

Лабораторная работа

Модель калькулятора, вычисляющего факториал

Постановка задачи

Рассмотрим процесс создания моделей в среде AnyLogic на примере упрощенной версии калькулятора, вычисляющего факториал.

Последовательно выберем пункты меню *Файл – Создать – Модель* или щелкнем по пиктограмме палитры инструментов



. Открывается диалоговое окно начальных настроек модели (рис. 2.4).

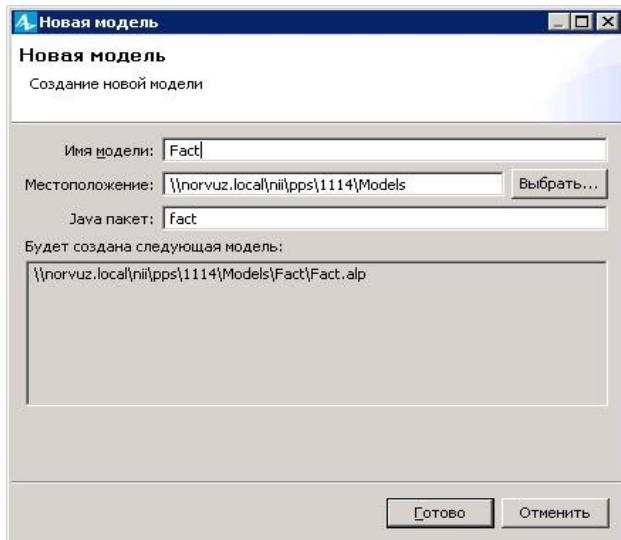


Рис 2.4 Окно начальных настроек

По умолчанию AnyLogic предлагает имя проекта, в виде: *Model#*, где # – номер, соответствующий порядку создания модели.

Изменим его на *Fact* и, если необходимо, выберем другое местоположение, где будут сохраняться все файлы модели. Поскольку используются языки Java, компоненты приложения (классы) собираются в пакеты. Таким образом, в AnyLogic можно создавать несколько моделей с разными именами, хранящих ресурсы в одном пакете. При попытке сохранить модель под

новым именем, например, *Fact_1* – пункт меню *Файл–Сохранить как* – имя пакета по умолчанию останется прежним.

Если его не изменить (например на *fact_1*), то появятся две модели с разными именами и общим пакетом. Одновременно работать с ними (загружать оба проекта) будет нельзя.

В Java учитывается регистр букв в именах. Название модели и ее компонентов может содержать русские буквы.

Нажмем кнопку *Готово*.

В качестве основы дизайна приложения выберем *Калькулятор*, который есть в стандартном наборе программ Windows (рис. 2.5).

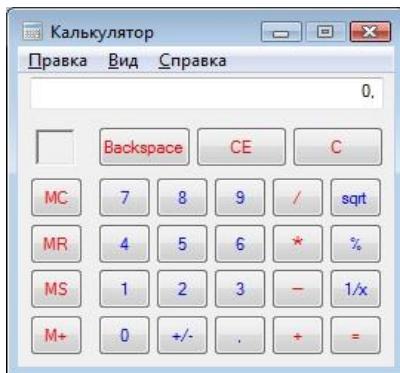


Рис. 2.5. Калькулятор Windows. Вид «Обычный»
В результате получится вид как на рис. 2.6.

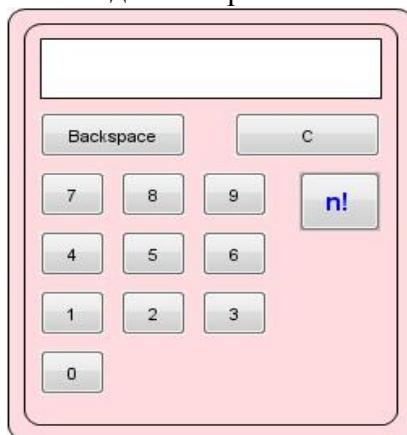


Рис. 2.6. Вид модели калькулятора

Сформулируем задачу: построить функциональную модель калькулятора. С помощью цифровых кнопок набирается значение « n », которое отображается на «цифровом табло». Кнопкой *Backspace* можно последовательно убирать последние цифры. Нажатие кнопки $n!$ приводит к появлению значения факториала в цифровом табло. Нажатие кнопки *C* (*Clear*) очищает табло.

Нам понадобятся две переменные, элементы управления, элементы презентации, а также фрагменты, определяющие логику модели.

Опишем алгоритм реализации задачи в виде последовательности шагов.

Шаг 1. Первая переменная будет принимать значение n , набираемое с помощью кнопок. Раскроем закладку палитры элементов. Перетащим в окно графического редактора элемент *Переменная*, как показано на рис. 2.7.

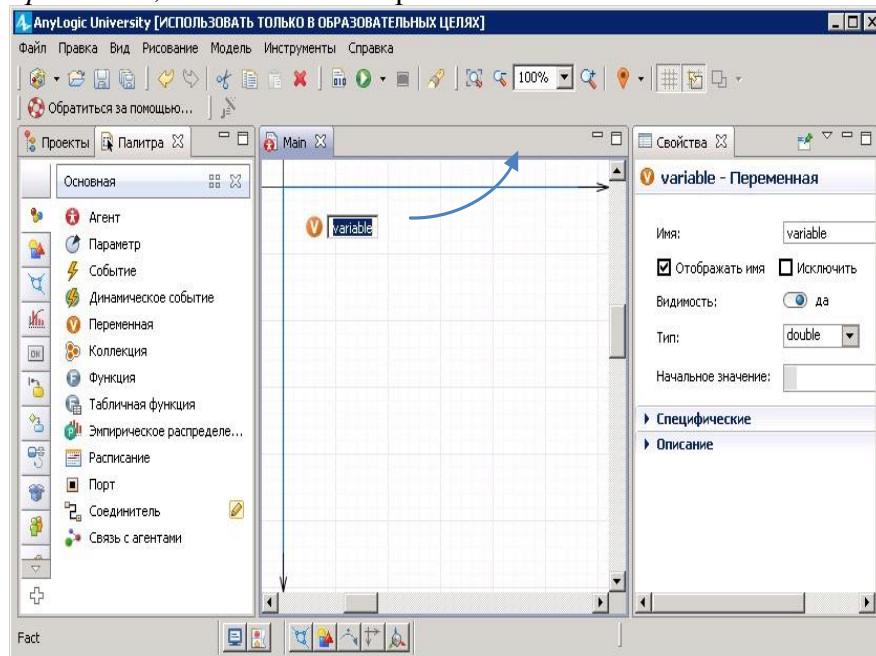


Рис. 2.7. Перетаскивание элемента *Переменная* в окно графического редактора

Изменим свойства переменой в соответствии с рис. 2.8.

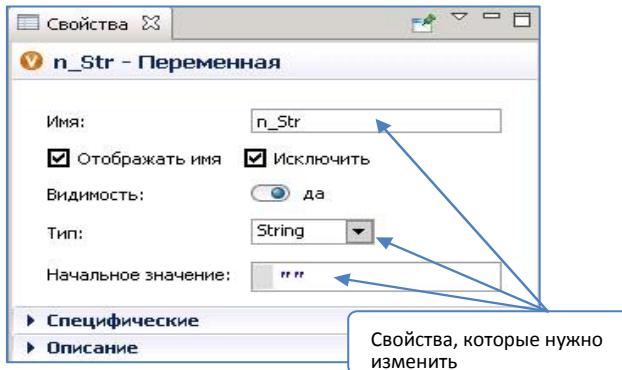


Рис. 2.8. Панель свойств переменной

Шаг 2. Изменим имя со стандартного *variable* на *n_Str*. Учтем правила языка Java. Так как это объекты, их имена (а также имена методов) должны начинаться со строчной буквы. Всем переменным AnyLogic устанавливает по умолчанию тип *double* (вещественный) и начальное значение, равное нулю (если его не изменить, в поле *Начальное значение* не показывается).

В данном случае нужна переменная, которая будет хранить «число», набираемое кнопками, т.е. его изображением на табло должна быть *строка* (а не сумма) цифр, позиция которых будет соответствовать их разрядам.

Изменим тип переменной на *String* (строковый) и установим с помощью двух кавычек ее начальное значение («пустая строка»). В противном случае оно всегда будет *null* (так как это переменная ссылочного типа).

Шаг 3. Введем в модель вторую переменную, которая будет принимать конечное значение параметра цикла, необходимое для вычисления факториала. Для этого повторим шаг 1. Изменим стандартное имя переменной на *n*.

В свойствах переменной установим тип *int* (целый).

Шаг 4. Добавим в модель элемент *Текст* из группы палитры инструментов *Презентация*. Он будет отображать на табло введенное кнопками число *n*. Внесем изменения на панели свойств (рис. 2.9).

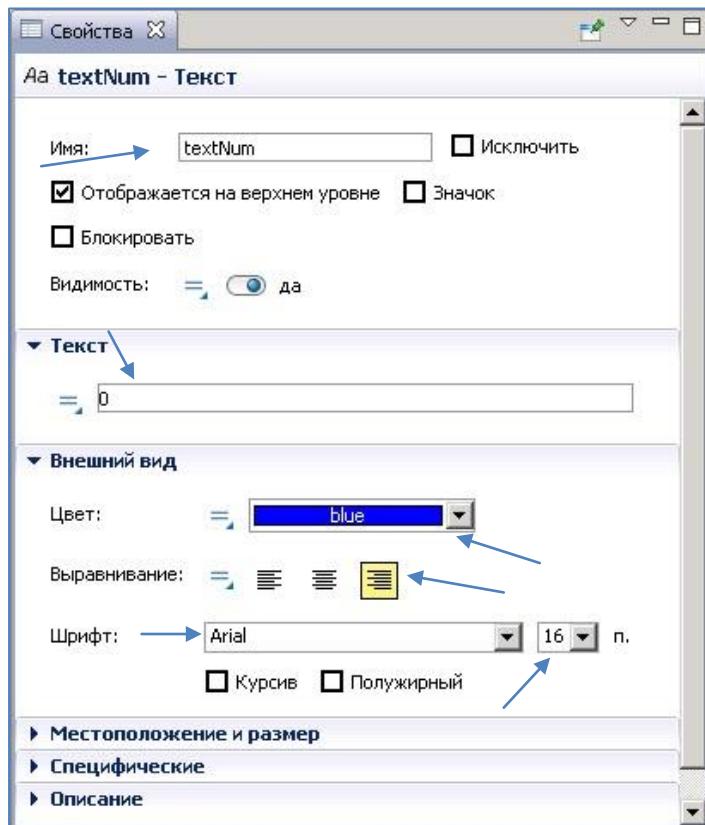
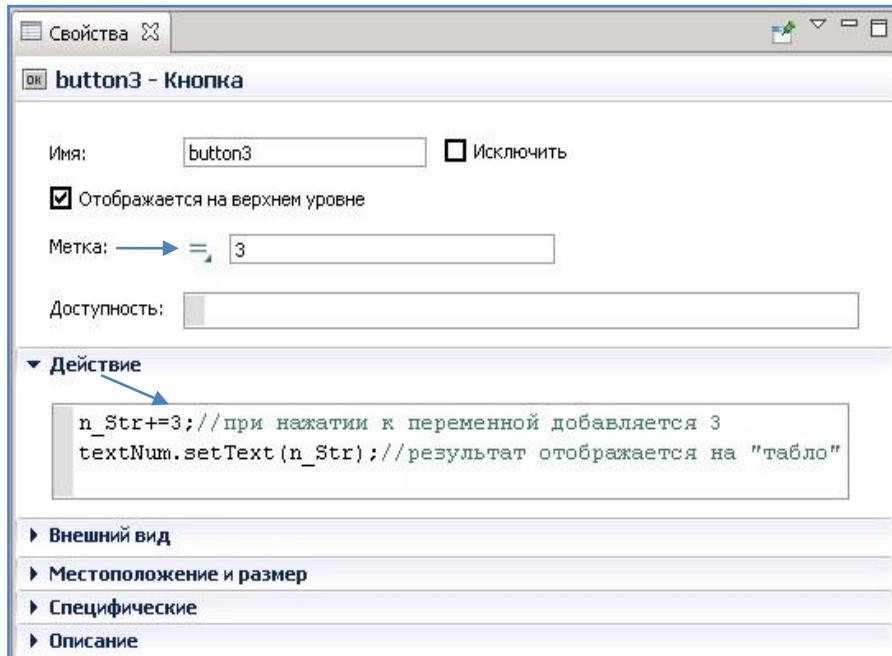


Рис. 2.9. Панель свойств элемента *Текст* (стрелками показаны измененные свойства)

Шаг 5. Добавим в еще один элемент *Текст*. Назовем его *textFact*. Он будет показывать значение факториала на табло. Повторим для него операции шага 4. Выравнивание установим по левому краю.

Шаг 6. Поместим в поле графического редактора кнопку из группы *Элементы управления*. Чтобы сразу получить убедительный результат вычислений, пусть это будет кнопка с цифрой «3» ($3!=6$) (рис. 2.10).



**Рис. 2.10. Панель свойств элемента Кнопка
(стрелками показаны изменения)**

Впервые в модели появляется фрагмент кода на языке Java. Это всего лишь две строки. Первая изменяет значение текстовой переменной, вторая вызывает метод *setText()* элемента (объекта) *textNum*. Результат ясен из комментариев.

В Java предусмотрены два вида комментариев: для строки – начинается с двух косых черт *//*, и для фрагмента, расположенного между парными символами */* и */*.

Шаг 7. Поместим в поле графического редактора еще одну кнопку. Потянув за угловой маркер, немного увеличим ее размеры. Результатом ее нажатия будет вычисление факториала и вывод результата на табло. Свойства кнопки и вид окна редактора модели показаны на рис. 2.11.

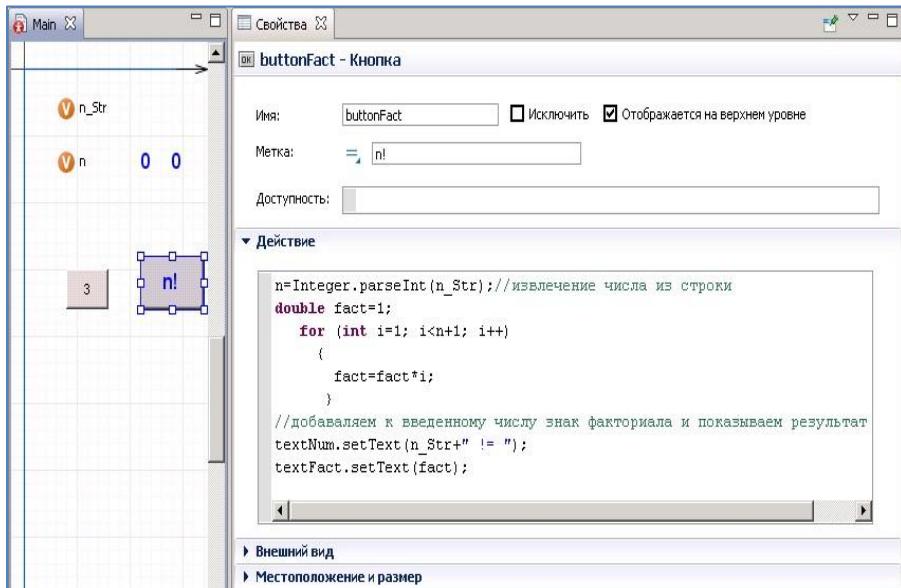


Рис. 2.11. Вычисление факториала по щелчку кнопки

При вычислении факториала невозможно напрямую использовать строковое представление числа в переменной *n_Str*, поэтому в первой строке кода «извлекаем» его с помощью класса-оболочки *Integer* и присваиваем результат переменной *n*.

Вычисление факториала будем осуществлять с помощью цикла с параметром.

В теле кода объявим вспомогательные переменные *fact* (тип вещественный) и параметр цикла *i* (целый), пробегающий значения от 1 до *n*.

Выполнение двух последних строк кода будет приводить к изменению соответствующих текстовых элементов. В *textNum* появится значение *n*, символы «!=» и «=>». В *textFact* отобразится значение факториала.

Шаг 8. Запустим модель с помощью кнопки на панели инструментов или клавиши *F5*. Вначале откроется *Окно настроек эксперимента* (рис. 2.12).

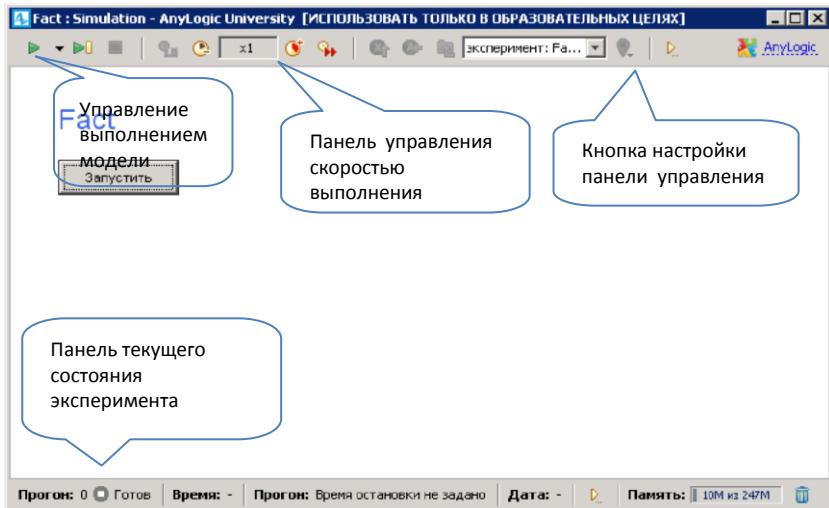


Рис. 2.12. Окно настроек эксперимента

Щелчком по кнопке запуска модели попадаем в *Окно презентации*. Модель работает. Последовательно нажимаем кнопки «3» и «n!». Текстовые элементы отображают корректный результат (рис. 2.13).

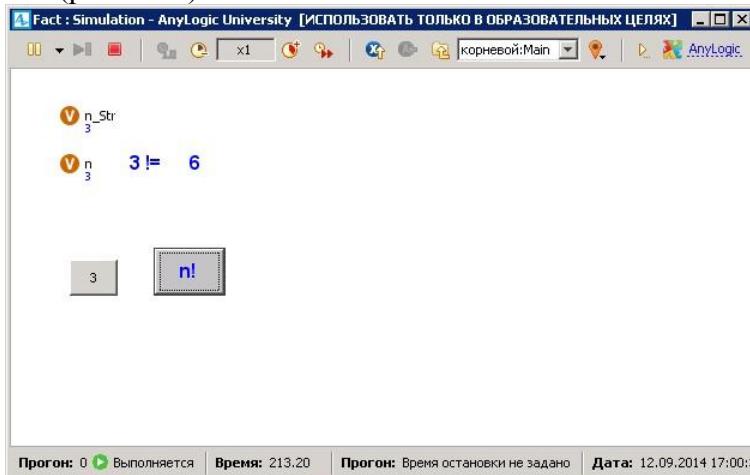


Рис. 2.13. Вид окна работающей модели

Дальнейшие действия будут сводиться к добавлению кнопок, оформлению дизайна «калькулятора» и вида готового приложения.

Шаг 9. Поместим элементы в соответствующие позиции. Цифровые кнопки «размножим» с помощью операций последовательного копирования-вставки уже имеющейся у нас кнопки «3». Они будут наследовать ее свойства, включая размеры и код. Необходимо изменить пункты свойств *Имя кнопки*, *Метка* и *Действие* (код). Так, для кнопки «0», код действий примет вид:

```
n_Str+=0;
```

```
textNum.setText(n_Str);
```

Шаг 10. Напишем фрагмент кода для кнопки «C»

(Clear):

```
n_Str="";
```

```
textNum.setText("");
```

```
textFact.setText("");
```

Щелчок по ней приведет к начальным значениям строковой переменной *n_Str* и двух текстовых элементов, отображающих ввод *n* и вывод *n!*

Состояние модели на данном этапе показано на рис. 2.14.



Рис. 2.14. Вид модели на стадии 9

Шаг 11. Окончательная стадия разработки модели. Поместим в окно графического редактора элемент *Скругленный прямоугольник* из группы *Презентация* панели *Палитра*. Он будет выполнять функцию электронного табло калькулятора. Подгоним его размеры под ширину клавиатуры. Чтобы он не закрывал текстовые элементы, щелчком правой клавиши мыши вызовем контекстное меню *Порядок–На задний план*.

Новые элементы, помещаемые в окно графического редактора, последовательно располагаются на переднем плане.

В свойствах *Радиус уменьшим радиус углов с 10 до 5* (рис. 2.15).

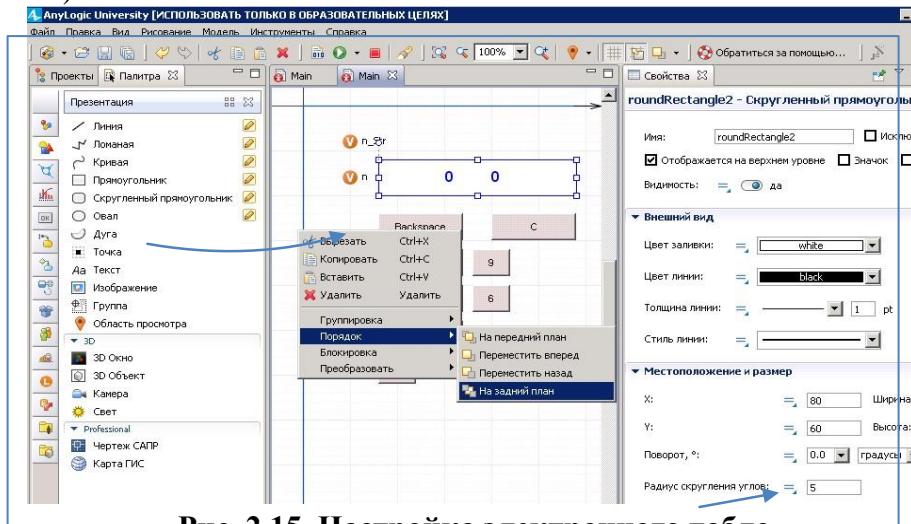


Рис. 2.15. Настройка электронного табло

Проделаем аналогичные действия с двумя другими прямоугольниками, которые будут имитировать корпус калькулятора. В свойствах первого прямоугольника уберем заливку.

Немного увеличим размер второго прямоугольника (рис. 2.16).

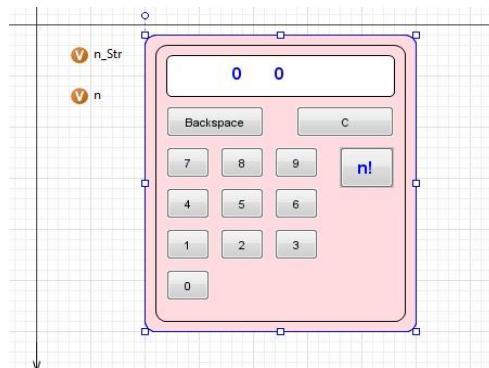


Рис. 2.16 Заключительная стадия разработки калькулятора

Шаг 12. Настройка приложения. Переместим переменные за пределы область показа, например, влево от оси Y. Выделим мышью все элементы калькулятора и передвинем их немного вниз и влево.

В окне презентации по умолчанию показывается область графического редактора определенных размеров, расположенная под осями координат (ось ординат Y «перевернута».) Модель может быть больше или меньше этого пространства, поэтому область первоначального показа можно изменить и организовать несколько таких зон и навигацию между ними. Сделаем это позже на примерах других моделей.

В панели проектов мышью щелкнем по элементу *Simulation:Main*, затем по *frame* (рамка). Рядом с окном графического редактора появится закладка *Simulation*. Для показа калькулятора и работы с ним вместе с необходимыми полями, достаточно области размером 350 точек по горизонтали и 400 по вертикали.

Изменим *Свойства* рамки в соответствии с этими значениями, затем скорректируем другие элементы (рис. 2.17).

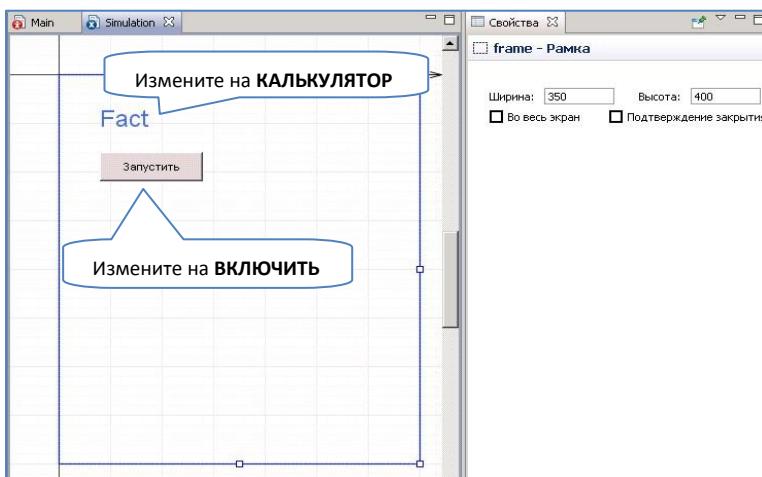


Рис. 2.17. Настройка области показа модели

В окончательном варианте приложение примет вид, как на рис. 2.18.

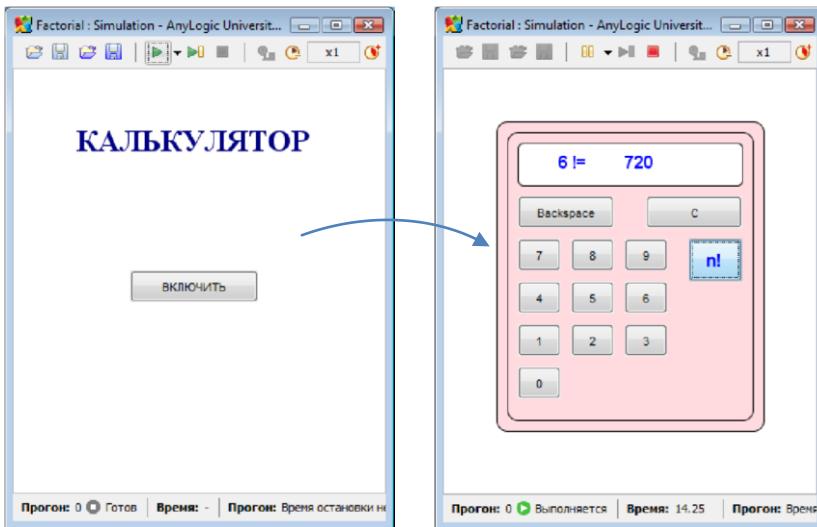


Рис. 2.18. Окончательный вариант модели

Шаг 13. Сохранение приложения. В AnyLogic можно экспорттировать проект в независимый от среды разработки Java-апплет, запускаемый в браузере (рис. 2.19).

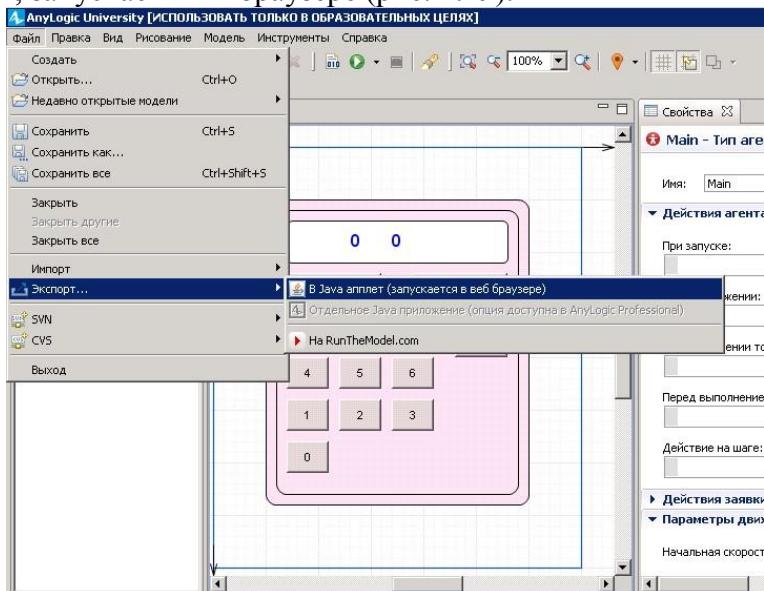


Рис. 2.19 Меню экспорта модели

В диалоговом окне выберем каталог для сохранения апплета (рис. 2.20).

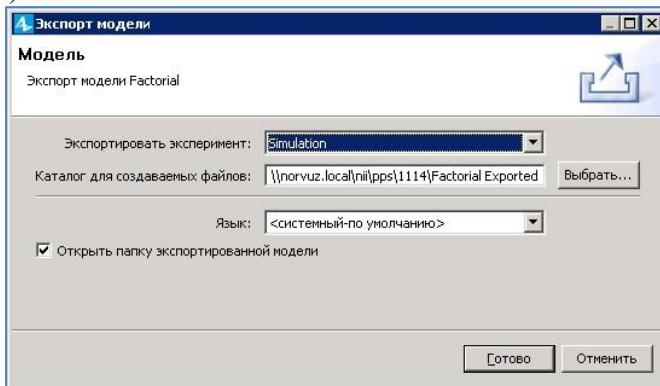


Рис. 2.20. Диалоговое окно экспорта модели в апплет
Приложение работает в браузере Internet Explorer (рис. 2.21).

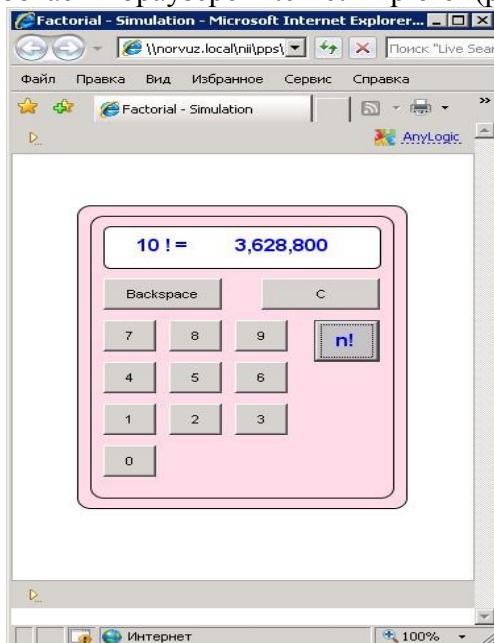


Рис. 2.21. Вид страницы с апплетом в браузере Internet Explorer