**Игорь Цисарь. Моделирование экономики в iThink\_STELLA**

В книге Игоря Францевича Цисаря представлена технология компьютерного моделирования в программе iThink. Название не вполне отвечает содержанию, так как основное внимание уделено самой программе iThink. Моделирование способствует пониманию причинно-следственных связей в экономике, прогнозированию, планированию, принятию решений менеджерами. Тематика примеров позволяет освоить идеи и технологию моделирования экономической динамики. Интересны примеры проектирования оптимальных ставок налогообложения, анализа кризисов и циклов, инфляции, регулирования надежности банков.

И.Ф. Цисарь. Моделирование экономики в iThink\_STELLA. Кризисы, налоги, инфляция, банки. – М.: Диалог-МИФИ, 2009. – 224 с.



Купить книгу в [Ozon](http://www.ozon.ru/context/detail/id/4625014/?partner=baguzin)

Впервые с программой iThink для моделирования системных процессов я познакомился в книге [Денниса Шервуда. Видеть лес за деревьями. Системный подход для совершенствования бизнес-модели](http://baguzin.ru/wp/?p=3899) (недавно Альпина Паблишер выпустило второе издание зачем-то под новым названием [Деннис Шервуд. Системное мышление для руководителей: Практика решения бизнес-проблем](http://baguzin.ru/wp/?p=13718)). Я установил [демоверсию программы](http://www.iseesystems.com/community/downloads/ithink/ithinkDemo.aspx) (если вы хотите не просто прочитать заметку, а выполнить предлагаемые задания, предлагаю вам также установить iThink 10.1, доступную на сайте производителя), нашел в Интернете [инструкцию](http://www.unn.ru/pages/e-library/aids/2007/55.pdf) и более детально разобрал одну из глав книги Шервуда в заметке [Моделирование системной динамики в iThink](http://baguzin.ru/wp/?p=3968) (рекомендую начать с этой небольшой статьи).

Примеры, используемые в книге можно скачать на [сайте издательства](http://dialog-mifi.ru/mifi/?page_id=165).

Моделирование используется для анализа, прогнозирования и планирования практически всей иерархии экономических систем и объектов от мировой экономико-экологической системы (модели Римского клуба) до отдельных рабочих мест. Аналитические департаменты Международного валютного фонда моделируют финансовое состояние стран. Они разрабатывают стабилизационные антикризисные программы. Многие попавшие в кризис страны считают эти программы вредными. ☺

В зависимости от задачи экономисты выбирают программную систему. Они выбирают то, что знают, Excel. Но Excel недостаточен для задач экономической динамики и в нем плохо отображаются причинно-следственные связи. Лучше использовать iThink или другие программы для моделирования экономической динамики.

**Уровни модели**

Модели в iThink представляются пятью иерархическими уровнями (рис. 1; закладки слева), или уровнями детализации модели: Interface – уровень схем; Map – уровень блок-схем; Model – уровень модели; Equation – уровень формул (на языке программирования Dynamo), программа формируется автоматически на основе уровня Model; SubModel – уровень разбиения модели на подсистемы с целью избавиться от чрезмерной детализации.

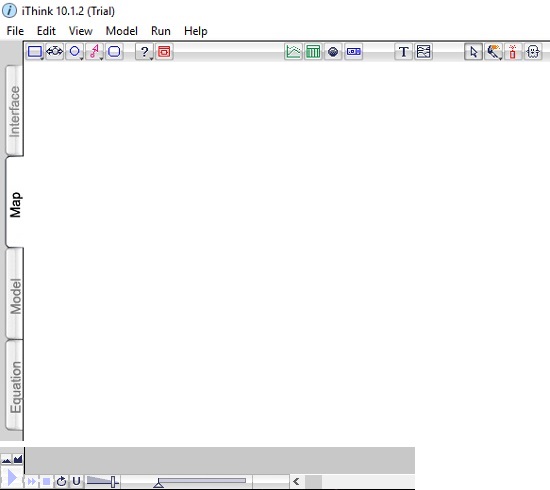


Рис. 1. Начальное окно iThink v.10.1.2 (Trial)

Под строкой меню расположены управляющие элементы – иконки вызова стандартных блоков, в которые уже встроен функционал (формулы). Набор кнопок отличается для различных уровней детализации модели. В левой нижней части окна верхние две кнопки управляют масштабом представления модели. Угловая и следующие кнопки дублирует меню Run, т.е. запуск модели на исполнение. Правый значок, похожий на потенциометр, отображает время выполнения программы. Значком левее его вы можете регулировать скорость выполнения программы. Значок U восстанавливает исходное состояние блоков модели. Кнопка круга со стрелкой используется при анализе циклов.

### Модель сберегательного счета

Начнем с простейшей модели. Перейдем на закладку Map. Введем заголовок. Для этого щелкнем по кнопке текстового блока Т. Поместим курсор вверху по центру окна модели и щелкнем. Наберем текст, как в обычных редакторах: «Технология разработки модели сберегательного счета» (рис. 2). Банковский счет – это хранилище наших денег. Щелкнем по левому прямоугольнику на панели инструментов – это Stock (фонд, запас, хранилище). Поместим курсор в окно модели и щелкнем. Появится блок с именем Stock1. Щелкнем по второй иконке с краником. Поместим курсор в модель и протянем до хранилища. Это поток денег на наш счет. Щелкнем на панели кружок конвертера (переменной), третья слева. Поместим ее ниже потока. Щелкнем на четвертой слева стрелке коннектора и протащим курсор от хранилища до переменной. Появилась линия связи со стрелкой. Добавим слева еще кружок переменной и соединим стрелкой с правой переменной. Блок-схема модели готова. Дадим блокам содержательные имена. Для этого щелкнем по блоку (выделим его). Переведем курсор в имя и заменим его текст. Account – счет, Rate – процентная ставка, Interest – начисленный процентный доход, InterestF – поток процентного дохода, капитализируемого на счете.

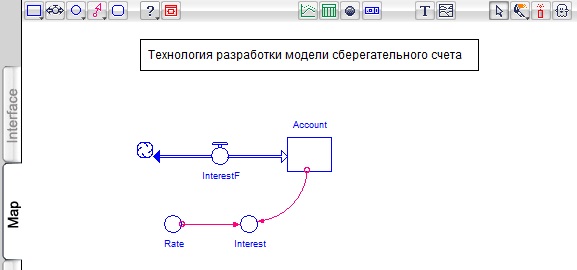


Рис. 2. Окно *Map* модели сберегательного счета

Теперь в режиме *Model* надо в каждый блок ввести формулы и параметры. Двойным щелчком открываем диалоговое окно каждого блока. В нижнюю часть диалогового окна *Account* введем начальный вклад в 100 руб. (рис. 3).

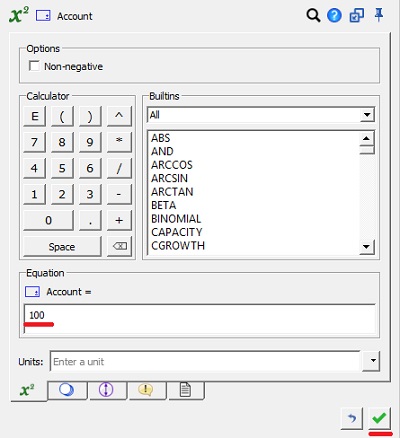


Рис. 3. Диалоговое окно *Account*

В нижнее поле *Rate* введем 10%-ную ставку в десятичном формате 0.1 (именно точка является разделителем; *iThink* не русифицирован). В нижнее поле переменной *Interest* вставляем формулу произведения суммы счета на ставку: *Account\*Rate* (рис. 4). Для ввода формулы нажмите последовательно *Account* в поле *Required Inputs*,знак\*, *Rate*. Проведем стрелку от *Interest* к *InterestF.* В окне финансового потока *InterestF* в нижнее поле вставим переменную *Interest*.

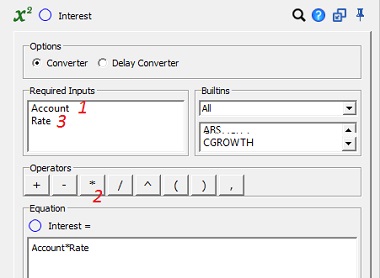


Рис. 4. Диалоговое окно переменной Interest

Чтобы увидеть результаты моделирования, добавьте в модель таблицу и график. Щелкните по их иконкам, а затем по их месту в окне модели. Их иконки появятся в окне модели. Работы с таблицей и графиком во многом похожи. Надо внести в них имена интересующих нас переменных. Дважды щелкните по таблице в окне модели. Раскроется отдельное окно таблицы. Дважды щелкните в окне таблицы. Появится окно настройки таблицы (рис. 5).

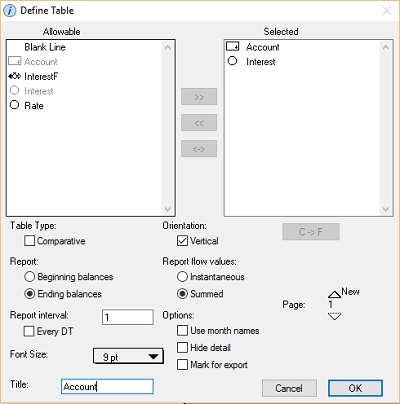


Рис. 5. Окно настройки таблицы

В левом поле представлены имена переменных, которые можно включить в таблицу (Allowable). В правом поле (Selected) имена выбранных для отображения переменных. Для выбора переменной из левого списка щелкните по ее имени. Затем щелкните по кнопке >>. Имя переменной появится в правом поле. Слева внизу введите имя таблице. Кликните *Оk*. В заголовках столбцов появляются имена переменных. Аналогично поступаем с графиком. Дважды кликаем по нему в окне модели и выбираем Account и Interest, в качестве переменных оси Y (ось Х отражает время). Ваша модель готова. Запустите ее кнопкой Run. Результаты моделирования показаны на рис. 6.

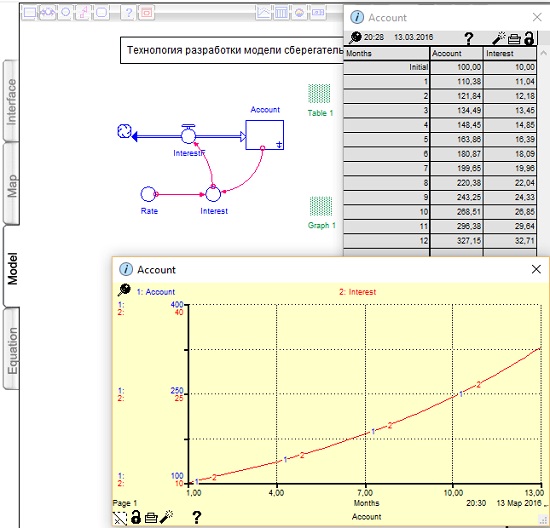


Рис. 6. Результаты моделирования сберегательного счета

### Налоговая модель

Бюджетно-оптимальной называем налоговую ставку на прибыль или общую налоговую нагрузку на бизнес, при которой поступления в бюджет максимальны. Вы нигде не встретите обоснования величин налоговых ставок. Думы, кнессеты, парламенты, конгрессы, бундестаги и проч. берут их «с потолка» и регулируют методом проб и ошибок. Исследуем зависимость оптимальной ставки налога от горизонта планирования. На этот раз начнем с уровня интерфейса (рис. 7). Две подмодели (Process Frame) взаимодействуют между собой через потоковые (жирные линии, Bundled Flow) и коннекторные (тонкие линии, Bundled Connector) связи. Схема отражает взаимодействие финансовых потоков, фондов (запасов) и информационных переменных.

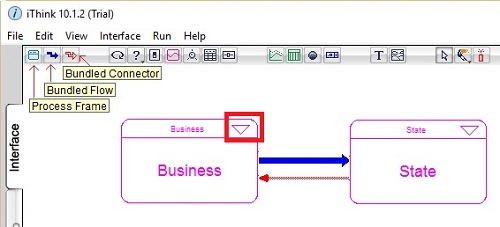


Рис. 7. Модель для определения оптимальной ставки налога на прибыль предприятия, уровень интерфейса

При проектировании «сверху-вниз», каждая подмодель может быть «раскрыта» детальной блок-схемой из стандартных блоков (рис. 8). Надо отметить, что Trial-версия не содержала перевернутых треугольников внутри блоков, так что попасть с уровня интерфейса на уровень блок-схемы мне не удалось. Однако, если открыть файл Taxrate\_Arr\_Tplan.STM, то всё получится (напоминаю, что примеры можно скачать на [сайте издательства](http://dialog-mifi.ru/mifi/?page_id=165)). Если перейти на уровень формул, то можно увидеть уравнения и начальные параметры (рис. 9).

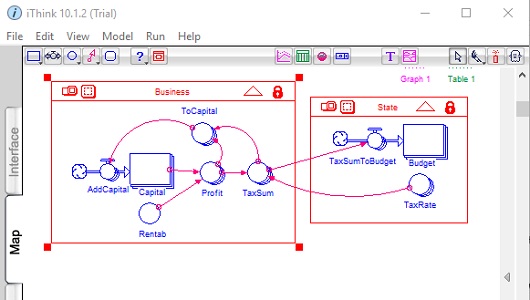


Рис. 8. Модель для определения оптимальной ставки налога на прибыль предприятия, уровень блок-схемы

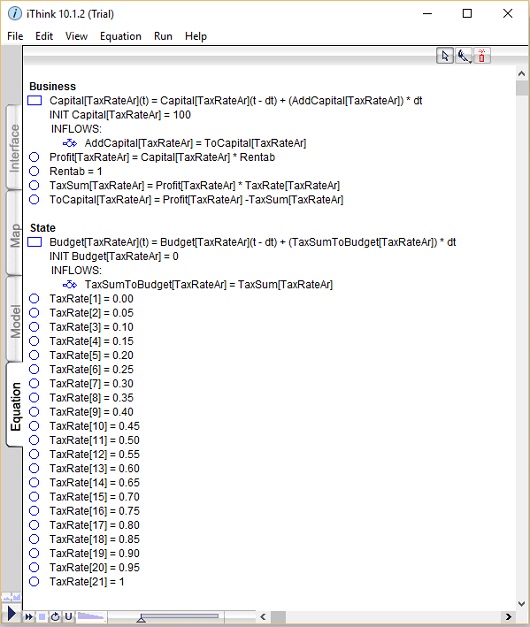


Рис. 9. Модель для определения оптимальной ставки налога на прибыль предприятия, уровень формул

Государство задает налоговые ставки блоком TaxRate. Как информация стрелкой они отправляются в подсистему бизнеса. Из бизнеса стрелкой налоговые отчисления поступают в бюджет. Поток налоговых сумм отражается блоком TaxSumToBudget. Они стрелкой отправляются на бюджетный счет. Это квадратный блок Budget. При выполнении исследовательских экспериментов мы будем задавать различные налоговые ставки и замерять накопление средств на бюджетном счете за какой-то интервал времени. Так мы будем исследовать эффективность налоговой системы с позиций государства.

В подсистеме Business квадратный блок моделирует капитал бизнеса. Слева на него поступает поток прибыли, остающейся в распоряжении предприятия (AddCapital). Тем самым обеспечивается рост капитала фирмы. Нижним блоком Rentab задается рентабельность фирмы. В блоке Profit вычисляется прибыль как произведение капитала на рентабельность. В блоке TaxSum вычисляется сумма налоговых платежей как произведение прибыли на налоговую ставку. Эти суммы поступают в бюджет. В блоке ToCapital вычисляется разница между доналоговой прибылью и налоговыми отчислениями. Эта прибыль, остающаяся в распоряжении предприятия, полностью поступает для наращивания капитала.

В качестве исходных данных задаются числовые значения: налоговой ставки, рентабельности, начального капитала фирм. Задается временной интервал моделирования (20 лет). Отображение информации экспериментов выполняется в блоках Graph и Table, расположенных в правом верхнем углу экрана (см. рис. 8). Поскольку в iThink нельзя построить зависимость оптимальной ставки налога (максимизирующей бюджетные поступления) от горизонта планирования, данные экспортируются в Excel (рис. 10). Подкрасьте максимальные значения в каждой строке с помощью условного форматирования. Представьте зависимость бюджетно-оптимальной ставки от периода планирования в графическом виде (рис. 11).

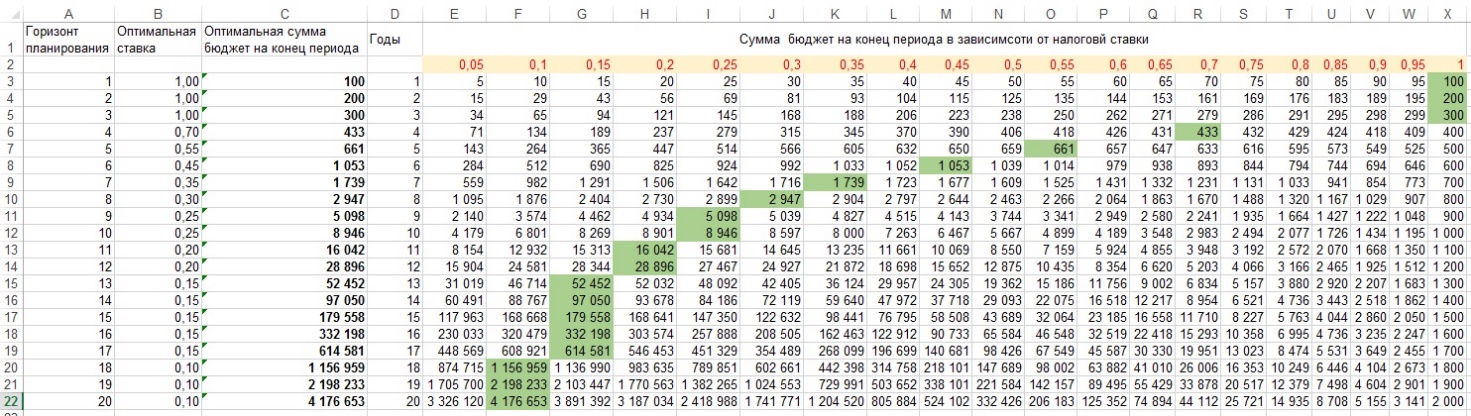


Рис. 10. Результаты моделирования для ставок налога от 5% до 100%



Рис. 11. Зависимость бюджетно-оптимальной ставки от горизонта планирования

Результаты моделирования показывают, что с увеличением горизонта планирования оптимальная ставка уменьшается. Поэтому целесообразно уменьшать налоговые ставки, поощрять развитие бизнеса. Решение о фиксации горизонта ставок является субъективным. Оно зависит от того сколько лет мы собираемся прожить в нашей экономике.

### Строительные блоки

На уровне блок-схемы (Model Layer) iThink предоставляет: хранилище (Stock), потоки (Flows), конвертеры (Converters), коннекторы (Connectors).

**Хранилища.** По умолчанию все хранилища имеют тип Резервуар (Reservoir). Тип хранилища можно сменить, кликнув на нем правой кнопкой мыши и выбрав опцию Change to (рис. 12).

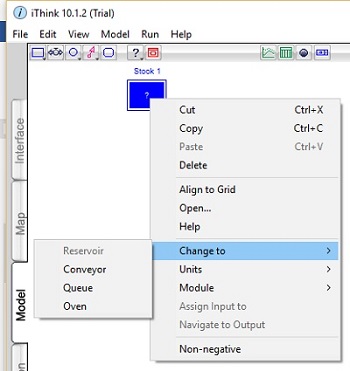


Рис. 12. Смена типа резервуара

В поле Equation можно задать начальные значения запаса (рис. 13) – константу или алгебраическое выражение. Калькулятор используется для ввода цифр и символов операций в область для задания уравнения. Нажатие на клавишу Alt предоставляет доступ к дополнительным кнопкам калькулятора. Комментарием считается все, что стоит внутри фигурных скобок {}.

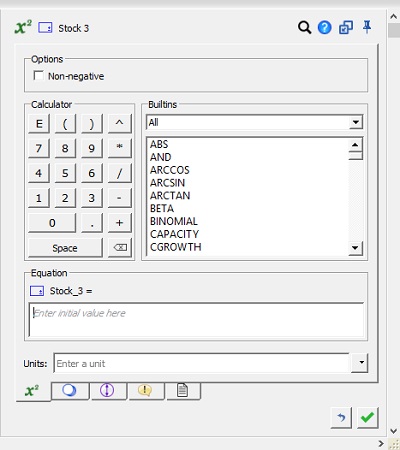


Рис. 13. Диалоговое окно параметров резервуара

В нижней части окна можно выбрать единицу измерения запаса (рис. 14). Например, деньги.

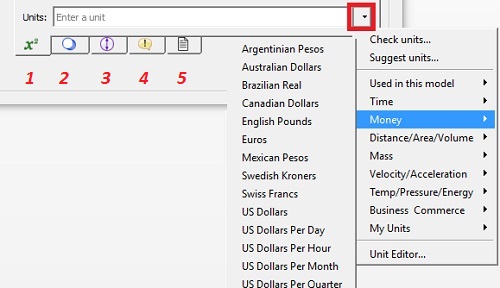


Рис. 14. Выбор единицы измерения запаса

Под полем Units имеются кнопки, позволяющие открыть дополнительные закладки. 1 – Equation мы только что разобрали. 2 – Array (массив), 3 – Scales and Ranges, 4 – Message (сообщение) позволяет ввести сообщение, сигнализирующее во время выполнения о том, что произошло определенное событие. 5 – Documentation (документация) позволяет задать дополнительную информацию об объекте, например, описать его назначение.

*Конвейер (Conveyor)* можно рассматривать как движущийся эскалатор или конвейерную ленту. Вещество попадает на конвейер, едет некоторый период времени, и затем сбрасывается. Время передачи (Transit time) конвейера может быть постоянным или переменным. Операции конвейера могут быть приостановлены. В конвейер может входить несколько однонаправленных потоков. Поступление вещества на конвейер может ограничиваться как емкостью (Capacity), так и лимитом входного потока (Inflow Limit). Возможна утечка содержимого конвейера. В зависимости от природы входящих потоков окно Conveyor может принимать различные формы.

*Очередь (Queue)* рассматривается как выстроенный в линию набор элементов, ожидающих вхождения в какой-то процесс. Очереди функционируют по принципу FIFO (first in – first out, первый пришел – первый ушел). Вещество попадает в конец очереди, движется по ней и выходит из нее. Допустимо применение нескольких входных потоков, все они должны быть Uniflow (однонаправленными). Если на входе очереди задано несколько потоков, то для этих потоков должны быть заданы приоритеты, определяющие последовательность их доступа потоков к очереди.

*Печь (Oven)* – процесс с дискретными порциями вещества. Печь открывается, наполняется (в соответствии с мощностью или до тех пор, пока не наступит время закрыть дверцу), «выпекает» свое содержимое некоторое время (определяемое логикой выходного потока), затем единовременно освобождается. Операции печи могут быть остановлены. В печь может входить только один однонаправленный поток. Входной поток Печи может исходить от облака, Резервуара или Очереди.

*Создание сообщений (Messages).* Сообщения позволяют напомнить и объяснить пользователям особенности поведения модели. Они могут включать текст, картинки, видеоролики и звуки. Вы можете создать сообщения, порождаемые потоками, хранилищами и конвертерами модели.

**Поток (Flow).** Потоки пополняют содержимое хранилищ и наоборот, уменьшают его. Только по потокам физические и финансовые единицы могут попасть в хранилище. Поток снабжен стрелкой, показывающей, откуда и куда перетекает содержимое хранилищ. Чтобы создать поток (рис. 15):

* Выберите Flow, щелкнув на соответствующей иконке в палитре Строительных блоков.
* Поднесите курсор к месту, в котором должен начаться поток. Это не обязательно должно быть хранилище. Могут существовать потоки, выходящие как бы извне диаграммы (из так называемого «облака»).
* Щелкните мышью и, не отпуская кнопку, растяните поток до хранилища, в которое он должен «втекать», и отпустите кнопку мыши. Аналогично поток не обязан «втекать» в хранилище. Он может уходить за пределы диаграммы (в «облако»).

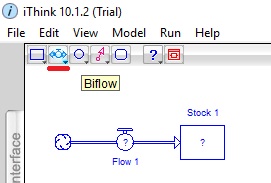


Рис. 15. Создание потока

Замечания по размещению потоков:

1. Поток может быть Uniflow (однонаправленным) или Biflow (двунаправленным). Чтобы переключиться между ними, кликните на потоке правой кнопкой мыши и выберите Change to (изменить на).
2. Чтобы «согнуть» стрелку, представляющую поток, переместите один из ее концов выше или ниже горизонтальной линии.
3. Чтобы стрелка связывала хранилища, обязательно надо попасть внутрь границы. В противном случае стрелка будет выходить из «облака». Аналогично вы должны быть уверены, что закончили стрелку внутри границы конечного хранилища, поскольку в противном случае поток закончится «облаком».
4. Чтобы заменить облако на хранилище, выберите иконку хранилища из палитры строительных блоков. Растяните над облаком. В тот момент, когда курсор будет располагаться непосредственно над облаком, последнее изменит цвет. Убедитесь, что вы захватили стрелку, представляющую поток, и отпустите кнопку мыши. В результате облако исчезнет, а его место займет только что созданное хранилище.

**Конвертер (Converter).** Конвертеры (переменные) служат для регулирования потоков (на схемах они изображаются окружностями). Они задаются алгебраическими выражениями с использованием встроенных функций. Их значения пересчитываются в каждый такт модельного времени. Чтобы создать конвертер щелкните по его иконке на панели инструментов. Поднесите курсор к месту, в котором хотите поместить его на диаграмме, и щелкните кнопкой мыши еще раз. Если в режиме Model дважды щелкнуть внутри конвертера, откроется окно установок параметров.

**Коннекторы** связывают элементы модели, и изображаются тонкими стрелками. Они не могут подсоединяться стрелкой к хранилищу (рис. 16), но могут выходить из него. Единственная возможность изменить величину накопленного в хранилище вещества – использование потоков. При создании коннектора вы можете столкнуться с сообщением о том, что подобная связь недопустима. Это означает, что вы пытаетесь создать замкнутый цикл. Чтобы разрешить эту проблему, добавьте в цепочку хранилище. Коннектор не имеет диалога, позволяющего конфигурировать его параметры, поскольку ни от каких параметров он не зависит.

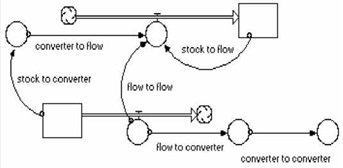


Рис. 16. Примеры применения коннекторов

### Объекты

С помощью объектов комментируются, преобразуются и отображаются входные данные и результаты имитационных экспериментов. На разных уровнях детализации модели набор объектов немного различается (рис. 17).

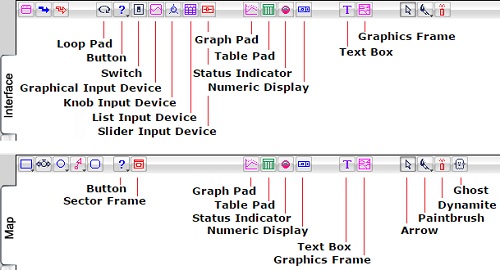


Рис. 17. Иконки, кнопки объектов: наверху – вкладка Interface, внизу – Map/Model

*Цикл (Loop Pad).* Для некоторых моделей диаграмма циклов позволяет лучше понять структуру модели, причины ее поведения, изучить все существующие в модели циклы обратной связи (см., например, [Питер Сенге. Пятая дисциплина. Искусство и практика обучающейся организации](http://baguzin.ru/wp/?p=1200)). Для создания цикла выберите иконку Loop Pad, щелкните кнопкой мыши в произвольном месте рабочей области (рис. 18). Если Loop Pad содержит более одной страницы, становится доступным аппарат для «переворачивания» страниц, расположенный в нижнем левом углу окна.

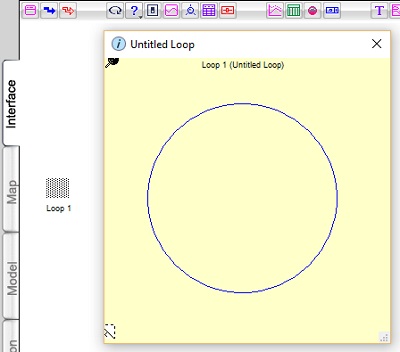


Рис. 18. Окно цикла

Чтобы открыть окно параметров цикла дважды кликните на нем (рис. 19). Кнопками добавьте хранилища в цикл. Список Variable (Loop Number) [Переменная (Номер цикла)] отображает все циклы модели, проходящие через отобранные хранилища. Если через выбранные хранилища не проходит ни один цикл, этот список окажется пустым. Щелкните название того цикла, который вы хотите построить, чтобы имена составляющих его объектов появились в списке Entities in Selected Loop. Для указания вида цикла можно выбрать из: None (никакой), R-reinforcing (усилительный) или С-counteracting (ослабляющий). Название страницы можно указать в поле Title. Чтобы добавить страницу нажмите на стрелку вверх [рядом с надписью New (новая)] и новая страница будет добавлена.

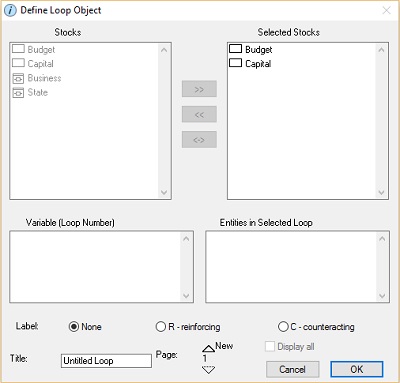


Рис. 19. Диалог Loop Pad

*Кнопки (Button)* облегчают перемещение между уровнями без необходимости использования стрелок навигации и полос прокрутки. Они позволяют работать с моделью, не используя выпадающие меню или палитру инструментов. Кнопки также могут использоваться для обеспечения всплывающей информацией конечных пользователей. Кнопки бывают 5 типов: информационная, навигационная, дублирующая меню, рассмотрения цикла/логики, воспроизведение видеороликов, звуковые.

*Переключатель (Switch)* изменяет порядок работы элемента, с которым он связан, на поведение on/off (включен/выключен). Если связать переключатель с конвертером, то он будет возвращать 1, если переключатель включен, и 0, если выключен. Если связать переключатель с сектором, то при моделировании этот сектор будет активным, если переключатель включен. Переключатели удобны при рассмотрении моделей, которые могут развиваться по нескольким различным сценариям. Переключатели могут выглядеть как обычная кнопка (Push Button) или «рубильник» (Toggle, рис. 20). Если надо «сцепить» более одной переменной, чтобы только одна из них могла быть «включена» в каждый момент времени, то выберите вариант Push, и отметьте «галочкой» поле Chaining.

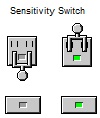


Рис. 20. Примеры переключателей различных видов

*Графический элемент ввода (Graphical Input Device)* используется для ввода случайной переменной, описываемой некой функцией (например, колоколообразной). Если графический элемент ввода не нужен, удалите его с помощью Динамита (см. рис. 17).

*Ручка управления (Knob)* полезна для задания начальных значений хранилищ, регулирования постоянных значений конвертеров и потоков. После того, как вы связали хранилище с Ручкой управления, последняя определяет начальное состояние этого хранилища. Если переместить точку, представляющую значение связанной переменной, в новую позицию, то на Ручке управления появится кнопка с буквой U. Чтобы восстановить прежнее значение, нажмите на эту кнопку.

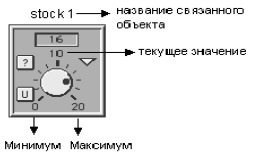


Рис. 21. Ручка управления

*Списковый элемент ввода (List Input Device, LID)* совмещает несколько полей ввода и позволяет задавать входные данные для одного или более элементов модели в одном окне, что во многих случаях очень удобно. Списковый элемент ввода может применяться только для задания начальных значений хранилищ. Нельзя изменять значение во время моделирования.

*«Сцепленные» ползунки (Chained Slider).* Работа с этим объектом базируется на стандартных принципах работы с ползунками в Windows. С ползунком вы можете связать только конвертеры и потоки. «Сцепленные» ползунки полезны в тех случаях, когда вы имеете дело с взаимосвязанными величинами. Например, если допустимо менять каждое из значений, но общая сумма не должна превосходить определенного значения (рис. 22).

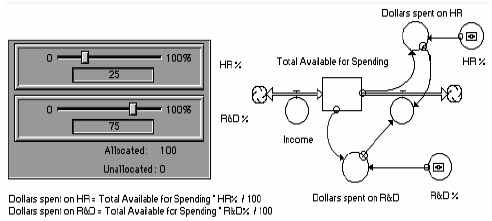


Рис. 22. Распределение общего фонда между двумя отделами

*Объект График (Graph Pad)* используется для отображения результатов моделирования. iThink поддерживает 4 вида графиков:

* Time series; зависимость X, Y, Z от времени;
* Scatter plots; зависимость X от Y;
* Bar graphs; столбчатые диаграммы;
* Sketchable; график с эскизом.

Вы можете вывести только график для последнего акта моделирования, а при выборе опции Comparative, можно сравнить результаты нескольких последовательных «прогонов». Это позволяет проводить анализ чувствительности.

Sketchable Graph позволяет задать эскиз ожидаемого поведения связанной с графиком переменной, чтобы сравнить предполагаемое поведение с фактическим. Он также выполняет функцию ввода данных. Остальные типы графиков могут являться лишь устройствами вывода.

*Таблица* *(Table Pad*) служит представлению данных, а также передаче данных другим приложениям (например, Excel)

*Устройство предупреждения (Warning Device)* комбинируют измеритель, числовые значения, мигание и звуковое предупреждение для подачи сигналов о поведении переменной модели во время выполнения. Эти устройства доступны только на уровне фреймов (Interface).

*Числовой дисплей (Numeric Display)* используется для отображения текущего состояния связанной с ним переменной.

*Текстовый блок (Text Block)* служит для ввода текста в окне модели.

*Графический фрейм (Graphics Frame)* позволяют дополнять модель графикой и видеороликами. Они доступны только на уровне фреймов.

*Фрейм сектора (Sector Frame)* на уровне потоковых диаграмм (Map) может использоваться для достижения следующих целей: с его помощью вы можете объединять в одно логическое целое взаимосвязанные элементы модели; он может использоваться для визуализации изображений и воспроизведения видеороликов.

**Инструменты.** К инструментам относятся (см. правая часть рис. 17): стрелка (Arrow), кисть (Paintbrush), динамит (Dynamite), фантом (Ghost, приведение, призрак). Инструменты позволят выполнять основные действия над объектами модели, например, выделять, перемещать, удалять, раскрашивать и определять значения соответствующих параметров.

Фантом является эффективным инструментом при конструировании сложных потоковых схем. В таких схемах коннекторы (линии) настолько перепутываются, что невозможно отследить взаимодействия. С помощью Фантомов можно создавать копии, «призраки» объектов, а затем размещать их рядом с удаленными объектами и тем самым упрощать исходные схемы, делая их наглядными и структурированными.

**Встроенные функции.** Функции IThink объединены в группы:

* Тестовые входные значения.
* Математические функции.
* Тригонометические функции.
* Логические функции.
* Статистические функции.
* Экономические функции.
* Дискретные функции.
* Функции циклического выполнения.
* Встроенные функции для работы с массивами.
* Функции специального назначения.

Например тестовая функция TREND(<input><averaging time>[,<initial>]) может использоваться трейдерами при анализе движения цен на фондовом рынке. Параметр Input задает предполагаемое текущее значение цены, которая движется взлетами и падениями. Время действия тренда задается значением Averaging time и определяет, краткосрочный или долгосрочный тренд. По результатам имитационных экспериментов можно увидеть, какой тренд определен на нужный вам срок («бычий», «медвежий» или «боковой»), и в зависимости от этого какой стратегии вам следует придерживаться (рис. 23).

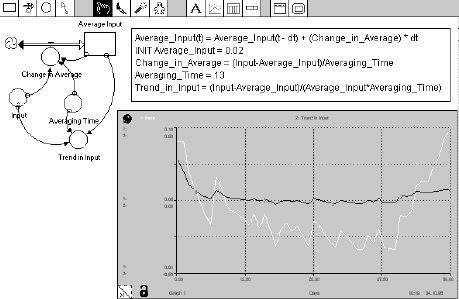


Рис. 23. Анализ направления движения цены

*Экономические функции.* В других языках их называют финансовыми. Например, текущее значение (Present value, pv), Будущее значение (future value, fv), периодические выплаты (periodic payment, pmt), число периодов (number of periods, nper) и процентная ставка за период (interest rate per period, rate). Функции FV, PMT и PV вычисляют будущее значение, периодические выплаты и текущее значение, основываясь на значениях других финансовых параметров. Наиболее часто используется NPV(<input>,<rate>[,<initial>]) – чистое текущее значение (net present value, рис. 24).

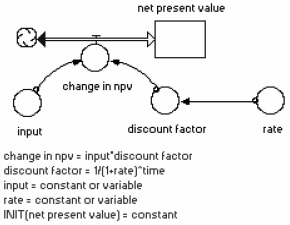


Рис. 24. Структура функция NPV

*Дискретные функции* предоставляют информацию о внутренних характеристиках очередей, конвейеров или печей (CAP, COOKTIME, OSTATE, QELEM, QLEN, TRANSTIME). Функция DELAY позволяет задерживать входной сигнал на заданное время.

*Функции циклического выполнения.* Например, функция CTFLOW(<flow>[<initial>) возвращает значение, равное количеству вещества, проходящего по потоку, для которого производятся замеры. На рис. 25 конвертер time stamped flow volume вычисляет количество вещества, прошедшее по потоку exit process 2.

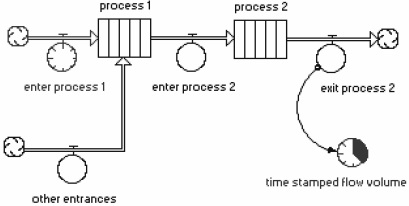


Рис. 25. Применение функции CTFLOW

**Массивы.** Для выполнения исследовательских экспериментов при различных значениях параметров и экзогенных (внешних) переменных удобно представлять их массивами. Пример дан в налоговой модели. Мы задаем массив налоговых ставок и получаем массив налоговых поступлений в бюджет. Процесс добавления массивов в модель состоит из 3 шагов: создание одной или более Величин (Dimension), используя Редактор массивов; создание одной или более переменных типа «массив»; создание уравнений. Чтобы добавить Величину (Dimension), пройдите по меню *Model* –> *Array Edit* или *Interface* –> *Array Edit* (рис. 26). Кнопкой + добавьте новый массив, определите его тип (Number/Label) и размер (Size, число элементов, рис. 27).

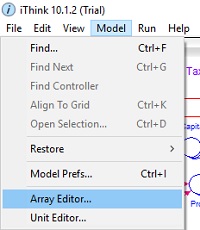


Рис. 26. Запуск редактора массивов

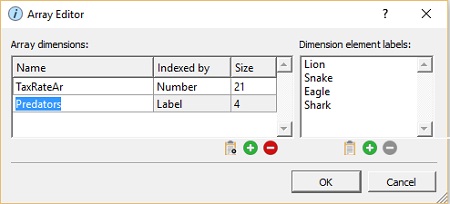


Рис. 27. Задание переменной типа массив

**Базовые потоковые процессы.** Динамика процессов и систем, как бы они сложны ни были, выражается через функционирование всего пяти типов базовых конструкций:

* прирост фондов;
* уменьшение фондов;
* рабочий процесс;
* совмещение потоков;
* адаптация фондов.

Каждая из этих конструкций входит практически в любые инфраструктуры современного менеджмента.

Процесс *приращения фондов* используется в ситуациях, когда необходимо организовать регулирование роста фонда интенсивностью его приращения. Приращение фондов определяется формулой: входной поток = фонд \* интенсивность приращения (рис. 28).

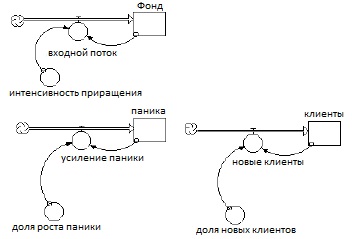


Рис. 28. Потоковые процессы приращения фондов

**Моделирование циклов и кризисов.**Экономику стран и мира периодически потрясают кризисы. Например, перед нашим дефолтом 1998 г. Китай либерализовал внешнюю торговлю. Поток товаров хлынул в страны Юго-Восточной Азии, к «тиграм». У тигров упали объемы продаж, прибыли. Они не смогли вернуть банкам кредиты. Стали банкротиться банки. Рушилась банковская система. Международный валютный фонд со своим несовершенным мониторингом предкризисных состояний вынужден был дать для выхода из кризиса по $40 млрд. Южной Корее, Индонезии, Таиланду и др. России дали лишь 6 миллиардов, но они, как слишком ликвидные, где-то испарились.

Причины кризисов остаются не раскрытыми. Мы исследуем причинно-следственный механизм возникновения циклов и кризисов перепроизводства с использованием компьютерного моделирования. Наша модель будет сильно упрощенной (рис. 29, файл Cris\_Stla2conv.STM). Любая модель – это абстракция, огрубление реальности. Чтобы уменьшить модель, мы не включаем в нее много факторов. При дальнейшем развитии модели вы можете пополнить ее временно отложенными факторами.

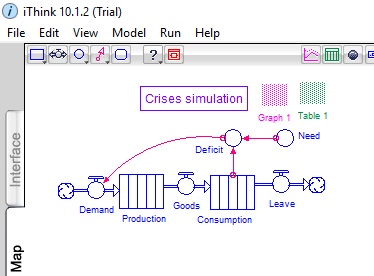


Рис. 29. Модель для анализа циклов и кризисов перепроизводства

Последние 80 лет кризисы перепроизводства, как правило, начинались в автомобильной промышленности и через 5–10 лет после окончания войн. Поэтому предположим, что промышленность выпускает оборудование в виде автомобилей. В модели 2 подсистемы: Производство (Production) и Потребление (Consumption). Жирные линии с вентилями задают финансовые или материальные регулируемые потоки. Need – потребность пользователей в парке оборудования (продукции). Deficit вычисляется, как разность между потребностью в оборудовании и наличием его в эксплуатации. Demand – поток спроса или заказов на производство продукции. Переменная Goods – это поток товаров из производства к потребителям. Leave – поток выбывающих из эксплуатации по ветхости и износу изделий. Облака нейтральны и отражают лишь возможность связи с внешней средой.

Блок производства с его системой управления задаем в виде грубейшей модели. Производство выполняет заказ полностью, но с фиксированным сроком исполнения, задаваемым блоком временной задержки (конвейер, лаг исполнения заказа). Например, Владимирский тракторный завод принимает заказ с предоплатой и через полгода выдает вам трактора. Минский тракторный завод выполняет заказ за 3 месяца и поэтому выигрывает в конкуренции. Химчистка на вашей улице, надеюсь, выполняет заказ за один день. Блок потребления, эксплуатации изделий, задан стандартным блоком конвейер. Он отражает поступление изделий к покупателям в эксплуатацию. По истечении срока службы изделия выбывают из потребления.

После ввода в диалоговые окна блоков числовых параметров приступаем к имитации. Щелкаем Run и через доли секунды получаем графики показателей развития процесса во времени (рис. 30). На горизонтальной оси отображается время в годах. На вертикальной оси отображаются значения соответствующих показателей.

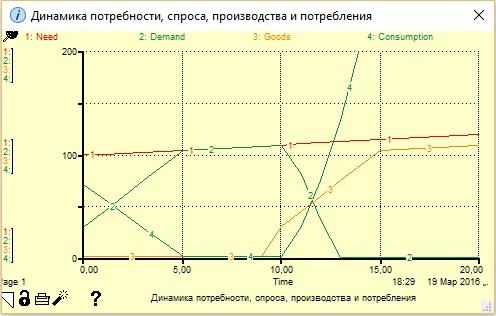


Рис. 30. Динамика потребности (1), спроса (2), производства (3) и потребления (4)

Парадокс! Мы задали единственную внешнюю (экзогенную) переменную: потребность в изделиях. Это медленно растущая прямая линия (1). Но мы получили колебания, циклы и кризисы по всем остальным показателям: спросу, производству, наличию товаров в эксплуатации. В чем причина? Объяснение дается общей теорией систем управления. На рис. 29 мы видим контур (цикл, петлю) отрицательной обратной связи: дефицит, спрос, производство, потребление, дефицит. Увеличение потребности ведет к увеличению дефицита, спроса, производства, потребления. С запаздыванием начинается уменьшение этих показателей вплоть до прекращения производства. В математической теории систем управления доказано, что в системах с отрицательной обратной связью запаздываниями и накоплениями возможны колебания и неустойчивость. В экономике это циклы и кризисы. Таков закон нашей системы.

Основные характеристики циклов – это период и амплитуда колебаний (глубина цикла). С увеличением задержки, т.е. отставания реакции производства на спрос, возрастает амплитуда и период колебаний показателей экономической системы. Снижается устойчивость системы производство-потребление, возрастает неустойчивость, возможность кризисов (см. также [Моделирование работы системы с запаздываниями](http://baguzin.ru/wp/?p=3589)).