

# ПРАКТИЧЕСКАЯ РАБОТА № 6

## Системная динамика

### *Общие сведения*

#### **Цель работы**

Приобрести навыки моделирования функционирования сложных систем с помощью методов системной динамики и пакета iThink.

#### **План выполнения**

1. Изучить теоретическую часть;
2. Получить задание преподавателя;
3. Выполнить задание 1:
  - 3.1. Построить модель, указанную в задании. Модель должна содержать графики изменения емкости каждого из резервуаров;
4. Составить отчёт по лабораторной работе. Отчёт должен иметь следующую структуру:
  - 4.1. Титульный лист, который должен содержать следующую информацию:
    - 4.1.1. Название университета и кафедры, ответственной за дисциплину;
    - 4.1.2. Заголовок — номер и название лабораторной работы;
    - 4.1.3. Подзаголовок — номер варианта и номера задач;
    - 4.1.4. ФИО и должности студента и преподавателя;
  - 4.2. Основная часть, содержащая следующее информационное наполнение:
    - 4.2.1. Формулировка индивидуального задания;
    - 4.2.2. Ответы на вопросы задания;
    - 4.2.3. При необходимости, снимки экрана монитора, содержащие основные моменты решения задачи;
    - 4.2.4. Результаты решения и выводы.

## *Теоретическая часть*

### **Системная динамика**

**Системная динамика** — направление в изучении сложных систем, исследующее их поведение во времени, в зависимости от структуры элементов системы и взаимодействия между ними. Компьютерному моделированию подобных систем в современной науке уделяется не малое внимание.

Большинство систем, которые в настоящее время привлекают внимание исследователей, достаточно сложны. Соответственно сложными становятся и их математические модели, а создание подобных моделей требует значительной математической подготовки. Для специалистов из многих отраслей это является серьезной проблемой. Получается, что математические модели способны создавать лишь немногие избранные, в то время как остальным отводится роль пользователей или, в лучшем случае, консультантов.

С этой проблемой столкнулся Джей Форрестер во время преподавания в Слоановской бизнес-школе при МТИ. Результатом его работ конца 1950-х гг. стала системная динамика — новый «язык», описывающий поведение сложных систем, в зависимости от их структуры и взаимодействий (обратных связей, задержек реакции, влияния среды и др.). Системная динамика позволяет разрабатывать не только «формульное», но и визуальное представление системы, схему ее устройства.

Для изучения основных понятий системной динамики в настоящей лабораторной работе будет использоваться программный комплекс iThink.

### **Программный комплекс iThink**

Существует множество аналогичных по своим целям и возможностям программных средств, которые могут быть использованы для решения тех или иных задач, возникающих в экономической практике. Одной из наиболее важных задач современной экономической практики является управление финансовыми потоками, циркулирующими на предприятии, в холдинге, между банком и его клиентами. Специфика

«потокового» подхода к управлению заключается в планировании и контроле перемещений финансовых и материальных ресурсов, при этом в виде потока можно рассматривать не только движение денежных средств, но и функционирование предприятия в целом. Оперативное и эффективное управление этими потоками весьма затруднено будучи неавтоматизированным. Средства потокового управления в настоящий момент на рынке программных средств представлены в виде специализированных систем имитационного моделирования и экспертных систем. Одной из таких систем является программный комплекс iThink от компании iSeeSystems.

### **Назначение iThink**

iThink — мощное средство имитационного моделирования производственных и финансовых проектов и процессов, предназначенный для следующих наиболее важных групп пользователей:

1. *инвестиционных компаний, брокеров, дилеров ценных бумаг.* эта группа пользователей с помощью Ithink осуществляет планирование инвестиционных операций, прогнозирование рыночной конъюнктуры и доходности вложений;
2. *аналитических отделов банков и финансово-промышленных групп.* в аналитических отделах банков – самая широкая сфера применения имитационных моделей: прогнозирование, «обкатка» структурных схем и инвестиционных проектов, выбор и обоснование оптимальной стратегии. Ithink – это экспертная программа, пригодная для решения задач планирования и управления финансово-промышленной группой;
3. *отделов проектных исследований банка.* Ithink способен обеспечить достаточно глубокую детализацию проектных документов, например, плановые графики и таблицы могут быть сделаны в нужном временном масштабе;
4. *консультационных и проектных компаний.* Ithink позволяет наглядно продемонстрировать суть рекомендаций и последствия их реализации. Так, схема автоматизации бухгалтерии и управленческого учета может быть предварительно просчитана на имитационной модели;
5. *региональных органов власти.* С помощью Ithink возможно моделирование региональных экономик: планирование хозяйственных систем областного, городского и муниципального масштаба. Пакет обеспечивает повышение эффективности управления процессами снабжения и распределения ресурсов. Возможно моделирование сезонных циклов, что актуально для сельского хозяйства. Ithink может

применяться для планирования социальной сферы и медицинских услуг. Актуальной задачей может быть прогнозирование налоговых сборов и управление местными бюджетами;

- б. «отраслевых» областей. Ithink может применяться для управленческого моделирования объектов топливно-энергетического комплекса, металлургических, химических и других предприятий, включая предприятия с непрерывным производственным циклом.

В процессе моделирования в окне модели формируется структурная схема модели из блоков встроенного графического языка моделирования. Между блоками устанавливаются взаимосвязи и генерируется программный код. Оператору требуется ввести функциональные зависимости и числовые параметры, после этого модель готова к запуску.

### **Основные блоки языка моделирования iThink**

В языке моделирования iThink используется пять основных блоков: **резервуар** (Stock), **поток** (Flow), **конвертер** (Converter), **коннектор** (ActionConnector), **процесс принятия решения** (ProcessDecisionDiamond).

*Резервуар*— количество ресурса, существующее в данный момент времени и измеряемое либо в денежных, либо в физических единицах (2 тысячи рублей, 5т макарон, 200 рейтинговых баллов и т.д.). Резервуар в Ithink изображается прямоугольником, который способен накапливать, аккумулировать единицы резервуара.

*Поток* - это процесс, протекающий непрерывно во времени, оценить который можно в физических или денежных единицах, соотнесенных с каким-либо временным интервалом (рубли в месяц, литры в час, стоимость акций на время закрытия биржи в данный день и т.д.). В Ithink поток изображается фигурой, состоящей из путепровода, вентиля, регулятора потока и указателя направления. По характеру использования, потоки подразделяются на ограниченные и неограниченные, однонаправленные и двунаправленные, конвертируемые и неконвертируемые.

*Конвертеры* в Ithink – преобразователи модельных единиц, которые изображаются окружностями. Они могут содержать значения констант или внешних входных переменных, подсчитывать значения алгебраических выражений или использоваться для хранения графических функций.

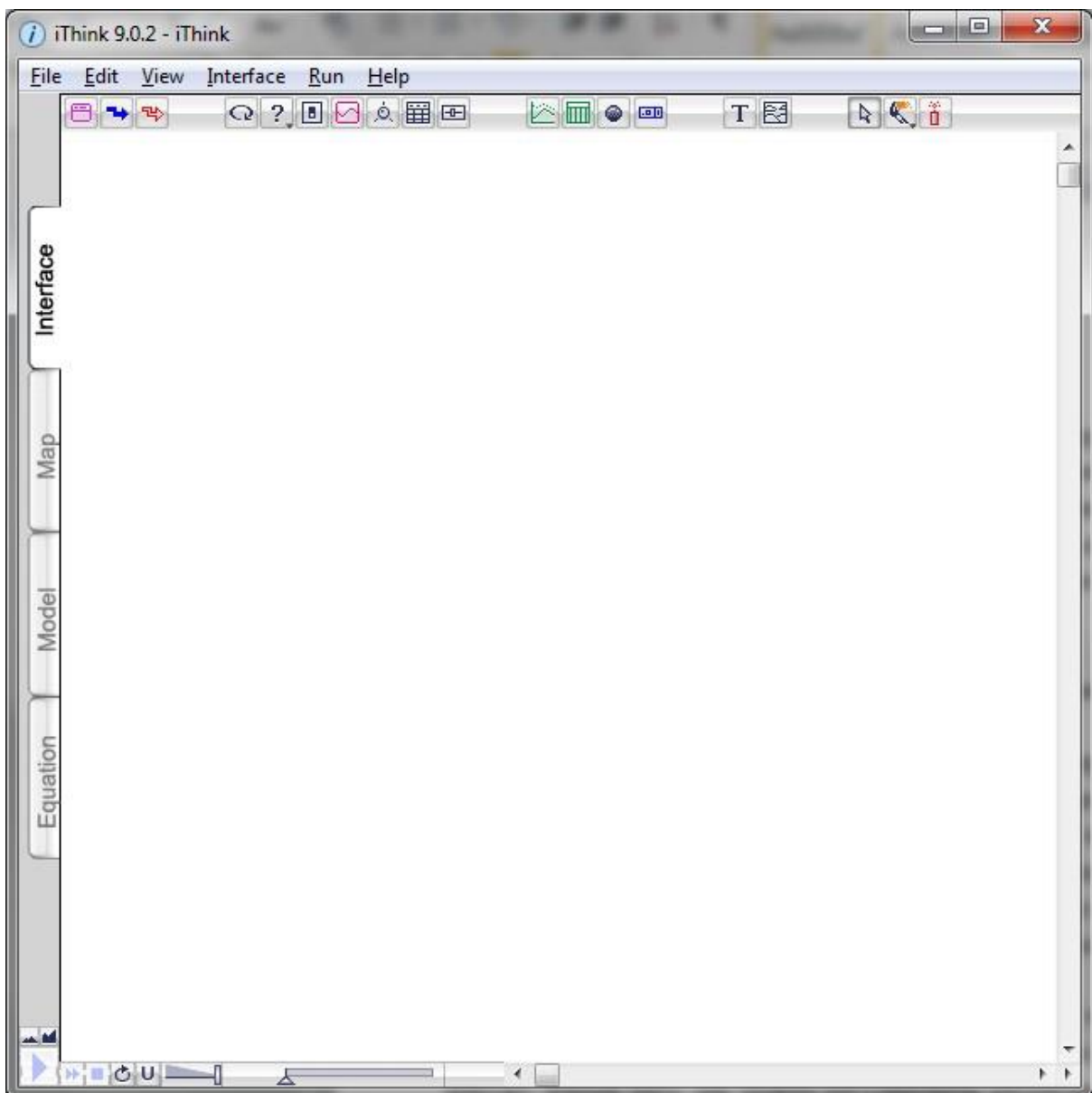
*Коннектор* предназначен для связи между собой элементов модели.

*Процесс принятия решения*— это механизм для управления запутанными схемами, связанный с представлением определения процессов внутри модели. С помощью этого блока можно скрыть сложность определенных операций.

## Уровни представления модели в iThink

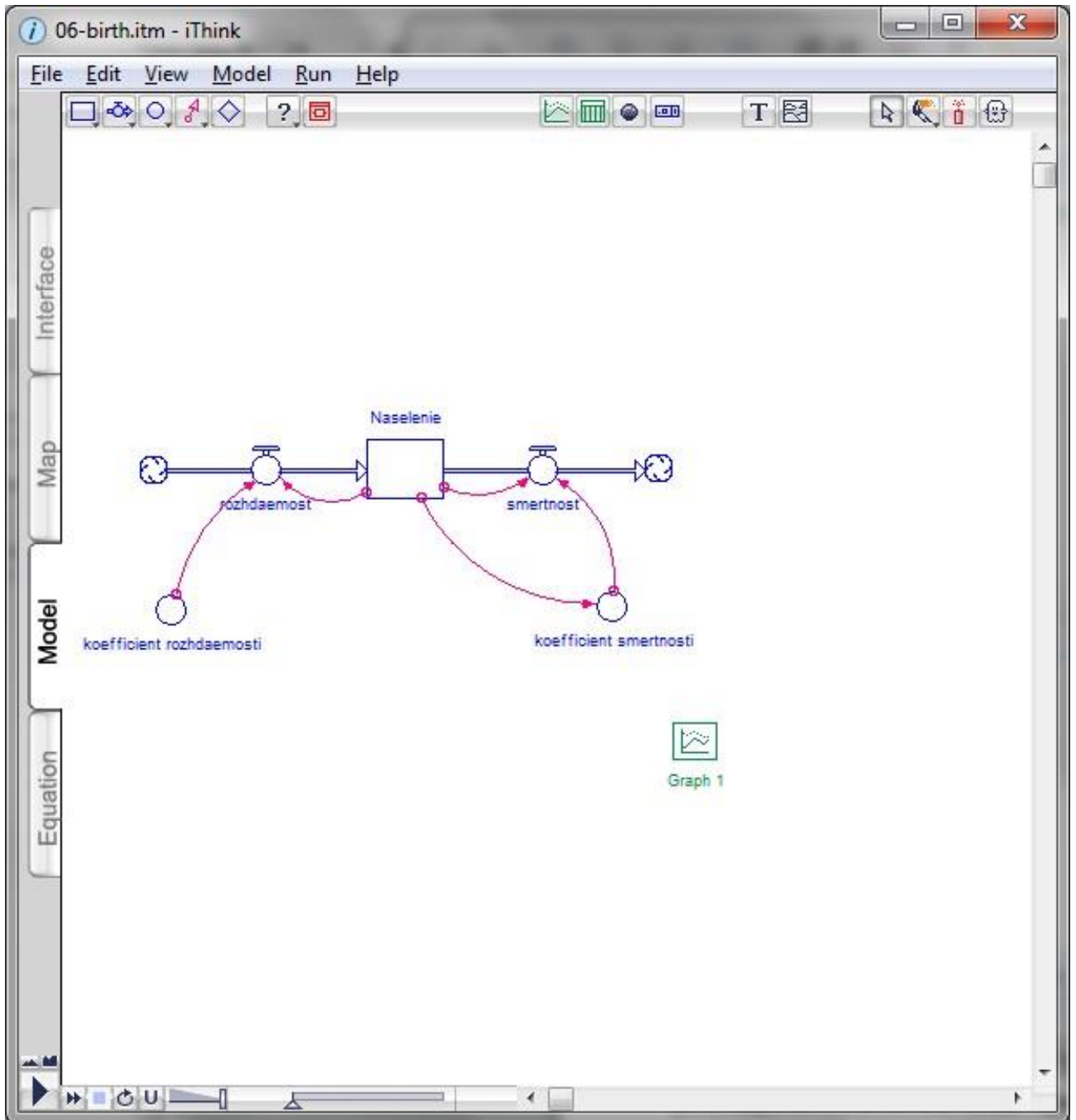
В iThink модели представляются тремя иерархическими уровнями: уровень интерфейса (высокоуровневое представление блок-схемой), уровень модели, уровень программного кода. Переключение производится путём выбора соответствующей вкладки в левой части окна программы.

**Уровень интерфейса.** На данном уровне производится работа с интерфейсом модели. Могут быть заданы элементы управления различными частями модели (см. Рисунок 76). В текущей лабораторной работе уровень интерфейса использоваться не будет.



**Рисунок 76. Окно iThink при активном уровне Интерфейса**

**Уровень модели** — базовый уровень, на котором строится модель при помощи потоковых схем (см. Рисунок 77).



**Рисунок 77. Модельный уровень представления в iThink**

**Уровень программного кода.** В результате создания модели на модельном уровне программный код на этом уровне генерируется автоматически (см. Рисунок 78).

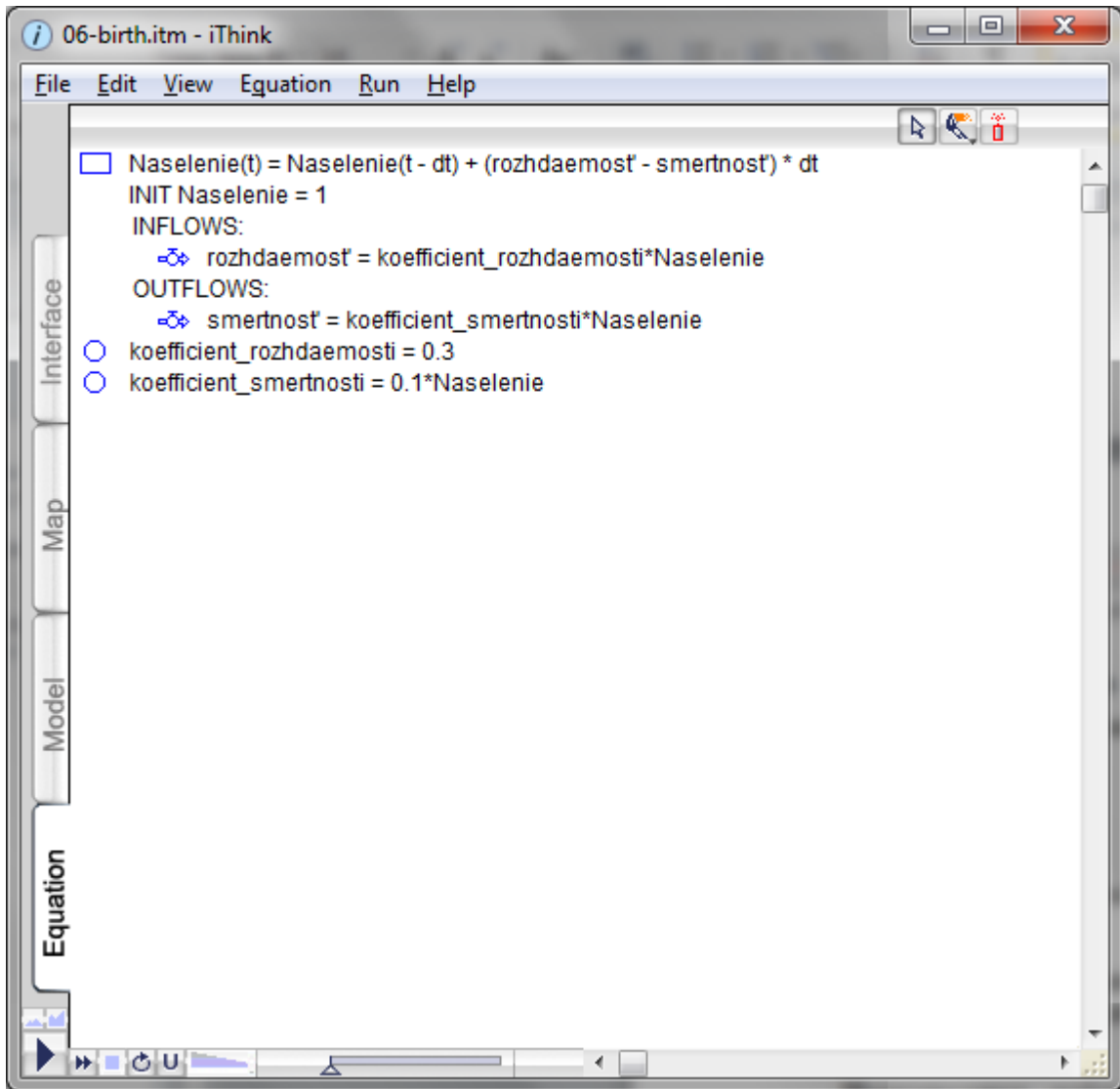


Рисунок 78. Представление на уровне программного кода в iThink

## Пример. Моделирование рождаемости

Для моделирования систем нам понадобятся блоки резервуара (stock) и потока (flow).

**Резервуары** (также называемые фондами) отражают существующие в системе накопления (количество товаров на складе, денег в банке). Представьте себе бассейн. Как и бассейн, резервуар может быть полон, частично заполнен или вообще пуст. В системной динамике (и в пакете iThink) резервуары представляются прямоугольниками. Резервуары могут накапливать все что угодно — воду, деньги, людей, мотивацию — как материальные, так и нематериальные объекты.

**Потоки** позволяют добавлять или отнимать что-либо из резервуаров. Поток напоминает трубу, через которую вода вливается в бассейн или

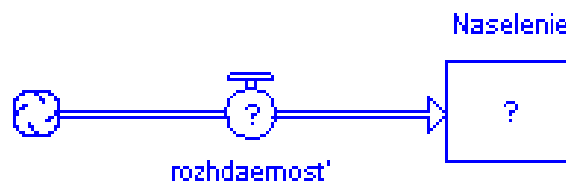
вытекает наружу. Кроме того, мы можем регулировать интенсивность потоков, также как регулируем количество воды при помощи крана.



**Рисунок 79. Фрагмент панели инструментов iThink. Слева направо кнопки создания: резервуаров, потоков, конвертеров, соединителей**

В качестве примера рассмотрим создание модели изменения численности населения. Заметим, что в iThink можно использовать русские имена элементов, однако мы в дальнейшем будем использовать транслитерацию.

Во вкладке **Model** создадим резервуар *Naselenie* и поток *rozdaemost'* с помощью панели инструментов (см. Рисунок 79), соединим их между собой и переименуем соответствующим образом. Результат показан на ниже (см. Рисунок 80, а также файл *ЛР7.Население1.itm*). Резервуар *Naselenie* представляет собой количество населения и пополняется за счет потока *rozdaemost'* — т.е. за счет рождения людей.



**Рисунок 80. Результат создания резервуара и потока**

«Облака», из которых начинаются (и которыми заканчиваются) потоки, представляют собой бесконечные источники (стоки) — источники (стоки), находящиеся за пределами системы.

Вопросительные знаки внутри элементов показывают, что необходимо задать соответствующие значения: начальное значение численности населения и скорость потока рождаемости.

Чтобы задать изменение потока *rozdaemost'* можно, например, ввести значение интенсивности этого потока (*koefficient rozhdapesti*) и связать его с *rozdaemost'* специальным элементом — соединителем (connector) отображаемым красными стрелками (см. Рисунок 81, файл *ЛР7.Население2.itm*).



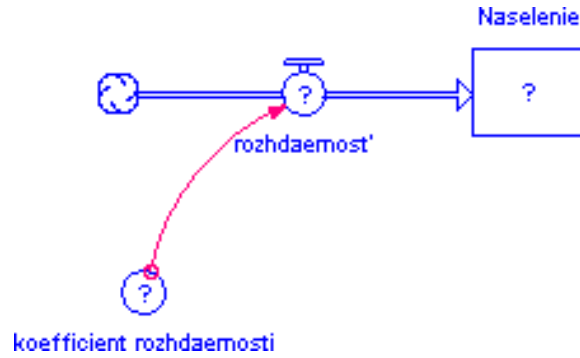


Рисунок 81. Задание интенсивности потока

Кружок, представляющий коэффициент рождаемости *koefficient rozhdadmosti* называется конвертером (converter). Конвертеры могут содержать константы или выражения и используются для модификации остальных частей модели.

Введем начальные значения элементов системы. Это можно сделать двойным щелчком по соответствующему элементу. Эта процедура показана ниже (см. Рисунок 82, Рисунок 83, Рисунок 84). Начальное значение  $Naselenie = 1$ ,  $rozhdadmost' = koefficient rozhdadmosti$ ,  $koefficient rozhdadmosti = 0.3$  (см. файл *ЛР7.Население3.itm*).

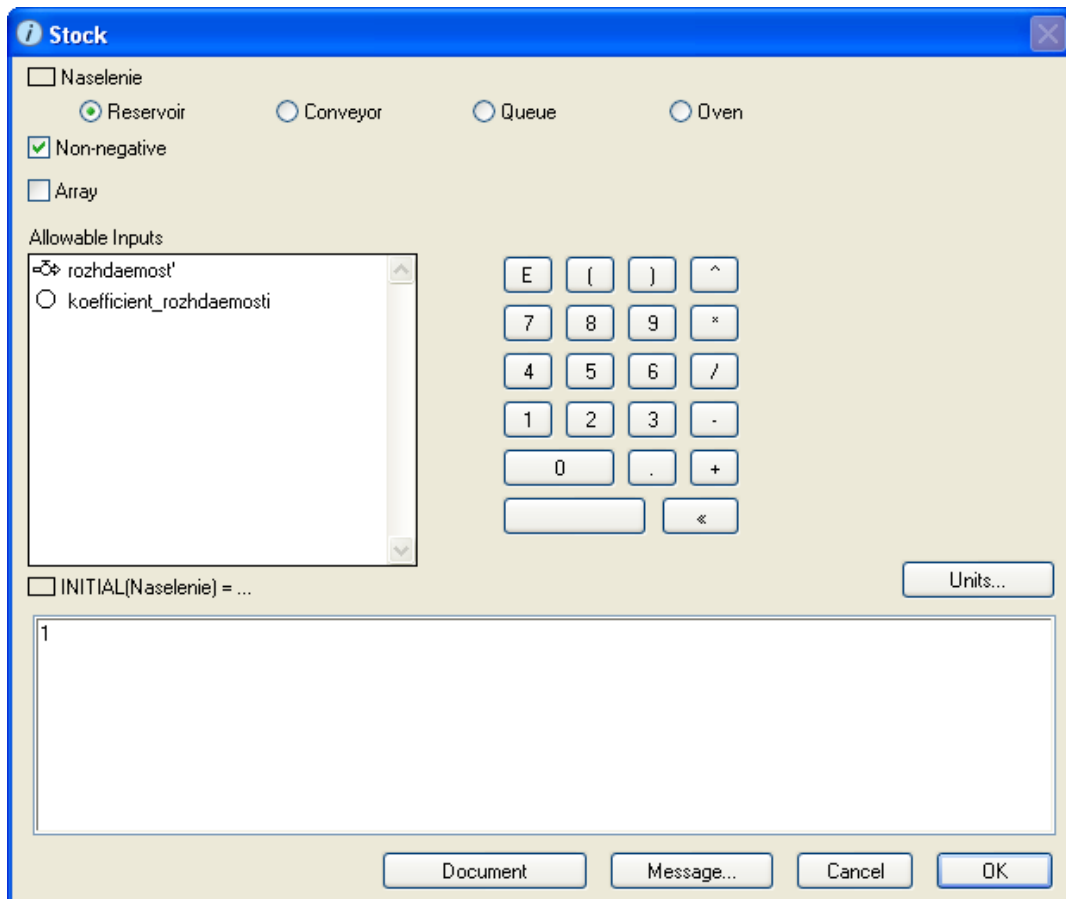
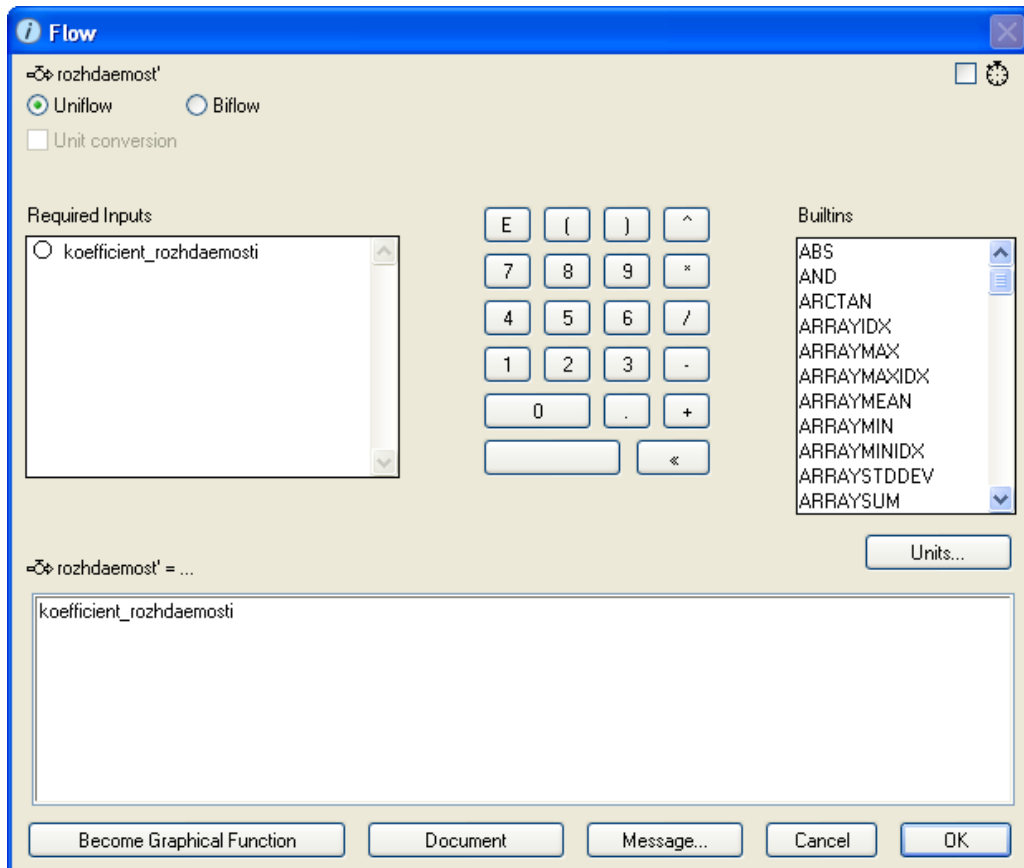
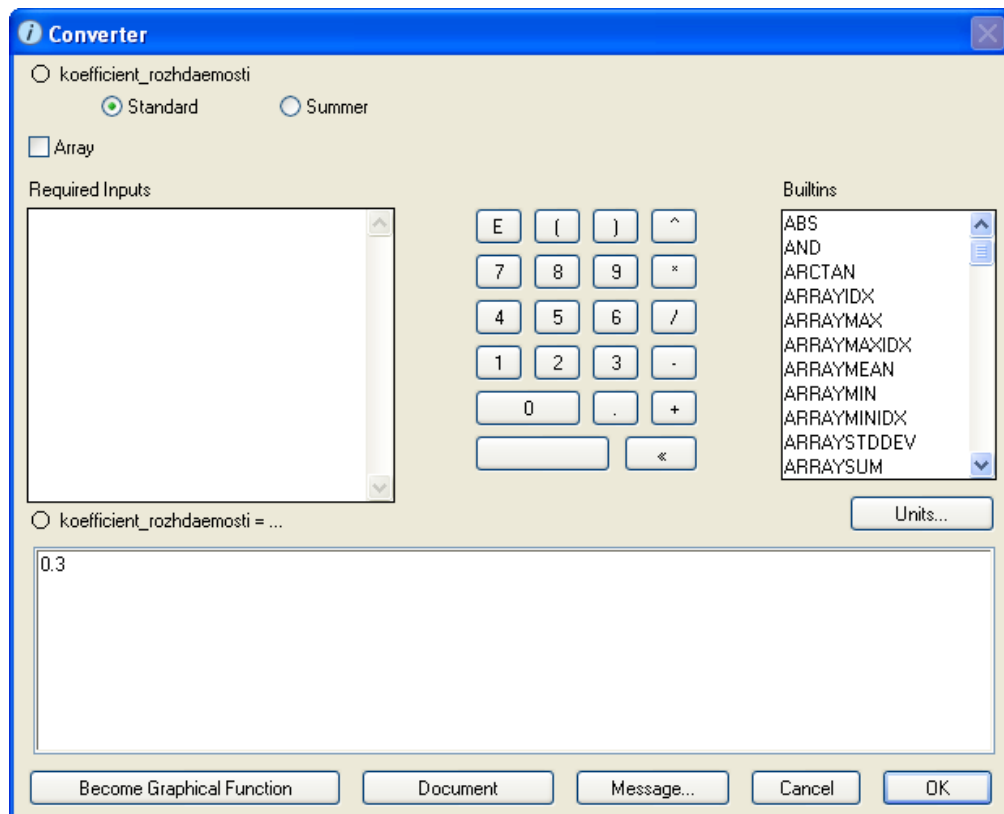



Рисунок 82. Ввод начальных значений элементов модели – 1

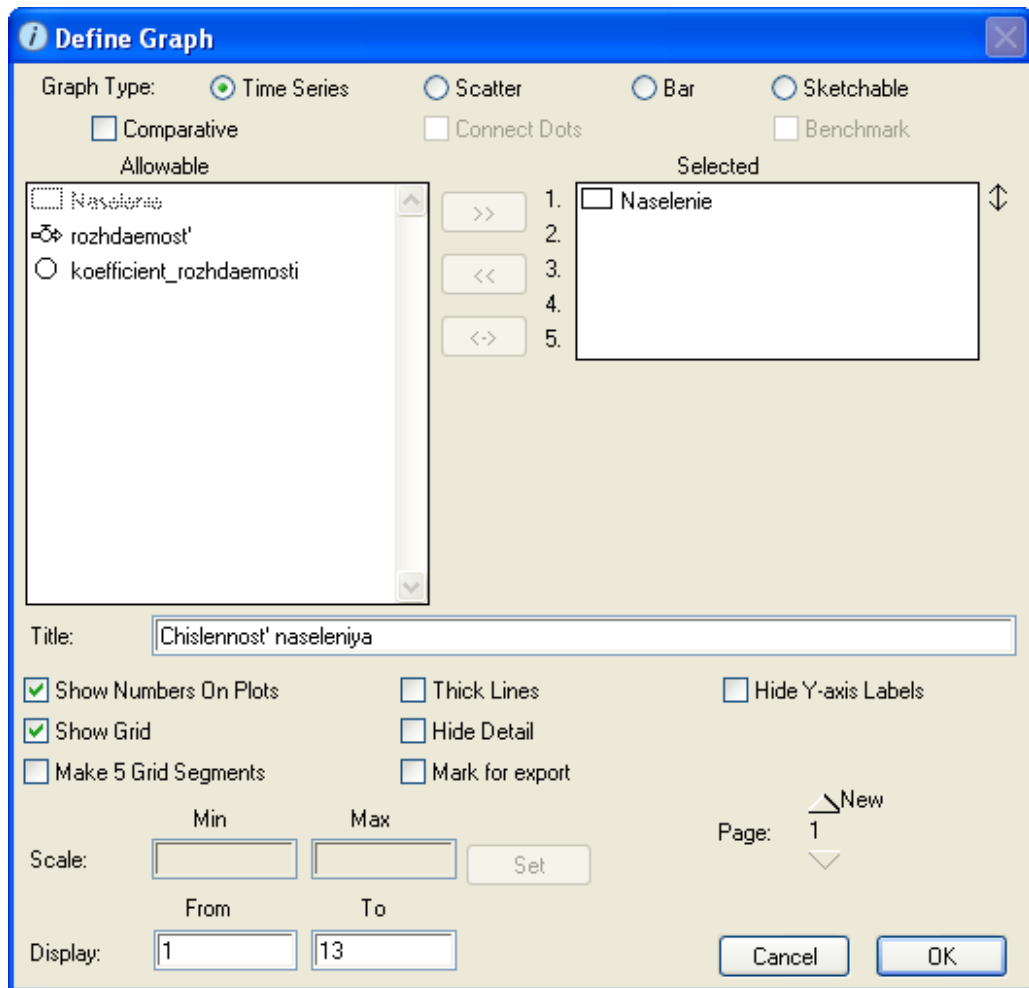


**Рисунок 83. Ввод начальных значений элементов модели – 2**




**Рисунок 84. Ввод начальных значений элементов модели – 3**

Построим график изменения численности населения во времени. Для этого воспользуемся кнопкой  панели инструментов. Двойной щелчок по графику позволяет настроить его параметры. Настройки, которые отображает Рисунок 85, позволяют вывести на графике содержимое резервуара *Naselenie* и задать заголовок графика — *Chislennost' naseleniya* (для остальных параметров используются их значения по умолчанию).



**Рисунок 85. Настройка графика для отражения содержимого резервуара**

После этого можно запустить модель на выполнение кнопкой  на панели запуска в нижней части окна iThink (см. файл *LP7.Население4.itm*).

Если люди рождаются с постоянной интенсивностью, то население будет расти линейно (см. Рисунок 86).

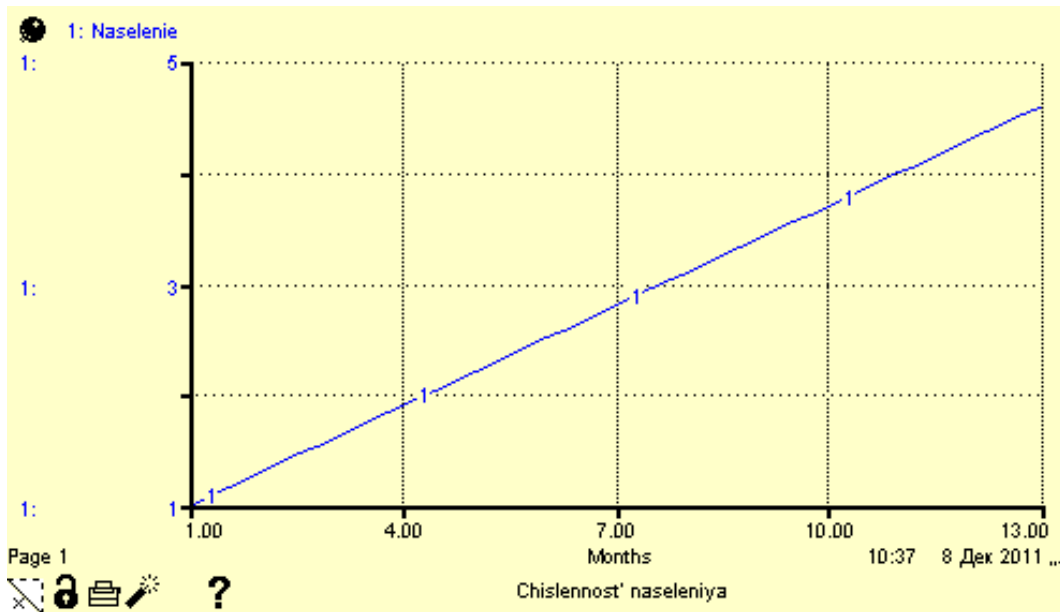


Рисунок 86. Линейный график рождения населения

Очевидно, что скорость изменения численности населения не является постоянной и зависит от текущего значения численности. Зададим эту связь, соединяя на схеме поток *rozdaemost'* с резервуаром *Naselenie*. Пусть также значение *rozdaemost'* будет равно: *koefficient rozhdapesti* \* *Naselenie* (см. Рисунок 87, Рисунок 88).

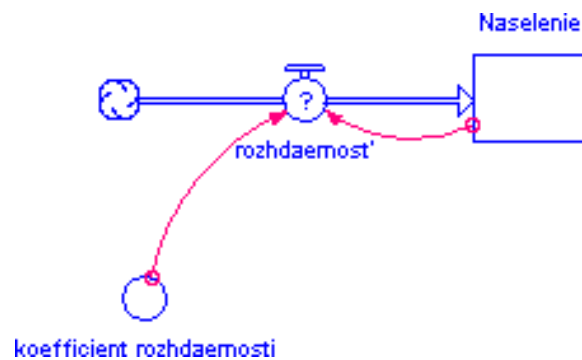


Рисунок 87. Задание скорости изменения численности населения – 1

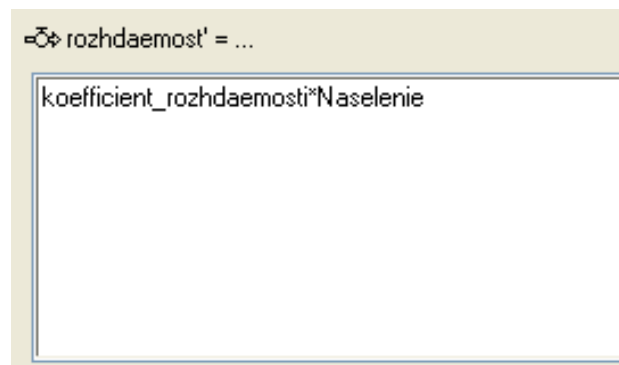


Рисунок 88. Задание скорости изменения численности населения – 2

Рисунок 89 отображает видим пример работы положительной обратной связи: количество населения увеличивается и это, в свою очередь, увеличивает прирост населения (см. файл *ЛР7.Население5.itm*).

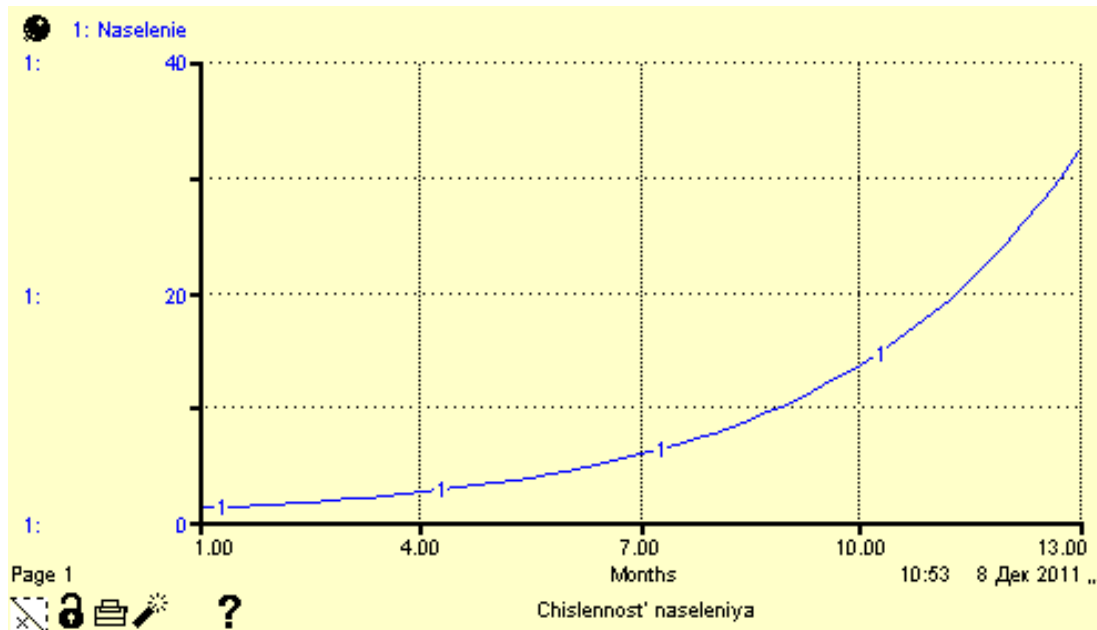


Рисунок 89. Результат работы положительной обратной связи

Уравнения системы, составленные iThink можно увидеть во вкладке Equation (см. Рисунок 90).

```

 Naselenie(t) = Naselenie(t - dt) + (rozhdmost') * dt
INIT Naselenie = 1
INFLOWS:
   rozhdmost' = koeficient_rozhdmosti*Naselenie
 koeficient_rozhdmosti = 0.3
  
```

Рисунок 90. Уравнения системы

В приведенной выше модели (см. Рисунок 87, Рисунок 88) нет средств, ограничивающих рост населения. Добавим в нее фрагмент, отражающий смертность населения — поток *smertnost'*. Смертность зависит от коэффициента смертности (*koeficient smertnosti*) и от численности населения (*Naselenie*).

Пусть значения элементов равны:  $koeficient\ smertnosti = 0.1 * Naselenie$ ,  $smertnost' = koeficient\ smertnosti * Naselenie$ . Соответствующая модель и график изменения численности населения показаны ниже (см. Рисунок 91, Рисунок 92).

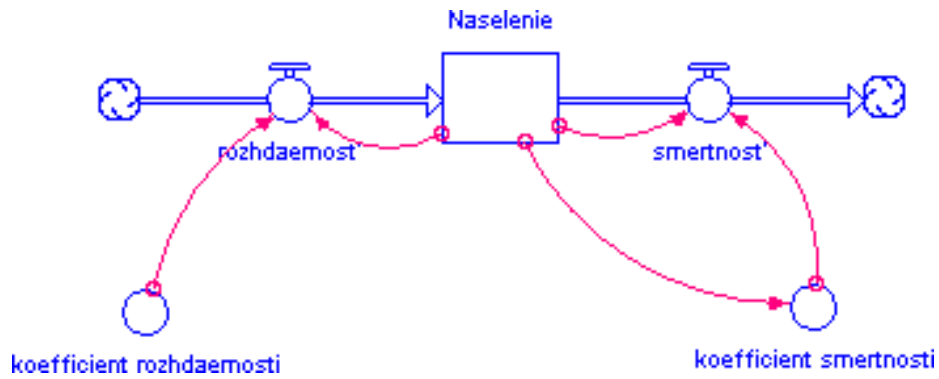


Рисунок 91. Модель, учитывающая смертность

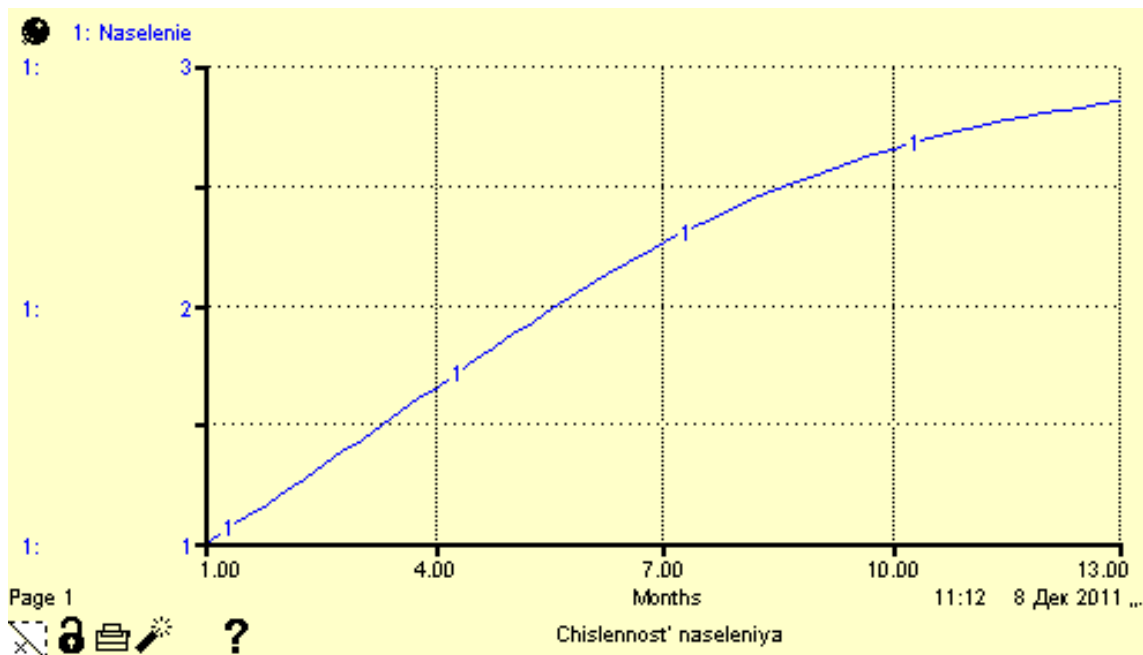
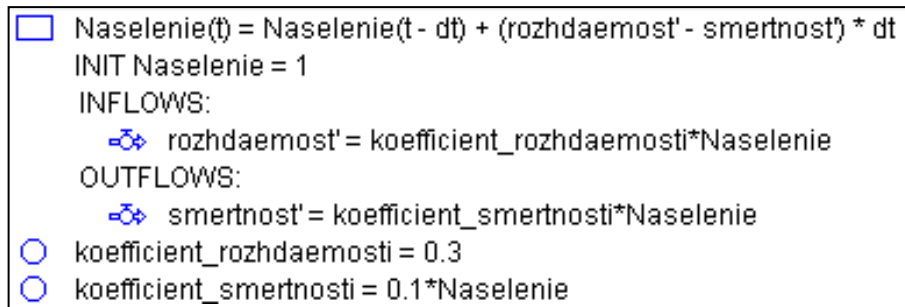


Рисунок 92. График модели, учитывающей смертность

Таким образом, в системе появляется отрицательная обратная связь (больше население -> больше смертей -> быстрее убыль населения), отвечающая за стабилизацию поведения системы, в данном случае — за стабилизацию численности населения.

Заметим, что коэффициент смертности, в свою очередь, зависит от численности населения: если плотность населения становится слишком велика, смертность увеличивается.

iThink в этом случае «конструирует» систему уравнений, приведённую ниже (см. Рисунок 93).



**Рисунок 93. Система уравнения модели с учётом смертности**

### *Литература*

1. Кузнецов Ю. А., Перова В. И Применение пакетов имитационного моделирования для анализа математических моделей экономических систем. — Нижний Новгород: ННГУ, 2007. — 98 с.
2. Сидоренко В.Н. Системная динамика. - М.: ТЕИС, 1998. – 205 с.
3. Емельянов А.А. Имитационное моделирование в экономических информационных системах./ А.А. Емельянов, Е.А. Власова - М.: Изд-во МЭСИ, 1998.-108 с.