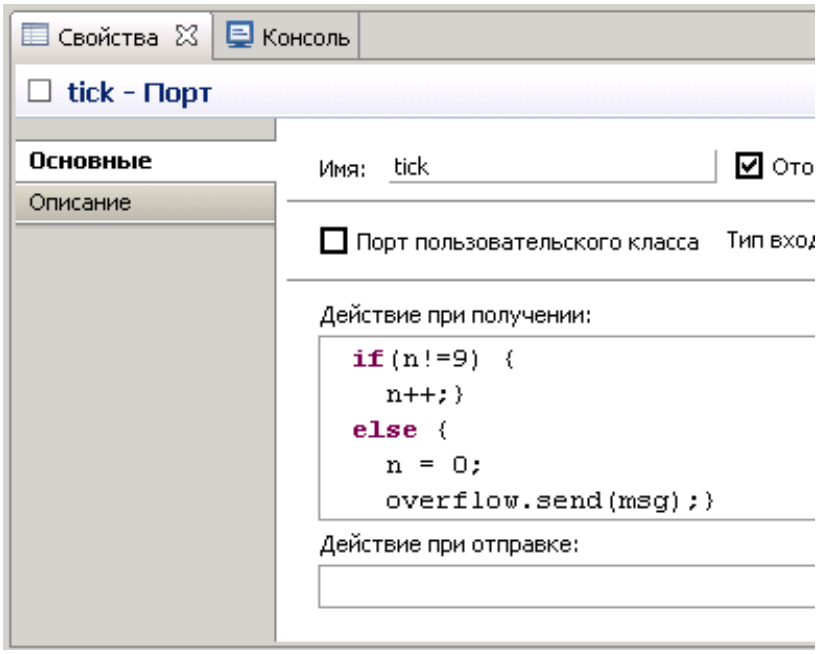


Рис. 4.5

Через порт `overflow`, таким образом, будет передаваться каждое десятое сообщение, полученное в порт `tick`.

4.3.3. КОРНЕВОЙ ОБЪЕКТ

Структура корневого объекта `Model` должна содержать один экземпляр активного объекта `Gen` и три экземпляра разряда счетчика `Counter`, соединенных по соответствующим портам.

Для создания экземпляра генератора в корневом объекте, откройте диаграмму класса активного объекта `Model` и перетащите на нее один экземпляр генератора (нажав левую клавишу мыши на имени `Gen`), а затем и три экземпляра счетчика – `Counter`. Экземпляры активных объектов получат имена по умолчанию и будут выглядеть так, как показано на рис. 4.6.

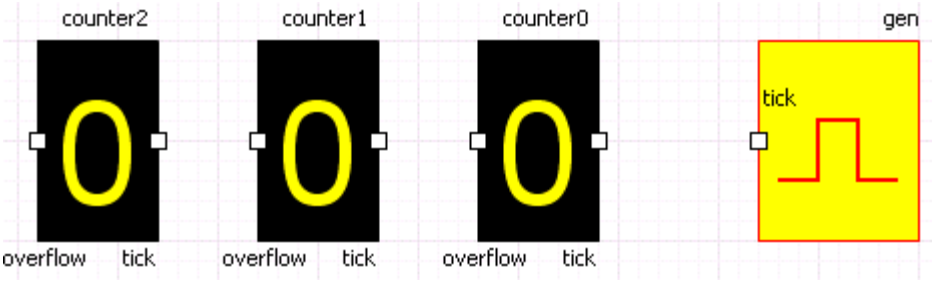
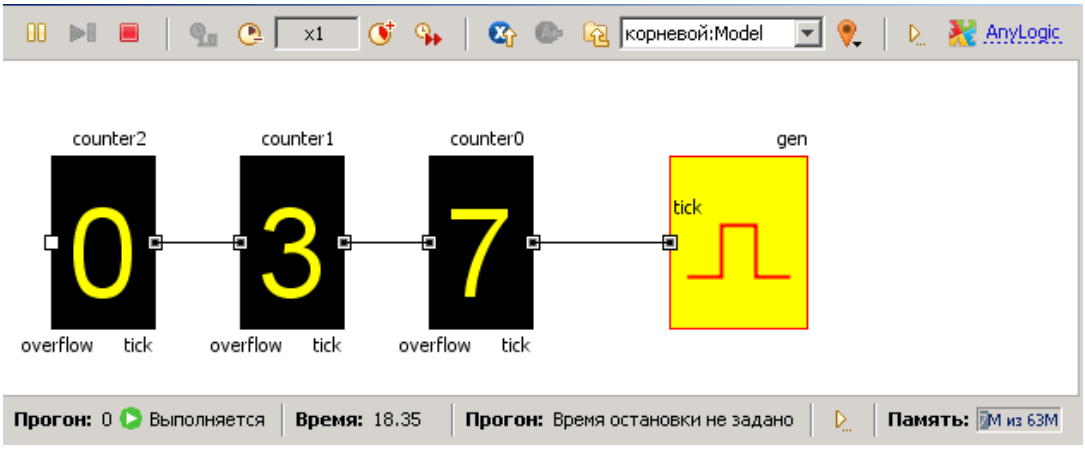


Рис. 4.6

Чтобы установить взаимодействие между объектами, Вам нужно соединить порты этих объектов с помощью соединителей. Соединитель - это линия, соединяющая два порта.

1. Сделайте двойной щелчок мышью по одному из портов.
2. Последовательно щелкните в тех местах диаграммы, где Вы хотите поместить точки изгиба соединителя (для нашей модели не обязательно).
3. Завершите процесс соединения, сделав двойной щелчок мышью по второму порту.



Запустите модель. Вы увидите изменение во времени разрядов счетчика (рис. 4.7).

Рис. 4.7

4.3.3.1. ПРЕЗЕНТАЦИЯ КОРНЕВОГО ОБЪЕКТА

Для придания созданной модели законченного вида следует добавить элементы описания и интерактивности.

Поместите в модель описание и слайдер, регулирующий частоту генератора, как показано на рис. 4.8. Продемонстрируйте свою модель преподавателю.

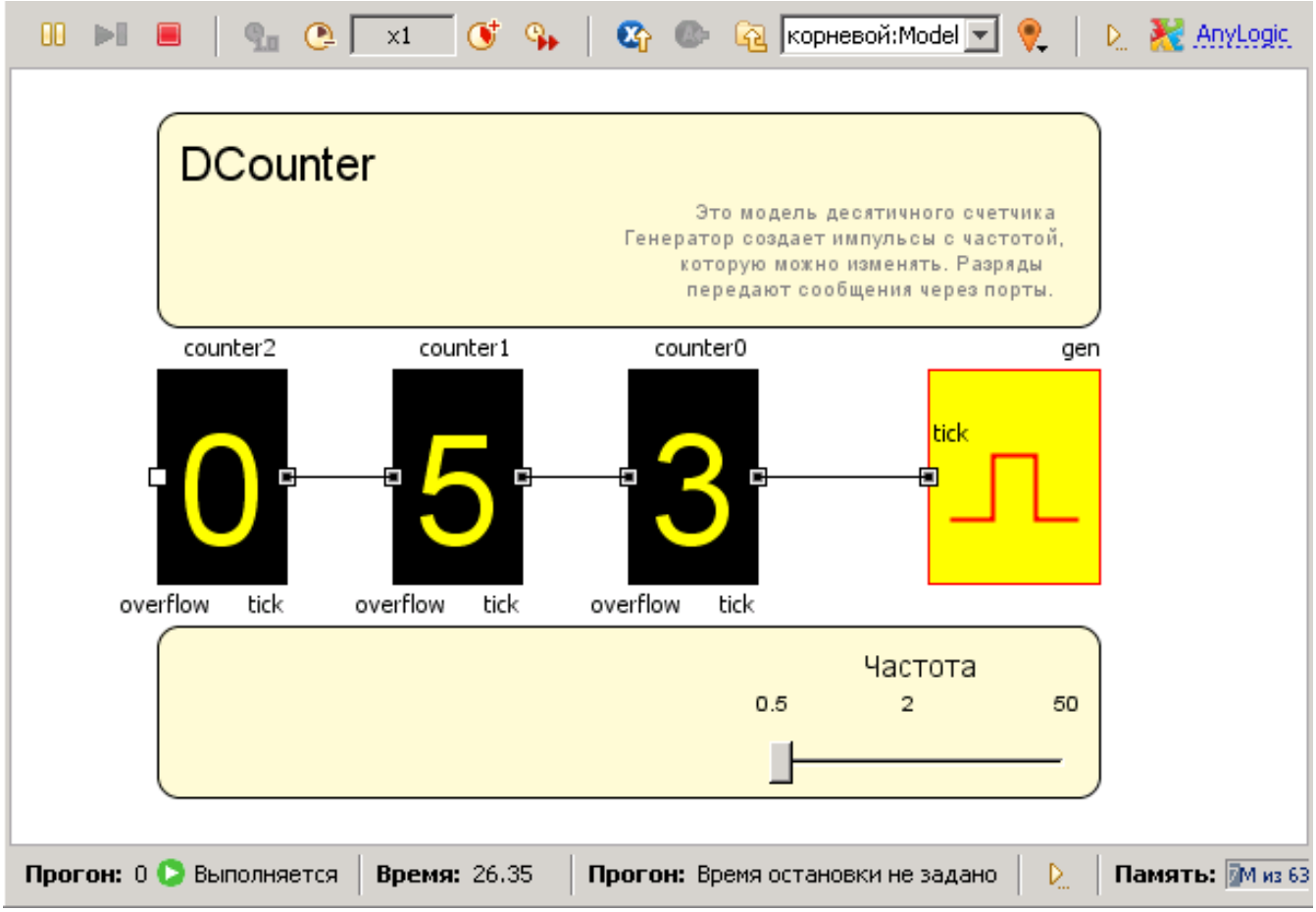


Рис. 4.8

4.4. КОНТРОЛЬНЫЕ ЗАДАНИЯ

1. Измените модель таким образом, чтобы счетчик начинал счет с произвольного числа. (5+)
2. Добавьте в модель переменную, которая будет принимать значения, показываемые счетчиком. (5+)
3. Измените модель таким образом, чтобы счетчик работал на убывание с определенного числа. (5+)
4. Измените презентацию модели таким образом, чтобы четные цифры показывались одним цветом, а нечетные - другим. (5)
5. Измените модель таким образом, чтобы параллельно основному счетчику работал второй счетчик, считающий каждый седьмой импульс. (5)
6. Измените презентацию модели таким образом, чтобы цифры мигали с частотой, которую можно изменять слайдером. (5)
7. Измените модель таким образом, чтобы цвет фона при значениях разряда счетчика = 0 был красным, а при остальных – синим. (5)
8. Измените модель таким образом, чтобы параллельно основному счетчику работал второй счетчик, считающий каждый четный импульс. (5)
9. Создайте модель 16-ричного счетчика, работающего параллельно с десятичным. (5)
10. Измените модель таким образом, чтобы параллельно основному - десятичному счетчику, работал двоичный. (4)
11. Измените модель таким образом, чтобы графическое изображение импульса генератора мигало синхронно генерации «тиков». (4)
12. Измените модель таким образом, чтобы счетчик работал на убывание. (4)

13. Измените модель таким образом, чтобы генератор создавал «тики» в случайные моменты времени. (4)
14. Измените презентацию модели таким образом, чтобы каждая цифра счетчика показывалась своим цветом. (4)
15. Дайте определение дискретно-событийной системы, приведите примеры. (3)
16. Охарактеризуйте все типы событий, реализованных в AnyLogic. (3)
17. Что такое порт, и какие элементы AnyLogic могут иметь порт? (3)
18. Добавьте еще один разряд к счетчику. (3)
19. Добавьте второй счетчик, работающий вместе с первым от одного генератора. (3)