

Лабораторная работа

Построение системно-динамической модели в AnyLogic

(модель распространения среди населения инноваций и новых продуктов)

Цель работы: изучить интерфейс и возможности пакета AnyLogic для построения моделей системной динамики.

Порядок выполнения работы

В ходе лабораторной работы необходимо создать и изучить типичную системно-динамическую модель, представляющую интерес в экономике. Это модель распространения среди населения инноваций и новых продуктов, разработанная Франком Бассом. Среди бизнес-аналитиков она является одной из самых популярных моделей исследования рынка новых продуктов.

В работе дано подробное описание этой модели, инструкции по созданию модели в пакете AnyLogic и предложены некоторые ее расширения. При возникновении трудностей при создании модели используйте учебное пособие по системной динамике пакета AnyLogic (находится в меню **Справка**).

Задача 1. Модель жизненного цикла продукта

Модель представляет собой динамику процесса превращения потенциальных покупателей нового продукта (*Potential_Adopters*) во владельцев продукта (*Adopters*). Изначально продукт никому не известен, и для того, чтобы люди начали его приобретать, он рекламируется. В итоге люди покупают продукт либо под воздействием рекламы, либо узнав о нем от знакомых, по «сарафанному радио». Эффективность рекламы пропорциональна числу людей, на которых она действует, т.е. числу потенциальных покупателей. В свою очередь, эффективность «сарафанного радио» зависит от числа людей, уже купивших продукт. Иными словами, в данной модели должна быть отражена структура взаимных зависимостей характеристик и параметров системы.

Для описания модели в терминах системной динамики необходимо определить ключевые переменные модели и их влияние друг на друга, а затем создать потоковую диаграмму модели. При создании потоковой диаграммы нужно учесть, какие переменные должны быть представлены накопителями, какие потоками, а какие – вспомогательными переменными.

При создании потоковой диаграммы выявляются переменные, которые накапливают значения с течением времени. В данной модели численности потребителей и потенциальных потребителей продукта являются накопителями, а процесс приобретения продукта – потоком.

Системно-динамическое представление данной модели показано на рис. 17. Накопители обозначаются прямоугольниками, поток – вентилем, а вспомогательные переменные – кружками. Стрелки обозначают причинно следственные зависимости в модели.

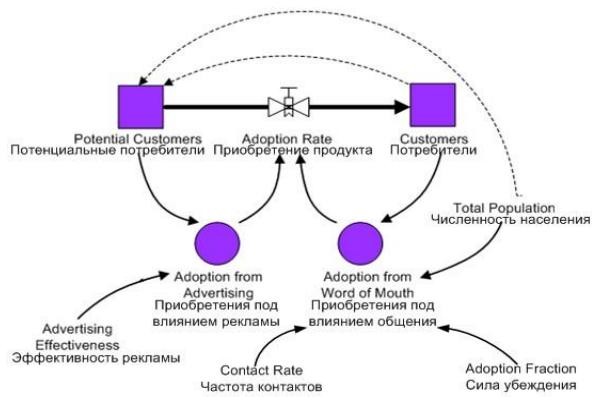


Рис. 17

Задание к лабораторной работе

В AnyLogic потоковая диаграмма создается с помощью структурной диаграммы. На диаграмме графически задаются накопители, потоки и вспомогательные переменные.

1. Создайте новый проект для будущей модели и сохраните его в своей папке. Откройте структурную диаграмму двойным щелчком мыши по элементу дерева *Main* в окне **Проект**.
2. Создайте два накопителя для того, чтобы смоделировать численности потребителей и потенциальных потребителей продукта. Для этого перетащите элемент **Накопитель** из палитры **Системная динамика** на диаграмму класса активного объекта. На диаграмме появится маленький голубой прямоугольник, обозначающий переменную-накопитель (что соответствует классической нотации системной динамики). Измените имя накопителя – окно **Свойства**, вкладка **Основные**, введите *PotentialAdopters* в поле редактирования **Имя**. Таким же образом создайте еще один накопитель, назовите его *Adopters* (рис. 18).

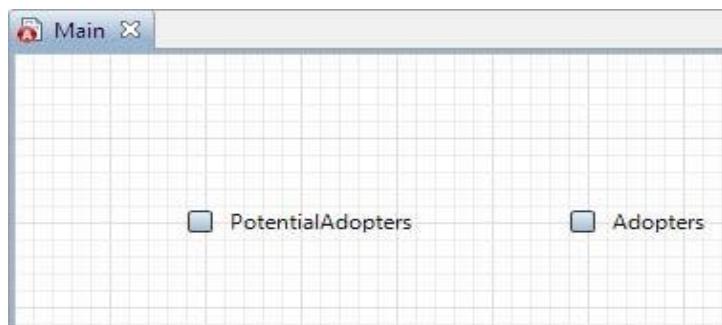


Рис. 18

3. Создайте поток приобретения продукта, увеличивающий число потребителей продукта и уменьшающий численность потенциальных потребителей.

В AnyLogic поток создается с помощью специального инструмента задания потоков. Инструмент задания потоков создает переменную-поток и автоматически конфигурирует накопители, в которые входит или из которых исходит данный поток, таким образом, что их значения изменяются в соответствии со значением этого потока.

Чтобы создать поток, сделайте двойной щелчок мыши по накопителю *PotentialAdopters*, а потом щелкните по накопителю *Adopters*. AnyLogic создаст новую переменную-поток и сделает ее исходящим потоком для накопителя *PotentialAdopters* и входящим – для *Adopters*. На диаграмме появятся стрелки, которые будут обозначать образовавшиеся зависимости между потоком и этими накопителями (рис. 19). Выделите созданную переменную в графическом редакторе и измените имя этого потока на *AdoptionRate*.

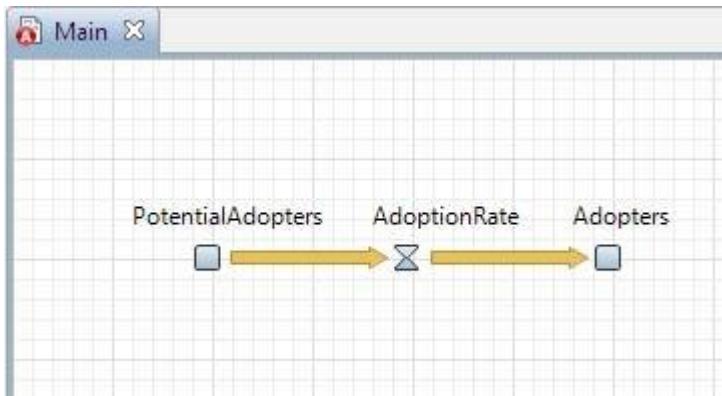


Рис. 19

4. Посмотрите свойства накопителей. Формулы накопителей должны выглядеть следующим образом (рис. 20).

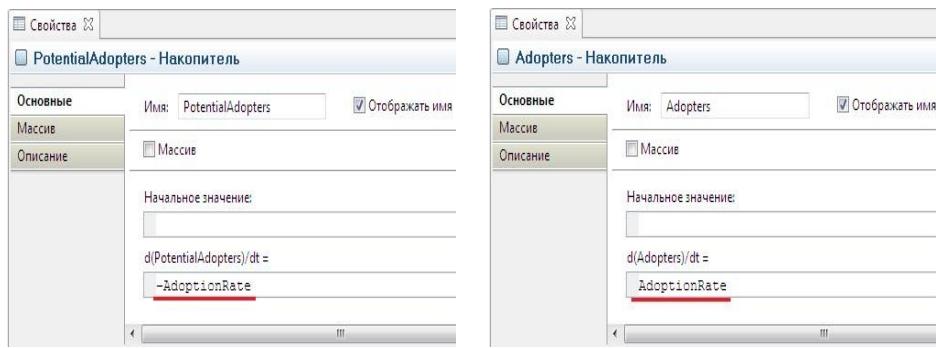


Рис. 20

Эти формулы были автоматически заданы инструментом задания потоков.

5. Создайте константы модели. Перетащите элемент **Параметр** из палитры **Системная динамика** на диаграмму класса активного объекта.

- 1) Создайте константу, задающую общую численность населения. Для этого создайте новый параметр и измените его свойства. В поле **Имя** введите *TotalPopulation*. В поле **По умолчанию** введите 100000. Можно задать также краткое описание константы в поле **Описание**;
- 2) в нашей модели интенсивность рекламы и вероятность того, что продукт будет приобретен под ее влиянием, полагаются постоянными. Создайте константу, задающую эффективность рекламы. Назовите ее *AdEffectiveness*. Задайте значение по умолчанию 0.011;
- 3) частота, с которой потенциальные потребители общаются с потребителями, принимается как постоянная величина. Поэтому задайте частоту контактов константой. Назовите константу *ContactRate*. Предположим, что каждый потенциальный потребитель в среднем встречается со 100 постоянными потребителями в год. Задайте значение по умолчанию 100;
- 4) задайте константой силу убеждения владельцев продукта, определяющую ту долю контактов, которая приводит к продажам продукта. Назовите константу *AdoptionFraction*. Задайте значение 0.015.

6. Задайте начальные значения накопителей.

Начальное число потребителей нашего продукта равно нулю, поэтому в окне свойств накопителя *Adopters* введите 0 в поле редактирования **Начальное значение**.

Начальное количество потенциальных потребителей будет равно общей численности населения. В окне свойств накопителя *PotentialAdopters* введите *TotalPopulation* в поле редактирования **Начальное значение**. Вы можете сделать это с помощью Мастера (**Ctrl + пробел**).

7. Создайте две вспомогательные переменные, которые будут соответствовать двум составляющим потока приобретения продукта – приобретениям, совершенным под влиянием рекламы и под влиянием потребителей продукта соответственно.

- 1) Перетащите элемент **Вспомогательная переменная** из палитры **Системная динамика** на диаграмму класса активного объекта и назовите ее *AdoptionFromAd*. В поле **AdoptionFromAd =** введите *AdEffectiveness * PotentialAdopters*.

Влияние рекламы моделируется следующим образом: некий постоянный процент потенциальных клиентов *AdEffectiveness* всё время переходит в разряд клиентов. Их доля в *AdoptionRate* равна *PotentialAdopters * AdEffectiveness*;

- 2) создайте еще одну переменную и назовите ее *AdoptionFromWOM*. Задайте формулу интенсивности продаж продукта под влиянием устного общения потребителей продукта с теми, кто данный продукт еще не приобрел: *ContactRate * AdoptionFraction * PotentialAdopters * Adopters / TotalPopulation*.

Проанализируйте данную формулу.

8. Задайте формулу для потока приобретения продукта. Значение потока определяется суммой двух его независимых составляющих – продаж в результате рекламного влияния и продаж под влиянием общения с потребителями продукта. В окне свойств переменной *AdoptionRate* на вкладке **Основные** введите формулу, по которой будет вычисляться значение потока, в поле **AdoptionRate=** : *AdoptionFromAd* + *AdoptionFromWOM*

Создание модели завершено. Диаграмма накопителей и потоков должна выглядеть, как показано на рис. 21.

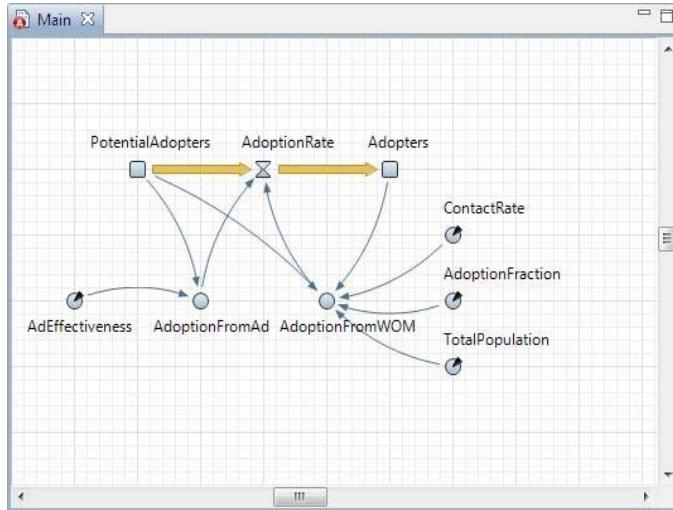


Рис. 21

9. Просмотрите причинно-следственные зависимости между накопителями, потоками и вспомогательными переменными в модели.

Зависимости будут показаны стрелками, как в общепринятой системнодинамической нотации. Стрелка, направленная от потока к накопителю, означает, что этот поток является входящим потоком для данного накопителя. Стрелка, направленная от накопителя к потоку, означает, что поток является исходящим. Тонкая стрелка, направленная от переменной *A* к переменной *B*, означает, что изменение значения переменной *A* вызовет изменение значения переменной *B*.

Можно увидеть, что модель содержит два цикла с обратной связью: компенсирующий и усиливающий. *Компенсирующий цикл* с обратной связью воздействует на поток приобретения продукта, вызванный рекламой. Поток приобретения продукта сокращает число потенциальных потребителей, что приводит к снижению интенсивности приобретения продукта. *Усиливающий цикл* с обратной связью воздействует на поток приобретения продукта, вызванный общением с потребителями продукта. Поток приобретения продукта увеличивает численность потребителей продукта, что приводит к росту интенсивности приобретения продукта под влиянием общения с потребителями продукта, и, следовательно, к росту интенсивности приобретения продукта.

10. Просмотрите код модели. Для этого постройте модель (клавиша **F7**), а потом на панели **Проект** выберите **Модель**, нажмите правую кнопку мыши и из ниспадающего меню выберите **Открыть в...** и затем – в **Java редакторе**.

11. Сконфигурируйте выполнение модели, для этого необходимо настроить текущий эксперимент модели.

Если сейчас запустить модель, то она будет работать бесконечно. Поскольку мы хотим наблюдать поведение модели только тогда, когда происходит процесс распространения продукта, поскольку нам нужно остановить модель, когда система придет в точку равновесия. Процесс распространения продукта в этой модели длится примерно 8 лет. Поэтому задайте останов модели после 8 единиц модельного времени.

Для этого в окне свойств эксперимента *Simulation:Main* перейдите на вкладку **Модельное время**, выберите **В заданное время** из выпадающего списка **Остановить**. В

расположенном ниже поле введите 8. Модель остановится после того, как истекут 8 единиц модельного времени.

Задайте выполнение модели в режиме реального времени (вкладка **Презентация** окна свойств эксперимента). Задайте скорость выполнения – 2.

Вы можете сменить метод, используемый для решения системы дифференциальных уравнений. Если вы не укажете никакого конкретного метода, т. е. оставите выбранный по умолчанию метод Automatic, то во время работы модели AnyLogic будет автоматически выбирать численный метод в соответствии с поведением системы. На вкладке **Дополнительные** окна свойств эксперимента выберите метод RK4 из выпадающего списка **Дифф. уравнения**.

11. Проверьте ошибки и запустите модель. Для проверки ошибок постройте проект с помощью кнопки панели инструментов **Постройте** (или клавиша F7). В окне **Ошибки** появится список всех ошибок, обнаруженных в проекте, если таковые имеются. Двойным щелчком мыши по ошибке в этом списке вы можете перейти к предполагаемому месту ошибки, чтобы исправить ее. После построения проекта запустите модель.

12. Просмотрите значения переменных в окне работающей модели.

13. Исследуйте динамику обеих составляющих потока продаж. Для этого откройте окно инспектора для переменной *AdoptionFromAd* в окне презентации.

Вы можете переключить окно инспектора в режим графика – оно будет отображать временной график изменений значения переменной в модельном времени. Текущее значение переменной будет отображаться рядом с началом координат графика. Окно инспектора автоматически масштабируется таким образом, чтобы полностью вместить кривые графиков от начала до конца периода моделирования.

Откройте окно инспектора переменной *AdoptionFromWOM* и переключите его в режим графика (рис. 22).

Можно увидеть, что при внедрении нового продукта на рынок, когда число потребителей равно нулю, реклама будет являться единственным источником продаж. Наибольший рекламный эффект отмечается в начале процесса распространения продукта; он неуклонно падает по мере уменьшения численности потенциальных потребителей.

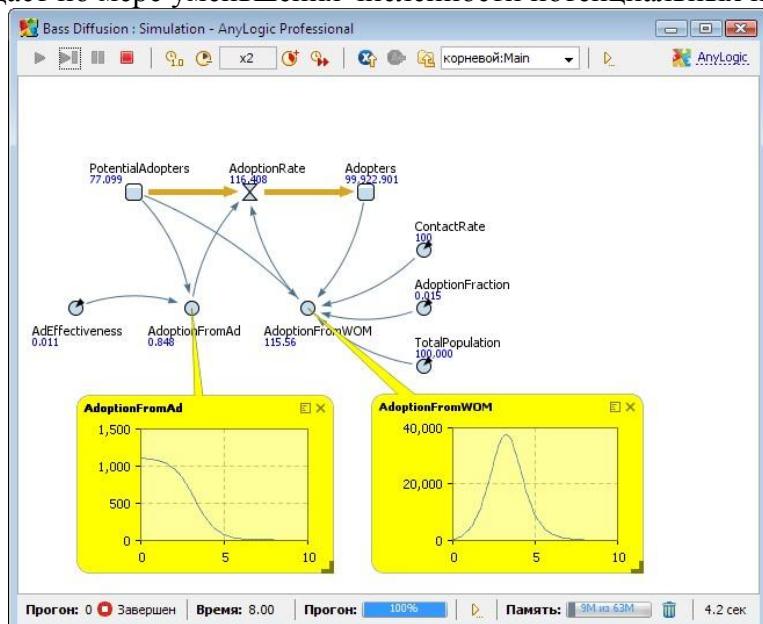


Рис. 22

14. Изучите динамику изменения численностей потребителей и потенциальных потребителей продукта с помощью диаграмм. Для этого создайте диаграмму для отображения переменных *Adopters* и *PotentialAdopters*.

Перетащите элемент **Временной график** из палитры **Статистика** на диаграмму класса *Main* и измените размер графика, как показано на рис. 23.

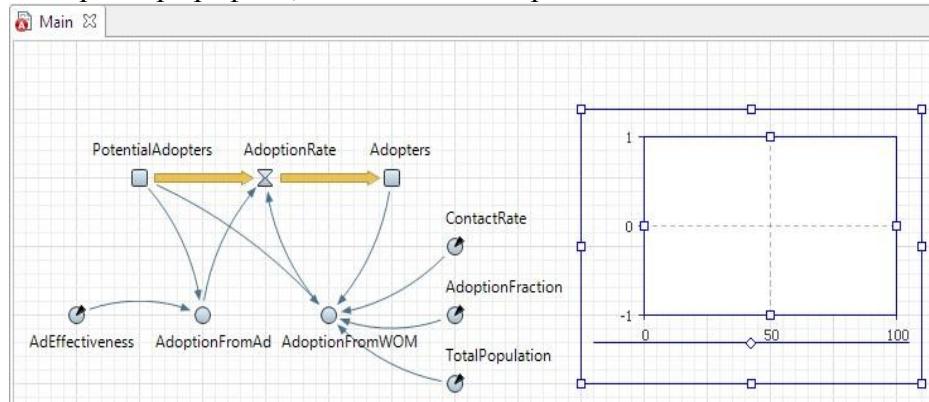


Рис. 23

Перейдите на вкладку **Основные** панели **Свойства**. В поле **Временной диапазон** задайте диапазон временной оси диаграммы – 8.

Диаграмма будет отображать график только для заданного временного интервала.

Добавьте элементы данных, историю изменения значений которых вы хотите отображать на временном графике: щелкните мышью по кнопке **Добавить элемент данных**. Введите в поле **Выражение** имя соответствующего накопителя – *PotentialAdopters*. В поле **Заголовок** введите *Potential adopters*. Данная строка будет отображаться в легенде диаграммы для этого элемента данных. Выберите первую опцию из выпадающего списка **Стиль маркера**, чтобы наносимые на кривую графика точки не отображались дополнительными точками – маркерами.

Добавьте на график еще один элемент данных, который будет отображать значение накопителя *Adopters*.

15. Добавьте график, отображающий изменение интенсивности продаж. Для этого добавьте на диаграмму еще один временной график, поместите его под добавленным ранее графиком. Измените свойства графика. В качестве **Выражения** должно быть задано имя потока *AdoptionRate*.
16. Запустите модель. Первая диаграмма показывает, как изменяются переменные *PotentialAdopters* и *Adopters* во время «прогона» модели. Они представляют собой классические S-образные кривые (рис. 24).

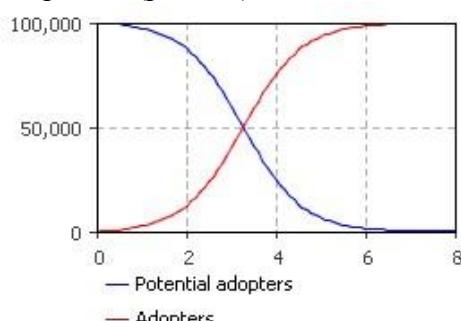


Рис. 24

На втором графике (рис. 25) вы увидите классическую колоколообразную кривую.

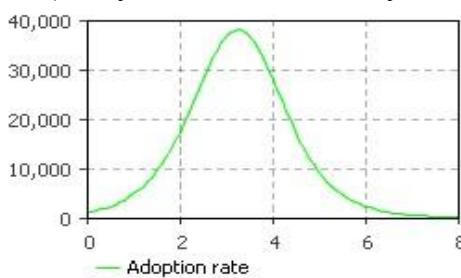


Рис. 25

Проанализируйте характеристики модели для своего варианта. Поместите графики и диаграммы в отчет. Сделайте выводы.

Вариант	Эффективность рекламы	Сила убеждения	Длительность процесса
1	0,010	0,01	6
2	0,010	0,02	7
3	0,010	0,03	8
4	0,005	0,04	9
5	0,005	0,05	10
6	0,005	0,01	11
7	0,015	0,02	10
8	0,015	0,03	9
9	0,015	0,04	8
10	0,010	0,05	7
11	0,010	0,01	6
12	0,010	0,02	7
13	0,005	0,03	8
14	0,005	0,04	9
15	0,005	0,05	10
16	0,015	0,01	11
17	0,015	0,02	10
18	0,015	0,03	9
19	0,020	0,04	8
20	0,020	0,05	7

Задача 2. Расширение модели жизненного цикла продукта

Расширенная модель поможет спланировать стратегию выпуска продукта на рынок, сориентироваться на конкретного потребителя и спрогнозировать спрос на продукт для того, чтобы выработать более рациональную и эффективную рекламную стратегию.

Моделирование повторных покупок

Созданная модель не учитывает того, что со временем продукт может быть израсходован или прийти в негодность, что вызовет необходимость его повторного приобретения. Смоделируем повторные покупки, полагая, что потребители продукта снова становятся потенциальными потребителями, когда продукт, который они приобрели, становится непригоден.

Определите константу, задающую среднее время жизни продукта, – *ProductLifeTime* (рис. 26). Пусть средняя продолжительность использования нашего продукта равна двум годам.

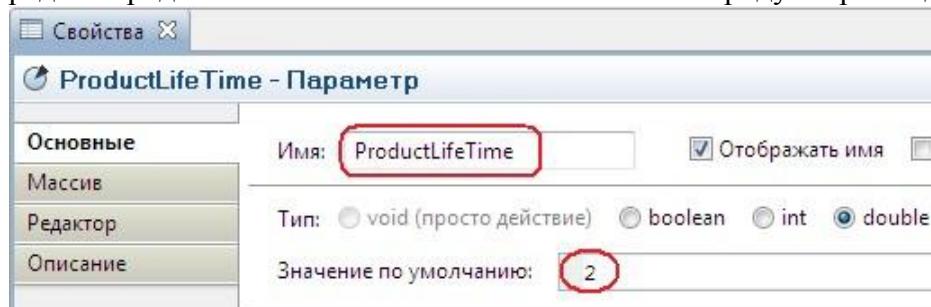


Рис. 26

Потребители продукта снова становятся потенциальными потребителями тогда, когда продукт, который они приобрели, расходуется и перестает использоваться. Поэтому поток прекращения использования продукта является ничем иным, как потоком приобретения, задержанным на среднее время пригодности продукта.

Создайте поток прекращения использования продукта, ведущий из *Adopters* в *PotentialAdopters* (сделайте двойной щелчок мыши по накопителю *Adopters*, а потом щелкните по накопителю *PotentialAdopters*). Назовите поток *DiscardRate* (нажмите **Ctrl+Enter** сразу после того, как вы закончите вводить новое имя). Формулы накопителей должны будут выглядеть следующим образом (рис. 27).

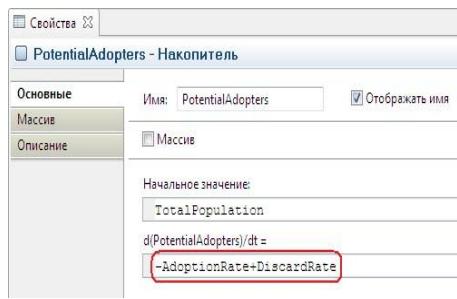
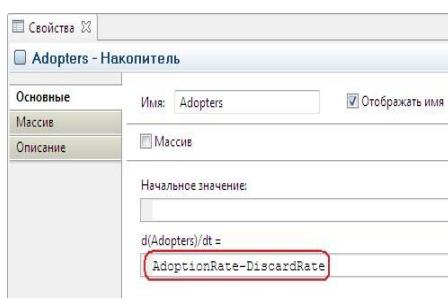
	
---	--

Рис. 27

Поместите поток на структурной диаграмме над потоком приобретения продукта (рис. 28).

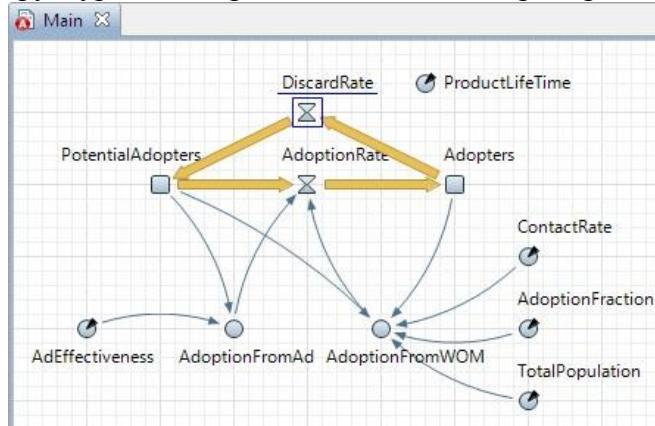


Рис. 28

Задайте следующую формулу для потока *DiscardRate*:

$\text{delay}(\text{AdoptionRate}, \text{ProductLifeTime})$

Функция *delay()* реализует временную задержку; она имеет следующую нотацию:
 $\text{delay}(<\text{задерживаемый поток}>, <\text{значение задержки}>, <\text{начальное значение}>)$

В нашем случае функция представляет собой *AdoptionRate* с временной задержкой *ProductLifeTime*. Пока не истекло время использования первого приобретенного продукта, поток равен нулю.

Проверить работу функции задержки проще всего с помощью диаграммы. Для этого добавьте на график, отображающий динамику изменения интенсивности продаж, еще одну величину – интенсивность отказа от продукта, определяемую нашим потоком *DiscardRate*.

Запустите модель. Проверьте, как работает функция задержки (рис. 29). На диаграмме видно, что поток прекращения использования продукта является потоком приобретения продукта, задержанным на 2 года – время пригодности продукта.

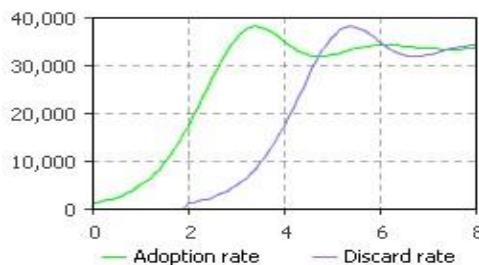


Рис. 29

С помощью другой диаграммы (рис. 30) проследите динамику изменения численностей потребителей.

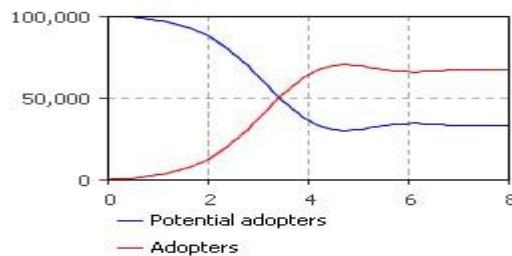


Рис. 30

Теперь численность потенциальных потребителей не уменьшается до нуля, а постоянно пополняется по мере того, как потребители заново покупают продукты взамен непригодных. Интенсивность приобретения продукта растет, падает и в итоге принимает какое-то значение, зависящее от средней жизни продукта и параметров, которые определяют интенсивность этого потока. Наличие в модели прекращения использования продукта означает, что какая-то доля населения всегда будет оставаться потенциальными потребителями.

Задача 3. Моделирование цикличности спроса

В текущей модели процент контактов потребителей продукта с потенциальными потребителями, который приводит к продажам продукта, полагается постоянным. На самом деле он изменяется, поскольку спрос на наш продукт зависит от текущего времени года. Продукт пользуется наибольшим спросом летом, зимой спрос на товар резко падает, за исключением небольшого предпраздничного периода в декабре. Необходимо смоделировать сезонную цикличность спроса.

Предположим, что у вас есть экспериментальные данные того, как изменяется средний спрос на продукт в течение года. Добавьте эти данные в модель с помощью табличной функции. Табличная функция – это функция, заданная в табличной форме, которая может быть сделана непрерывной с помощью интерполяции и экстраполяции.

- Промоделируйте кривую спроса табличной функцией. Создайте новую табличную функцию. В появившемся окне свойств задайте имя табличной функции – *demand*. На вкладке **Общие** окна свойств функции введите **Данные** функции. В поле **Аргумент** введите значение нового аргумента функции, в поле **Функция** введите значение функции для этого аргумента. Задайте параметры функции, как на рис. 31.

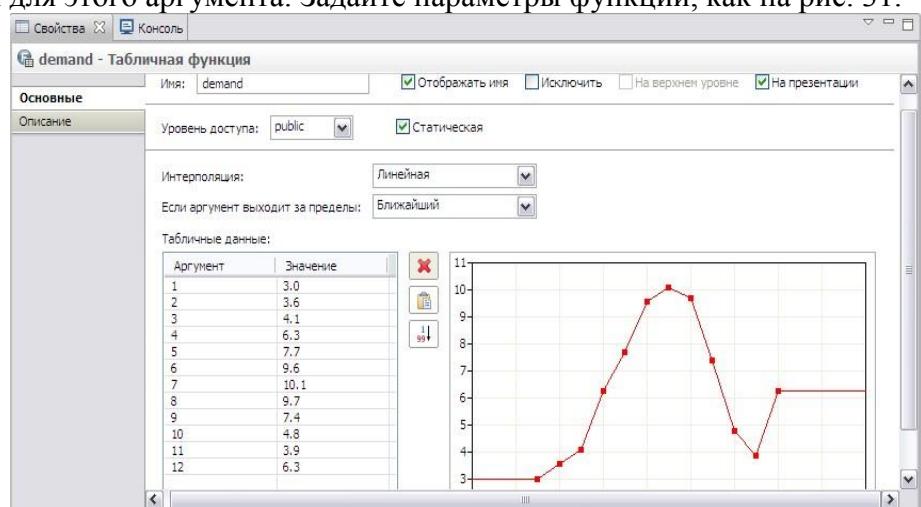


Рис. 31

Задайте тип интерполяции – **Линейная интерполяция** из группы элементов **Интерполяция**. Интерполяция будет производиться соединением табличных точек линиями.

Задайте тип реакции на аргументы, лежащие за пределами области допустимых значений функции. Выберите элемент **Использовать ближайший корректный** из группы элементов **Если аргумент выходит за пределы**. Щелкните мышью по кнопке **График....** В появившемся диалоговом окне будет показан получившийся график кривой спроса.

2. Промоделируйте, как спрос влияет на интенсивность приобретения продукта. Для этого создадим специальную математическую функцию и заменим параметр *AdoptionFraction* вспомогательной переменной, значение которой будет считаться этой функцией.

Создайте новую функцию. В появившемся диалоговом окне задайте имя функции *adoptFraction*. Убедитесь, что функция возвращает значение типа *real*.

Создайте аргумент, передающий функции текущее значение времени. В таблице **Аргументы** добавьте аргумент и назовите его *time*. Оставьте выбранный по умолчанию тип аргумента *double*. Задайте выражение функции. В поле редактирования **Выражение** введите

```
Return (demand((time-floor(time))*12+1)/200.0); .
```

Это выражение вычисляет номер текущего месяца и передает его табличной функции *demand*. Табличная функция возвращает значение спроса на продукт для данного месяца. В заключение для получения значения доли людей, покупающих продукт под влиянием общения, значение спроса делится на коэффициент преобразования. Функция *floor()* является предопределенной функцией AnyLogic.

Вводя выражения, пользуйтесь «Мастером Функций», в котором предопределенные функции присутствуют наряду с переменными, аргументами функций и табличными функциями модели. За детальным описанием функций и их параметров обращайтесь к руководству пользователя или справочнику классов (нужно смотреть методы класса *Func*).

3. Замените константу, задающую силу убеждения потребителей продукта, вспомогательной переменной, значение которой будет вычисляться по созданной функции.

Удалите параметр *AdoptionFraction*. Создайте вспомогательную переменную *AdoptionFraction*. Задайте формулу *adoptFraction(time())*. Теперь значение вспомогательной переменной будет вычисляться нашей функцией. Функция принимает один аргумент – *time()*, т.е. текущее модельное время.

4. Задайте останов модели в момент времени 25 и запустите модель. Теперь поведение модели колеблется около точки равновесия в силу того, что колеблются значения потока приобретения и потока прекращения использования продукта (рис. 32).

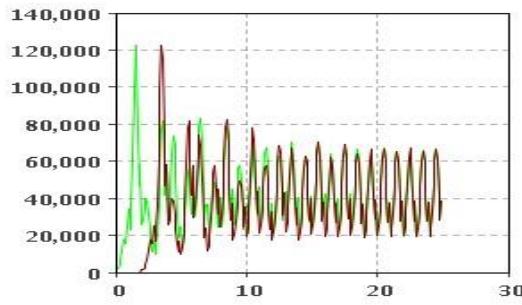


Рис. 32

Задача 4. Моделирование стратегии рекламной кампании

На данный момент эффективность рекламы в модели полагается постоянной. На самом деле она зависит от текущих расходов компании на рекламу. Улучшим нашу модель, чтобы иметь возможность управлять расходами на рекламную кампанию. Изменяя месячные расходы на рекламу, можно будет влиять на текущую эффективность рекламы.

1. Создайте константу, задающую месячные расходы компании. Назовите параметр *MonthlyExpenditures*. Установите значение по умолчанию – 1100.
2. Замените константу *AdEffectiveness* вспомогательной переменной. Удалите параметр *AdEffectiveness*. Создайте простую переменную *AdEffectiveness* с формулой *MonthlyExpenditures/10000.0*. Мы полагаем, что именно так эффективность рекламы зависит от текущих рекламных расходов компании.

Затем необходимо вести статистику всех расходов компании. Это может быть сделано созданием специальной переменной для хранения информации о том, сколько денег было потрачено на рекламу продукта. Каждый месяц таймер будет обновлять значение переменной, добавляя значение запланированных на предстоящий месяц расходов на рекламную кампанию продукта.

3. Добавьте статическую переменную (используйте палитру **Основные**) *TotalExpenditures*. Убедитесь, что у переменной нет уравнения. Задайте начальное значение – 0.0.
4. Создайте таймер для обновления значения переменной *TotalExpenditures*, щелкнув мышью по кнопке панели инструментов **Событие** и поместив его на структурную диаграмму. В окне **Свойства** назовите таймер *monthlyTimer*. Сделайте так, чтобы событие срабатывало каждый месяц. Убедитесь, что событие **Циклическое**. Поскольку одна единица модельного времени в нашей модели соответствует одному году, то одному месяцу будет соответствовать выражение $1.0/12.0$. Введите $1.0/12.0$ в поле **Таймаут**. Установите **Время первого срабатывания** = 0 (будет срабатывать при старте). В поле **Действие вве^{дите}** *TotalExpenditures+=MonthlyExpenditures*; Этот код будет выполняться каждый раз по истечении таймаута таймера. Он выполняет сбор статистики, а именно: добавляет значение запланированных рекламных расходов на предстоящий месяц к значению переменной *TotalExpenditures*. Поскольку реклама играет значительную роль только в начальной стадии процесса завоевания рынка, постольку необходимо в какой-то момент времени, скажем, через 3 года, остановить рекламную кампанию. В результате приостановки кампании мы сэкономим деньги, бесцельно тратящиеся на рекламу тогда, когда насыщение рынка будет определяться практически исключительно покупками продукта, вызванными общением потребителей с потенциальными потребителями.
5. Добавьте константу, задающую время переключения, с именем *SwitchTime* и значением параметра по умолчанию 3.0.
6. Создайте стейтчарт для моделирования рекламной стратегии.

Для того чтобы создать новый стейтчарт, выберите кнопку панели инструментов **Диаграмма**. Нарисуйте следующий стейтчарт (рис. 33).

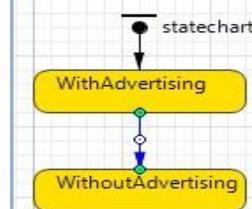


Рис. 33

Создайте первое состояние стейтчарта, щелкнув мышью по кнопке **Состояние**, переименуйте состояние в *WithAdvertising*. Добавьте еще одно состояние под только что созданным. Назовите его *WithoutAdvertising*. Когда стейтчарт перейдет в это состояние, мы должны будем остановить рекламную кампанию. Для реализации этого введите *MonthlyExpenditures=0.0*; в поле **Действие при входе**.

Добавьте переход из состояния *WithAdvertising* в состояние *WithoutAdvertising* (рис. 34). Укажите, что переход произойдет по истечении времени *SwitchTime*: в свойствах перехода выберите **По таймауту** из выпадающего списка **Происходит** и введите *SwitchTime* в поле **Таймаут**.

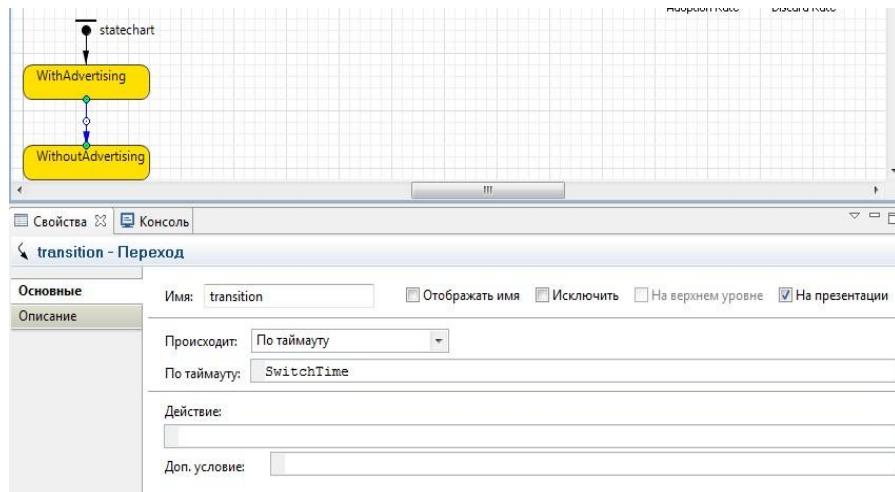


Рис. 34

Теперь, когда стейтчарт находится в начальном состоянии *WithAdvertising*, рекламные расходы кампании определяются переменной *MonthlyExpenditures*. Как только стейтчарт покидает данное состояние в момент времени *SwitchTime*, компания перестает рекламировать продукт.

7. Запустите модель и убедитесь, что рекламная кампания длится только три года.

Задача 5. Оптимизация рекламной стратегии

Рыночная стратегия в данной модели предельно проста: в определенный момент времени компания прекращает рекламировать продукт. Мы же хотим найти оптимальную рыночную стратегию для достижения требуемого количества потребителей к определенному моменту времени при минимальных затратах на рекламу. Можно решить эту проблему, используя оптимизацию, при которой выбранные параметры модели будут систематически изменяться для минимизации или максимизации значения целевого функционала.

1. Определите константу, задающую необходимый порог насыщения рынка, например 80 процентов от общей численности населения. Назовите параметр *ExpectedSaturation* и задайте значение по умолчанию *TotalPopulation*0.8*.
2. Добавьте константу, задающую момент времени, когда должно быть достигнуто требуемое количество потребителей. Назовите параметр *SaturationTime* и задайте значение по умолчанию 1.5.
3. Измените стейтчарт, чтобы выполнить проверку насыщения рынка.

Откройте диаграмму стейтчарта. Измените стейтчарт так, чтобы он выглядел как на рис. 35.

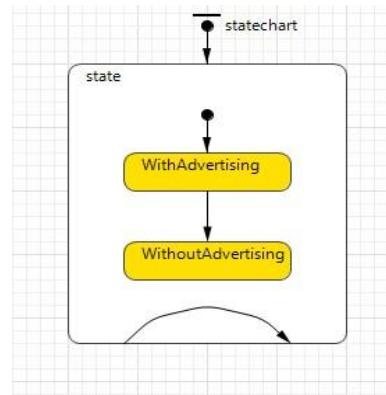


Рис. 35

Добавьте сложное состояние, включающее в себя два существующих состояния. Добавьте еще один указатель начального состояния, указывающий на сложное состояние. Добавьте внутренний переход в сложное состояние. Сделайте так, чтобы переход выполнял проверку насыщения рынка продукта по истечении времени *SaturationTime*. Для

этого выберите **По таймауту** из выпадающего списка **Происходит** и введите *Saturation Time* в поле **Таймаут**. В поле **Доп. условие** введите: *Adopters <= ExpectedSaturation*. Это условие проверяет, достигнуто ли необходимое число потребителей. Если выражение, заданное в поле **Доп. условие** истинно, то происходит переход и выполняется код, указанный в поле **Действие**. В поле **Действие** введите *TotalExpenditures=50000*; Мы увеличиваем значение переменной *TotalExpenditures* для того, чтобы показать, что наше требование не было выполнено.

4. Создайте оптимизационный эксперимент AnyLogic (рис. 36).

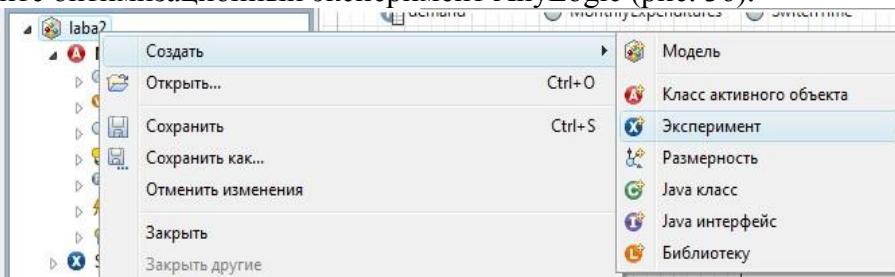


Рис. 36

В появившемся диалоговом окне задайте имя эксперимента и выберите **Оптимизационный эксперимент** как тип нового эксперимента.

5. Настройте созданный эксперимент (рис. 37-38).

На вкладке **Модельное время** окна свойств эксперимента задайте **Стоп по времени** 1.6.

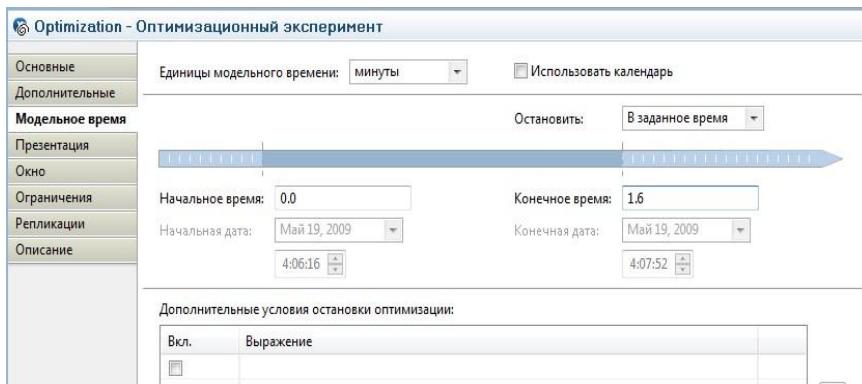


Рис. 37

На вкладке **Основные** задайте число «прогонов» модели 500.

Мы хотим минимизировать деньги, затраченные на рекламу продукта. На вкладке **Основные** выберите переменную *TotalExpenditures* в качестве **Целевого функции** и убедитесь, что установлен флагок **Минимизировать**.

Измените оптимизационные параметры в таблице **Параметры** на вкладке **Основные** окна свойств:

- тип параметра *SwitchTime* на непрерывный и установите максимальное 1.5 и минимальное 0.0 значения;
- тип параметра *MonthlyExpenditures* на непрерывный и установите максимальное 1500 и минимальное 1000 значения.

Во время оптимизации, значения параметров модели будут систематически изменяться, чтобы определить наименьшее значение переменной *TotalExpenditures*, выбранной в качестве целевого функционала.

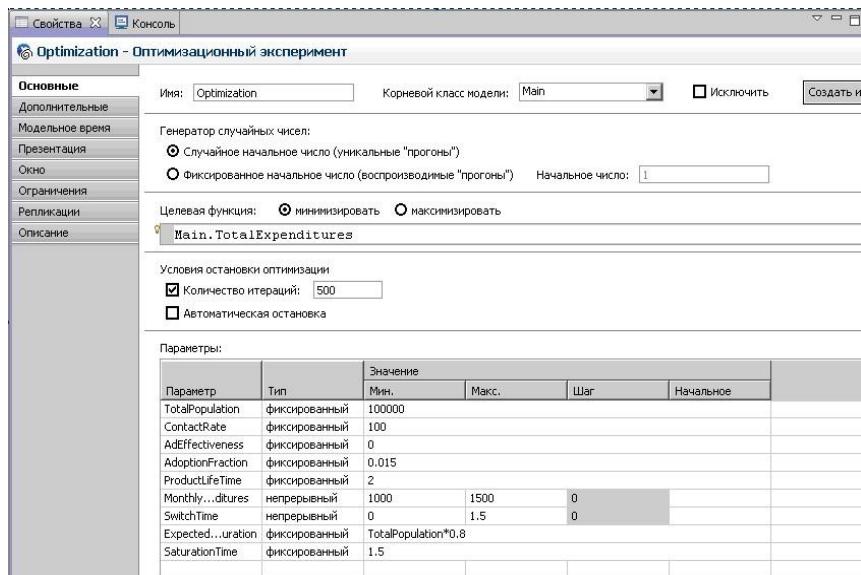


Рис. 38

Нажмите кнопку **Создать интерфейс** – будет создан стандартный интерфейс для данного эксперимента.

6. Запустите модель, выбрав оптимизационный эксперимент. AnyLogic запустит модель 500 раз с различными значениями параметров *MonthlyExpenditures* и *SwitchTime*. Итоговая статистика оптимизации отображается в окне оптимизационного эксперимента.

На рис. 39 видно, что значение функционала примерно равно 11 250, наилучшие значения оптимизационных параметров следующие: *SwitchTime* = 0.75; *MonthlyExpenditures* = 1 250.

lab2 : Optimization

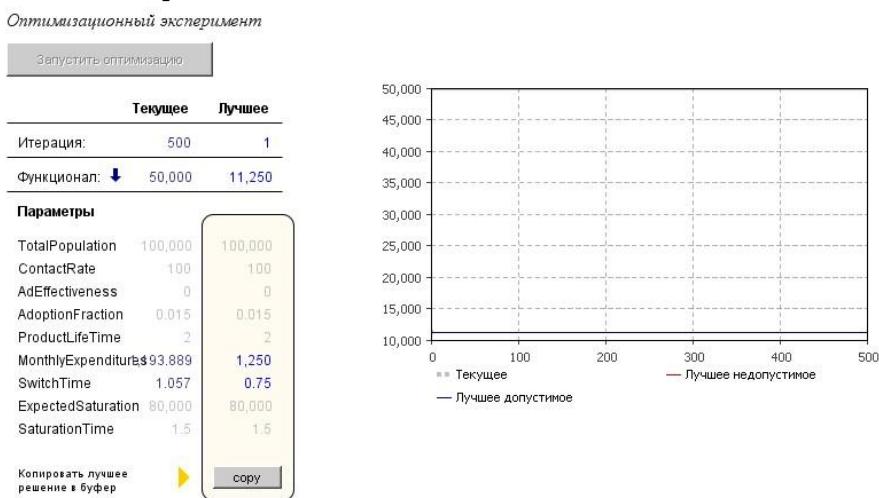


Рис. 39

Сделайте вывод о том, как спланировать стратегию завоевания рынка, чтобы рекламная кампания была наиболее рациональной и эффективной.

7. Примените результаты оптимизации.

В окне **Оптимизация** щелкните мышью по кнопке **copy**. В появившемся диалоговом окне выберите эксперимент, в который вы хотите скопировать результаты оптимизации. Оставьте выбранный по умолчанию эксперимент *Simulation* и щелкните мышью по кнопке **OK**.

Запустите модель с оптимизированными значениями параметров для того, чтобы убедиться в том, что к моменту времени *SaturationTime* будет достигнуто требуемое количество потребителей.

Результаты работы

Студент должен предоставить отчет по лабораторной работе с выводами, продемонстрировать работу модели, ответить на вопросы преподавателя.