



От редакции. Дорогие читатели журнала «Компьютерные инструменты в школе»! Напоминаем вам, что журнал, находящийся перед вами, является «потомком» журнала «Компьютерные инструменты в образовании» (КИО), в котором с 2008 года решено публиковать только научные статьи и материалы вузовского уровня. Однако редакция считает, что многие из публикуемых в КИО статей будут полезны учителям информатики и математики, а возможно и понятны интересующимся школьникам. Поэтому с текущего номера редакция решила помещать здесь краткие аннотации к публикуемым в КИО статьям. Оба журнала находятся на одном сайте www.kio.spb.ru/journal, и на этом сайте вы можете, как заказать отдельные статьи, так и подписаться на оба журнала.

Чтобы заинтриговать читателя, мы решили каждую публикацию предварить постановкой (в несколько упрощенном – «рекламном» – стиле) общей проблемы, на которую частичный или полный ответ дает данная статья.

*Кратко о содержании научно-методического журнала для вузов
«Компьютерные инструменты в образовании» №6, 2008 года*

1. МОЖЕТ ЛИ КОМПЬЮТЕР ДОКАЗЫВАТЬ ТЕОРЕМЫ?

Академик Ю.В. Матиясевич предлагает доступное изложение алгоритма Тарского. Алгоритм Тарского позволяет установить истинность или ложность любого утверждения про конечное количество вещественных чисел. Вместе с методом координат Декарта это позволяет автоматически доказывать широкий класс теорем элементарной геометрии. Изложенный здесь вариант алгоритма предназначен для первоначального знакомства с этой областью – его нетрудно понять, несложно запрограммировать, но полученная программа будет крайне неэффективной.

(Ю.В. Матиясевич. «Алгоритм Тарского», с. 4–14)

2. КАК КОМПЬЮТЕР РАЗБИРАЕТ ЛИТЕРАТУРНЫЙ ТЕКСТ И СПРАВЛЯЕТСЯ СО СМЫСЛОВЫМИ НЕОДНОЗНАЧНОСТЯМИ?

Аспирант кафедры системного программирования математико-механического факультета СПбГУ П.А. Громов излагает идею создания анализатора языка, выдающего в результате работы такую структуру данных, которая могла бы быть применима к как можно большему количеству прикладных задач, связанных с обработкой естественных языков. Показаны возможности этого формата по автоматическому разрешению лексических неоднозначностей и способы сведения его к более традиционным лингвистическим структурам.

(П.А. Громов. «Функциональное представление текста», с. 15–24)

3. КАК СОЗДАВАТЬ ПРОГРАММЫ, НА КОТОРЫЕ МЫ МОЖЕМ ПОЛАГАТЬСЯ?

Профессор кафедры информатики математико-механического факультета СПбГУ В.О. Сафонов объясняет смысл термина *trustworthy computing* (что в буквальном переводе означает *вычисления, заслуживающие доверия*). Он описывает существующие подходы к

решению этого вопроса и акцентирует внимание на разработанном под его руководством методе аспектно-ориентированного программирования, помогающего, в частности, превращать ненадёжные программы в надёжные.

(В.О. Сафонов. «Современные технологии разработки надежных и безопасных программ (trustworthy computing)», с. 25–33).

4. КАК КОМАНДЕ ИЗ ТРЁХ СТУДЕНТОВ РАСПРЕДЕЛИТЬ РАБОТУ, ЧТОБЫ ВМЕСТЕ РЕШИТЬ КАК МОЖНО БОЛЬШЕ ЗАДАЧ ПО ПРОГРАММИРОВАНИЮ НА ОДНОМ КОМПЬЮТЕРЕ?

На этот вопрос подробно отвечает студент СПбГУ ИТМО И.Р. Акишев, золотой призёр чемпионата мира среди студентов 2007 года (АСМ). В его статье рассказывается о командных студенческих турнирах по программированию формата АСМ ICPC. Автор излагает общие правила данных соревнований, обсуждает их отличия от личных соревнований по программированию, делится своим опытом участия в них, а также излагает советы и рекомендации будущим и нынешним участникам подобных соревнований и их тренерам.

Статья представляет интерес и для разработчиков систем компьютерного обучения. Последние должны так сформулировать критерии оценки учебной деятельности, чтобы они определили эффективную деятельность обучаемого, чтобы тактики поведения обучаемого, приводящие к хорошим результатам без освоения сути изучаемого материала, не могли реализоваться в системе.

Читатели-педагоги при прочтении статьи смогут наглядно увидеть, как правильно построенные правила соревнований АСМ приводят к эффективной организации труда команды. Оказывается, что для успешного выступления членам команды целесообразно хорошо документировать код программы, внимательно анализировать его перед тем, как послать на тестирование, сосредоточиваться на решении задач и не отвлекаться на пустое обсуждение хода соревнований, в то же время косвенно используя неудачи других команд для оценки сложности задач.

(Р.И. Акишев. «Об опыте участия в командных соревнованиях по программированию формата АСМ ICPC», с. 34–48)