

**МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
им. М. В. ЛОМОНОСОВА**

ФАКУЛЬТЕТ ВЫЧИСЛИТЕЛЬНОЙ МАТЕМАТИКИ и КИБЕРНЕТИКИ

**Первая Международная
научно-практическая конференция**

**Современные
информационные технологии
и ИТ-образование**

СБОРНИК ТРУДОВ

Под редакцией
проф. В. А. Сухомлина



УДК 004
ББК 32.81

К 250-летию МГУ имени М. В. Ломоносова

*Печатается по решению Рецензионно-издательского отдела
факультета Вычислительной математики и кибернетики
Московского государственного университета*

Рецензенты: профессор, д.ф.-м.н. А. Н. Томилин
профессор, д.ф.-м.н. Л. А. Калининченко

С56 Современные информационные технологии и ИТ-образование. Сборник докладов научно-практической конференции: учебно-методическое пособие. Под ред. проф. В. А. Сухомлина / отв. ред. Е. Н. Никулина. – М.: МАКС Пресс, 2005. – 892 стр.
ISBN 5-317-01413-1

В сборник трудов включены доклады Первой Международной научно-практической конференции «Современные информационные технологии и ИТ-образование», прошедшей в Московском государственном университете им. М. В. Ломоносова 19-23 сентября 2005 г. и посвященной 250-летию Московского университета. Целью конференции являлась интеграция усилий университетов, науки, индустрии и бизнеса в решении актуальной задачи построения востребованной национальной открытой системы ИТ-образования.

Материалы сборника предназначены для научных работников, преподавателей, аспирантов и студентов, интересующихся проблемами ИТ-образования, теоретическими, методологическими и прикладными вопросами в области информационных технологий.

УДК 004
ББК 32.81

ISBN 5-317-01413-1

© Издательский отдел факультета ВМК
МГУ им. М. В. Ломоносова

Как готовить информатиков высшей квалификации (анализ 20-летнего эксперимента)

*А. П. Бельтюкова, Н. Н. Непейвода, В. И. Родионов,
Удмуртский государственный университет, Ижевск,
nnn@uni.udm.ru*

Основное проблемное противоречие, без четкого осознания которого невозможно выстраивание системы эффективного образования в области информационных технологий: если на 1-ом курсе студента учат новейшим системам, то с 5-го он выходит, натасканный на морально устаревшие. Таким образом, здесь необходимо применять методологию опережающего обучения, основные положения которой были сформулированы В. Милошевичем (Владивосток).

Применительно к информационным технологиям это означает следующее.

Конкретная программная система заслуживает изучения не больше и не меньше, чем автомат Калашникова. Для рядового бойца это необходимо, для спецназовца, который должен уметь пользоваться любым оружием, это нужно лишь как пример.

Поскольку подавляющее большинство книг по информатике через 5-6 лет отправляются в макулатуру, нужно посмотреть, что же не устаревает, и учить классике, а не наиболее модному в данный момент.

Предыдущее положение кажется парадоксальным при сопоставлении с первым абзацем, но оно является частным случаем общего стратегического принципа. Лобовая атака проблемы либо позиций является худшим решением практически всегда, нужна стратегия не прямых действий.

Сегодня ясно, что Россия неконкурентоспособна с Индией и Китаем в области кодирования и программирования простых систем - неизбежные издержки, связанные с более высокой стоимостью рабочей силы и объективным уровнем затрат на поддержание жизненного уровня, приводят к тому, что российские фирмы сами привлекают индийских кодировщиков. Необходимо занимать свою нишу не ниже уровня *brainware*, а это вызывает необходимость пересмотра системы подготовки программистов. Российская фирма конкурентоспособна, если в ней на 5-7 программистов приходится один аналитик.

Соответственно, чтобы подготовить хороших программистов, нужно готовить аналитиков. Чтобы попасть в цель, нужно целиться выше цели. Все равно не более 20% студентов станут аналитиками, но оставшиеся станут программистами именно того класса, который нужен российским фирмам.

Поскольку имеется проблема согласования наших учебных программ с т. н. болонским процессом, нужно помнить, что можно перенимать опыт по-японски либо по-московски: или брать содержание, приспособливая его к своим национальным особенностям, или брать внешнюю форму, превращаясь в Иванушек из бессмертной комедии Фонвизина «Бригадир». Мы встали на японский путь, и поэтому данная статья является по сути своей беспощадной полемикой с книгой [1], представляющей и географически, и идеологически московский подход.

При этом необходимо оговориться, что сам подход, изложенный в книге [1], и предложенные в ней решения авторы считают разумными и правомерными, ЕСЛИ ОНИ НЕ БУДУТ НАВЯЗАНЫ ВСЕМ В КАЧЕСТВЕ ОЧЕРЕДНОГО СКОРОПАЛИТЕЛЬНОГО СТАНДАРТА.

Поэтому нужно думать вначале о существе дела, а уже затем о том, как это раскрасить в болонские цвета, тем более, что ценный опыт двоемыслия – как приспособить дело к дурацким бюрократическим бумажкам, слегка поменяв названия - у нас накопился за многие десятилетия.

В докладе анализируется опыт двадцатилетней работы на основе экспериментальных учебных планов, который дал отличные результаты. Он постепенно развивался в условиях тесного взаимодействия специалистов различных школ между собой и с бизнес-окружением (сформировавшимся на базе прежде всего наших выпускников). Многие основные черты данного опыта были заимствованы Новосибирским университетом при создании системы подготовки на факультете информационных технологий, и, в свою очередь, мы учли их опыт.

В итоге постепенной эволюции мы пришли к выводам, которые на первый взгляд кажутся революционными. Но по сути дела это перенос лучших черт сложившейся в России системы образования на новую область с учетом ее специфики. Сохраняя суть, мы не цеплялись за формы, когда они начинали противоречить конкретизации этой сути для совершенно других требований.

Суть российской методики образования – фундаментальность и аристократический подход. Студент еще не может определить, чему его нужно учить, и демократия здесь абсурдна. Фундаментальность же гарантирует, что человек получит те знания, которые не устареют в течение его активной жизни, а конкретике он обучится и сам. Жестко и

грубо говоря, создатели скоропалительных стандартов пытаются нас заставить готовить путан, которые через 4—5 лет сверхэксплуатации станут никому не нужны, а мы готовим как минимум куртизанок (гейш), у которых рабочий век не менее 20 лет.

За последние годы основную часть сектора ИТ-индустрии Ижевска заняли компании, возглавляемые выпускниками УдГУ, учившимися по экспериментальной программе. Когда представители компаний встречались со студентами, студенты были шокированы тем, что боссы и ведущие специалисты говорили им не о необходимости освоения новейших систем, а о необходимости обращать, особенно на первых курсах, основное внимание на фундаментальную подготовку, и в первую очередь на те курсы, которые «мозги в правильную сторону поворачивают». И первое место среди этих курсов заняла логика.

Аналогичная ситуация и в Новосибирске, где претензии типа: «Ваши выпускники не знают особенности работы Cold Fusion», сменились на осознание того, что выпускники полностью подготовлены к тому, чтобы за два месяца стать асами в любой конкретной системе, что возможно лишь при наличии общей фундаментальной подготовки. Существующие стандартные учебные планы в области преподавания информационных технологий в вузах неадекватны сложившейся ситуации, нужно создавать новую специальность, и единственный путь к этому — отпустить в «свободное плавание» тех, кто готов действовать в данном направлении, а затем обсудить полученный опыт и выработать рекомендации. Это соответствует и интересам студентов, потому что именно экспериментальные выпускники оказываются более всего востребованы рынком.

За 20 лет, естественно, выявились и недостатки. Прежде всего, не удалось наладить выпуск научной смены, в частности, потому, что подготавливаемые специалисты оказались слишком востребованы бизнес-окружением и не могут посвятить несколько лет развитию абстрактных исследований.

Далее, выяснилось, что требования министерства и жизни зачастую прямо противоположны, и экспериментаторы (после того, как ельцинский термидор закончился и опять стали пытаться наводить порядок порусски) находятся под сильнейшим бюрократическим давлением, любая уступка которому (переход на стандартные учебные планы) сразу же приводит к резкому снижению качества выпускников и, соответственно, к недовольству бизнес-окружения. И, наконец, стало очевидно, что предлагаемые решения НЕ МОГУТ НОСИТЬ АБСОЛЮТНОГО ХАРАКТЕРА, это — одна из альтернатив, предполагающая наличие и других.

Перейдем теперь к конкретике.

Специалистов высшей программистской квалификации в принципе можно готовить на базе хорошего гуманитарного курса (например, философии либо лингвистики), либо на базе математики. Ввиду отсутствия критической массы высококвалифицированных кадров в гуманитарных областях в России остается лишь один выход - готовить информатиков на базе математики.

Математика, сохраняя единство, все больше делится на две области с разным аппаратом и ценностями: численную и нечисленную. Нечисленная математика дает как раз тот «поворот мозгов», который требуется для успешной работы на средних и высших должностях в ИТ-производстве. Она великолепно поддерживает нынешние технологии индустриального программирования, что дает возможность всем успешно закончившим стать квалифицированными и востребованными специалистами. Отходы с 80% снижаются до приемлемого уровня 20%.

Известно, что хорошая организация обучения требует выделения главного курса, задающего общий тон и систему ценностей. В математике главной дисциплиной долгое время считалась геометрия, затем она уступила место анализу, а во второй половине прошлого века, особенно в ряде зарубежных университетов – алгебре. Логика, ставшая в XIX веке в значительной степени математической дисциплиной, накопила громадный потенциал идей, методов и результатов, задает тон в математическом языке нынешнего времени. Кроме того, логика, к счастью, не потеряла роль одной из ведущих гуманитарных дисциплин, математические методы естественно вписались в систему неформальных и полужформальных методов традиционной логики, и поэтому логика — одно из тех мест, где можно навести мосты между математикой и ее приложениями, прежде всего, нетрадиционными, не охватываемыми аппаратом математического анализа. Но и с традиционными приложениями логика работает не хуже, чем аналитические дисциплины, прежде всего ввиду большей концептуальной мощности ее идей и более глубокого анализа моделей. Поэтому логический цикл в настоящее время также подготовлен к тому, чтобы играть роль ведущего.

В целом необходимо подчеркнуть, что *выбор одного из четырех возможных базисных курсов диктуется квалификацией преподавательского состава и традициями конкретного университета, и недопустимо фиксировать его в жестком учебном плане, спускаемом сверху на основе опыта случайной кафедры случайного университета.* Хотя недопустимо делать выводы о том, какая из основ лучше, необходимо сравнивать подходы к преподаванию математики, базирующиеся на разных фундаментах. Разные базисные курсы благоприятствуют различным типам мышления и определяют типы математических моделей,

выбираемых в дальнейшем выпускниками. Так, аналитический цикл отработывает навыки манипулирования стандартными преобразованиями, символьными вычислениями и стандартными блоками рассуждений. Геометрический цикл всегда базировался на соединении образного и точного мышления. Алгебраический — на выделении абстрактных структур и их взаимных представлений. Логический цикл при соответствующем преподавании развивает творческое мышление, навыки понимания и критического анализа, приближая математику к гуманитарным наукам.

Самой трудной, но (теперь уже можно сказать) успешно решенной задачей реформирования учебного плана оказалась перестройка математического цикла. Логика (вместе с примыкающими к ней курсами алгоритмики, дискретной математики, теории программирования, математических структур) заняла ведущее положение. На второе место уверенно встала алгебра, в которой акценты смещены в сторону алгебраических структур. Анализ сохранил почетное положение одного из главных курсов, но поступился несколькими священными коровами, утратив положение безусловно первого. Поскольку топологические понятия необходимы для курса теории программирования, введен курс топологии. Таким образом, не снижая, и даже повышая, фундаментальность математического образования, математику приблизили к нуждам реальной жизни.

Еще раз подчеркнем, что предложенная система математических курсов является лишь одной из альтернатив, но анализ учебного времени показывает, что нынешняя «свобода» в рамках 10% учебного времени, за выходом из которых жестко следят министерские чиновники, это абсурд.

В книге [1] нам по сути предлагается взять худшие черты Болоньи и соединить их с худшими чертами России. Если мы отказываем в альтернативности студенту (мы **ВЫНУЖДЕНЫ** это сделать у себя, поскольку преподавателей, которые могут читать курсы на соответствующем уровне, недостаточно, и это объективно), мы **ОБЯЗАНЫ** предоставить альтернативность педагогам.

Как ни странно, намного легче оказалось реформировать информатику (может быть, потому, что преподаватели были уже подготовлены морально новой программой по математике). Основная суть реформы в том, что полностью отказались от «программирования на языке». Изучаются стили и методы программирования [2] на базе языков. На 1 курсе идут два параллельных курса программирования – традиционное и нетрадиционное. Традиционное включает методы структурного, автоматного и объектного стилей, нетрадиционное сосредоточено на сен-

тенциальном, событийном и функциональном стилях. Эти два курса тесно взаимодействуют, одна из лабораторных работ требует написать программу, использующую и традиционные, и нетрадиционные модули.

При выборе языкового инструментария основное внимание обращалось на концептуальную глубину и чистоту решений (вспомним проблемное противоречие, упомянутое вначале: никакой погони за модой на первых двух курсах – неизбежный вывод из него!). Поэтому сентенциальное программирование показывается на Рефале и Прологе, функциональное на Лиспе. Тяжелее всего оказалось с событийным. Этот стиль реализован в основном на системах, ориентированных на middleware, и, соответственно, перегруженных частностями, затрудняющими показ сути. Поэтому было принято парадоксальное решение: студенты программировали сценарии для игры Европа-2 на языке скриптов фирмы Paradox. В этом языке событийное управление представлено в чистейшем виде, а проблемы визуализации и проверки решений решаются с помощью окружающей системы.

В целом, с появлением концепции стилей программирования [3] поставлена задача разработки системы учебных языковых систем программирования для разных стилей. Ее уже начали решать инициативные группы студентов, и в этом году подана соответствующая заявка на грант.

Проблема обучения практическим методам и бизнес-процессам в программировании стала решаться лишь в последние два года, когда перед фирмами был жестко поставлен вопрос: скулите по поводу снижения качества подготовки студентов? Помогайте, причем не деньгами, которые все равно съест аппарат университета, а тем, чтобы вы сами и ведущие специалисты вели технические и бизнес-курсы. В этом отношении Новосибирскому университету (где также работал один из соавторов) было намного легче: там фирмы осознали данную проблему с самого начала. Но теперь уже можно сказать, что и она решена.

Отмеченный выше метод ее решения приводит к выводу, что и здесь излишняя детализация невозможна. От года к году состав технических и бизнес-курсов может меняться в связи с тем, что нужно максимально использовать сильные стороны приходящих преподавателей.

Болезненным местом, авгиевыми конюшнями, за которые противно, но необходимо браться, остается гуманитарный цикл. Если мы ориентируемся на подготовку аналитиков, он занимает равноправное место и в некоторых отношениях даже решающее. А говорить о качестве преподавания этого цикла, об отношении к нему авторов стандартов и учебных программ даже не приходится. Отрицательную роль здесь играет и появление «священных коров». Под предлогом, чтобы техники не вымели гуманитариев прочь, практически все (причем порою худшие,

скажем, политология либо культурология) гуманитарные курсы жестко фиксированы, и, соответственно, невозможно перестроить этот цикл, не вступая в прямое противоречие теперь уже не с обычаями чиновников, а с официальными бумагами. Поэтому тем, кто должен участвовать в эксперименте по подготовке АЛЬТЕРНАТИВНЫХ программ ИТ-образования, нужно дать карт-бланш и в данном вопросе. Проверку здесь могла бы осуществлять методическая комиссия из высококвалифицированных гуманитариев.

Курсом, который объединяет все три цикла, стал курс системного и логического анализа, идея которого — дать начала методов анализа сложных задач, учить неформально использовать формальные орудия и формально анализировать неформализуемые задачи, учить перепроверять формальное содержательным и содержательное — формальным.

Авторы благодарны руководству УдГУ, которое поддерживало их в труднейшей работе по перестройке образования и помогало отбить атаки чиновников. Один из авторов также приносит аналогичную благодарность руководству НГУ.

Литература

1. В. А. Сухомлин. ИТ-образование. Концепция, образовательные стандарты, процесс стандартизации. М.: Горячая линия – Телеком, 2005.
2. Н. Н. Непейвода. Стили и методы программирования. М.: Открытые системы, 2005.
3. Н. Н. Непейвода, И. Н. Скопин. Основания программирования. М.: Ижевск, РХД, 2003.