1. *Создается новая модель.*

2. *В модель помещается новый класс.* При этом определяется родительский класс и указывается имя нового класса.

3. *В классе определяются новые и (или) перекрываются старые свойства, методы и события.*

4. *Для каждой новой сущности класса программист кодирует только операторы исполняемого кода.*

5. *Новый класс помещается в модуль.*

6. *Выполняется генерация вновь созданной модели, в результате которой получается полный исходный текст модуля.*

7. *С помощью Delphi осуществляется отладка нового модуля*, и, если он содержит новые компоненты, эти компоненты регистрируются в VCL.

8. *С помощью Model Maker исходный текст документируется (снабжается комментариями).* Model Maker поможет также создать необходимый файл помощи.

Использование Model Maker делает возможным создание приложения от этапа проектирования и до готового исходного кода без применения каких либо дополнительных case-инструментов.

Таким образом, case-технология представляет собой совокупность методологий анализа, проектирования, разработки и сопровождения сложных динамических систем, поддерживаемую комплексом взаимосвязанных средств автоматизации.

Работа выполнена при финансовой поддержки Министерства образования и науки РФ (проект № 7.3975.2011 в Алтайском государственном университете).

Математическая модель оптимизации поставок сырья

К.Н. Галанов
АлтГУ, г. Барнаул

В докладе рассматривается решение задачи повышения эффективности управления логистическими процессами промышленных предприятий в условиях рыночных отношений. В качестве инструментария исследования выбраны методы математического и имитационного моделирования.

Исследование выполнено в условиях ОАО ПО «Алтайский шинный комбинат». Объектом исследования выступает процесс снабжения завода каучуками: СКИ-3, СКМС-30 АРКМ-15, СКД-2, SVR-3L, на которые приходится более 50% стоимости сырьевой корзины.

В итоге построена математическая модель позволяющая, оптимизировать процесс контейнерных поставок сырья. Компьютерная реализация модели, проведена в среде MS Excel.

Вычисления с ограничениями и алгоритмические проблемы

В.А. Ганов, В.Р. Карымов
АлтГУ, г. Барнаул

В работе [1] была доказана ω-противоречивость формальной арифметики. В частности, было установлено, что в этой системе доказуемы интуитивно ложные предложения. Поэтому безоговорочное использование методов формальной арифметики при доказательстве чего-либо оставляет сомнение в верности этого доказательства. В первую очередь это касается известных утверждений о нерекурсивности некоторых отношений, связанных с проблемой остановки машин Тьюринга.

Пусть $< z;x >$ обозначает инициальную машину Тьюринга, где z – код машины и x – ее аргументом. Известно, что не существует рекурсивной функции, $h_1(z,x)$, такой, что $h_1(z,x) = 1$, если $< z;x >$ останавливается, и $h_1(z,x) = 0$, если $< z;x >$ работает бесконечно [2, с. 43]. Это означает, что проблема остановки машин Тьюринга не является рекурсивной. Но доказательство этого факта осуществляется в рамках именно формальной арифметики. В связи с этим в данной статье рассматриваются различные способы сведения подобных неалгоритмических проблем к некоторым совокупностям родственных проблем, каждая из которых является рекурсивной.

Пусть $< z;x >_t$ обозначает машину $< z;x >$, которая работает с ограничением t на число тактов работы, при этом если $< z;x >$ не остановилась за t тактов, то работа $< z;x >_t$ считается бесконечной. Для таких машин также существует рекурсивная функция $h_2(z,x)$, аналогичная $h_1(z,x)$. При этом, если ограничение t является фиксированным, то $h_2(z,x)$ является вычислимой с некоторым ограничение t_1 которое находится эффективно по t. А если ограничение t не фиксировано, то ни $h_2(z,x)$ не является вычислимой ни с каким ограничением [3].

Более интересные подобные ситуации наблюдаются при рассмотрении обобщенных вычислениях с частичными оракулами. Но в этом случае понятие рекурсивности заменяется F-разрешимостью. Пусть