



Шальто Анатолий Абрамович

ПРОЕКТНЫЙ ПОДХОД ПРИ ОБУЧЕНИИ РАЗРАБОТКЕ ПРОГРАММ

Аннотация

Доклад на Ученом совете Санкт-Петербургского государственного университета информационных технологий, механики и оптики (СПбГУ ИТМО) 24.03.2009 года. В статье рассматриваются проблемы обучения проектированию программного обеспечения молодых специалистов в области информационных технологий, а также сохранения способной молодежи в университетах в качестве преподавателей.

Ключевые слова: автоматное программирование, молодые специалисты, проектирование программного обеспечение.

Встречаются два фермера. Один другого спрашивает:
– Как тебе удается получать от такой плохонькой коровы столько молока?
– А я ее каждое утро спрашиваю: «Что будешь давать, мясо или молоко?»

Анекдот от Алисы Фрейдлих

ПРОЕКТИРОВАНИЕ ПРОГРАММ

С 1970 года я работаю в «НПО «Аврора», которое занимается созданием систем управления судовыми техническими средствами. Разработка системы управления – это сначала создание проекта (проектной документации), а затем реализация его «в металле». Поэтому можно утверждать, что я «с молоком матери» осознал, *что в инженерной практике проектов без проектной документации не бывает*. Однако это относилось к системам, реализованным аппаратно.

С внедрением в контур управления ЭВМ появилась вторая компонента систем управления – программная. При этом я обратил внимание, что программы, в отличие от аппаратуры, *не проектировались, а писались*, что весьма странно, особенно для ответственных систем.

Это, видимо, было связано с тем, что программы, в отличие от аппаратуры, легко модифицировать, причем внесение изменений в них не требует привлечения специалистов других специальностей или организаций. Поэтому проектная документация на программы, если и выпускалась, то в значительно меньшем объеме, чем на аппаратуру, и *называлась такая документация не проектной, а программной*. Это нашло свое отражение в том, что номенклатура документов, выпускаемых на основе «Единой Системы Программной Документации» (ЕСПД), значительно меньше номенклатуры документов, разрабатываемых на основе «Единой Системы Конструкторской Документации» (ЕСКД).

В первое время в состав программной документации включались *блок-схемы алгоритмов*, которые в дальнейшем стали называться *схемами алгоритмов*. Однако

© А.А. Шальто, 2009

принципы их построения таковы, что они даже для относительно простых алгоритмов получались громоздкими. Поэтому, если такие схемы и строили на начальных этапах создания программ, то в дальнейшем их никто не корректировал. При этом происходила «рассинхронизация» схем алгоритмов с текстами программ, написанных «по мотивам» указанных схем.

Схемы алгоритмов в подавляющем большинстве случаев носили иллюстративный характер и не были исполняемыми – код по ним не генерировался. В тех редких случаях, когда указанные схемы были исполняемыми, они были неудобны для понимания человеком, так как из-за принципов построения на одном экране компьютера размещалось слишком мало элементов. Поэтому можно утверждать, что схемы алгоритмов не являются *гештальтными*, так как человек в основном понимает лишь ту информацию, которую воспринимает как единое целое. По моему мнению, негештальтна любая диаграмма, для просмотра которой необходимо использовать скролинг.

Со временем схемы алгоритмов из документации практически исчезли, за исключением телекоммуникаций, в которых для описания алгоритмов уже несколько десятилетий применяются *SDL*-диаграммы (*Specification and Description Language*), которые, наряду с рядом достоинств, обладают теми же недостатками, что и схемы алгоритмов. Это, видимо, является основной причиной того, что в других областях *SDL*-диаграммы не нашли применения.

Таким образом, к концу 80-х годов прошлого века сложилась ситуация, при которой в состав программной документации входили тексты программ и не включались диаграммы. Это привело к тому, что от программной документации оказались отрезанными специалисты разных уровней, которые не являются программистами, но обычно (особенно в технике) обладают большим опытом и глубокими знаниями предметной области. При этом, вместо знакомства с документацией, им приходится следить за функционированием программ в составе системы, когда «цена устранения ошибок» значительно выше по сравнению

с этапом спецификации задачи. По моему мнению, чем раньше и лучше указанные специалисты смогут понять, что собираются делать программисты, тем выше будет качество программного обеспечения (ПО). Я считаю, что это основной путь повышения качества ПО.

В это время я стал заниматься системами управления с ЭВМ и, не являясь программистом, понял, что тут что-то не так, и начал думать, как *проектировать программы*.

При этом из учебников по вычислительной технике было известно, что при разработке схем *широко применяются конечные автоматы*. Однако при создании программ автоматы использовались крайне редко. Они в основном применялись при написании компиляторов и описании сетевых протоколов. Кроме того, автоматы, которые используются в компиляторах, существенно отличаются от автоматов, применяемых в схемотехнике, так как они относятся к разным классам (абстрактные и структурные автоматы).

Также я обратил внимание и на то, что при обучении программированию автоматы обычно не использовались, за исключением спецкурсов по компиляторам. Книги по компиляторам писались математиками и для инженеров были весьма сложны. Это, видимо, и приводило к отторжению применения автоматов в практическом программировании.

Меня крайне удивила эта ситуация, и я стал разрабатывать технологию проектирования программ на основе применения *структурных* автоматов, задаваемых в виде *графов переходов*, так как я чувствовал, что «каждая компьютерная проблема решается еще одним уровнем абстракции» (Дэвид Уиллер).

К 1991 году (год первой моей публикации по этой теме) на основе автоматного подхода мною была предложена технология разработки ПО систем логического управления судовыми техническими средствами, которая была названа *SWITCH-технология*. Эта технология охватывала такие этапы жизненного цикла программ этого класса как спецификация, проектирование,

реализация, протоколирование и документирование.

При этом отметим, что в то время ни о каком унифицированном языке моделирования (*Unified Modeling Language – UML*) и поддерживающем его процессе (*Unified Rational Process – URP*) не было и речи. Правда, тогда в мире уже развивался подход, близкий к предлагаемому, который был назван его авторами *Statemate*. Этот подход, однако, и по сей день не является общепринятым при создании ПО управляющих систем, так как, например, программы для программируемых логических контроллеров обычно создаются на других принципах. Кроме того, отметим, что награда «нашла героев» лишь недавно, и создатели *Statemate* (среди них один лауреат премии Тьюринга) были удостоены премии *ACM Software Systems Award 2007*.

Технология автоматного программирования прошла апробацию в середине 90-х годов в НПО «Аврора» при создании ПО для ряда систем управления судовыми техническими средствами, разрабатываемыми, в том числе, и совместно с иностранными фирмами, для которых она была принципиально новой.

В 1995 году я подал заявку на грант Российского фонда фундаментальных исследований для издания книги по этой тематике, который выиграл в 1996 году. В 1997 году во время беседы с Дмитрием Александровичем Поспеловым (известный специалист по логическому управлению и искусственному интеллекту) родился термин *автоматное программирование*, который в настоящее время становится общепринятым. В начале 1998 года в издательстве «Наука» вышла моя книга «*SWITCH-технология. Алгоритмизация и программирование задач логического управления*» объемом 628 страниц, в начале которой этот термин был введен. Отметим также, что в 2004 году мною был предложен соответствующий англоязычный термин – «*Automata-Based Programming*».

С выходом этой книги в свет связано два события:

– знакомство с Никитой Туккелем, с которым мы стали в НПО «Аврора» разви-

вать предложенную технологию для задач управления, отличных от логических;

– знакомство с Алексеем Васильевым (студент кафедры «Компьютерные технологии» СПбГУ ИТМО, известной своими успехами в области олимпиадного программирования), который случайно прошел преддипломную практику в НПО «Аврора». Он написал под моим руководством магистерскую диссертацию на основе автоматного подхода и познакомил меня в конце августа 1998 года с Владимиром Глебовичем Парфеновым, который предложил почитать лекции по разработанной технологии для своих «мальчиков», «продвинутых» в области математики, физики и информатики.

Я стал читать лекции по автоматному программированию и уговаривать этих сильных студентов направить свои усилия на совершенствование излагаемой технологии. Однако никакие убеждения не действовали. Они учились на пятом курсе и практически все работали. Занятия проходили вечером, и ребята, естественно, уставали. Посещаемость была нормальной, но никакой их активности в направлении развития указанной технологии не наблюдалось. Проверка знаний по курсу осуществлялась в форме экзамена. Проку мне от всего этого не было практически никакого: ни технология не совершенствовалась, ни научные результаты не получались. Какой прок был студентам от моих лекций, я тоже не знал, хотя очень старался. Это, видимо, определило то, что мне предложили продолжить преподавать (по совместительству) на указанной кафедре, что было весьма почетно, особенно для человека, который не является программистом.

Через пару лет, кроме экзамена, в отчетность по курсу мною было добавлено выполнение *простейших* курсовых работ. Так как студенты были продвинутыми, то задачи на применение автоматного программирования они формулировали сами. Результаты этих работ сдавались на бумаге и на дискетах. При этом я наивно предполагал, что эти толковые ребята сделают что-то «человеческое», а со временем у меня появится какой-нибудь сильный аспирант, с которым мы будем приводить (с произво-

длительностью одна работа за встречу) указанные работы к такому виду, чтобы их можно было публиковать.

В 2002 году такой аспирант появился (Никита Шамгунов – математик и призер чемпионата мира по программированию). Однако наше знакомство с первыми двумя студенческими работами показало, что какими бы талантливыми ни были студенты, они остаются студентами и просто «отбывают номер», если преподаватель позволяет им делать это. Потратив на доводку первой работы около 100 часов, а второй – около 70 (эту работу мы исправляли уже втроем – вместе с Н. Туккелем), я понял, что до «победы» работу должен доводить никто иной, как сам автор, естественно, при моем участии, так как его кто-то должен учить, как это делать. При этом было ясно, что никакие правила, написанные на бумаге, и даже прототипы проблеме написания «человеческих» текстов не решат, поскольку программисты привыкли, что их ошибки обнаруживаются компилятором автоматически, роль которого в данном случае я уготовил себе. Стало понятно, что *без изменения подхода к выполнению и приемке работ их жизненный цикл мог быть только следующим: «работа на бумаге и в электронной форме – шкаф – помойка».*

В 2000 году мы с Никитой Туккелем создали модификацию предложенной технологии для построения событийных систем, реализуемых на процедурных языках программирования. Апробация этой модификации была проведена на примере проектирования ПО системы управления дизель-генератором. При этом был создан прототип программы и выпущена проектная программная документация объемом более 350 страниц, содержащая около ста диаграмм.

В 2001 году теми же силами была создана модификация предлагаемой технологии для построения управляющих систем, реализуемых на объектно-ориентированных языках программирования. Апробация этой модификации была проведена на примере проектирования ПО системы управления танком для известной компьютерной игры *Robocode*. Проектная программная документация была выпущена и в этом случае.

«ОБРАЗОВАТЕЛЬНЫЙ КОНВЕЙЕР»

Имея двух сильных аспирантов (Н. Туккеля и Н. Шамгунова), я хотел быстро совершенствовать технологию, но это не получалось, так как, во-первых, они работали, а во-вторых, хотели «жить». Меня это не устраивало, и я искал выход из сложившейся ситуации, который позволил бы проводить исследования и разработки значительно интенсивнее.

Такой выход я нашел в *обучении десятков студентов проектированию программ в ходе выполнения курсовых проектов*, используя разработанные нами автоматный подход и его модификации. Выбор объекта управления (моделирования) осуществлялся самими студентами, для того чтобы им было, по крайней мере, интересно делать проект.

Анализ результатов *простейших* курсовых работ, выполненных ранее, позволил понять причину того, почему я не получал желаемой отдачи от студентов:

- студенты имели знания в основном в области алгоритмов дискретной математики, а предлагаемый мной подход разрабатывался для проектирования ПО управляющих систем со сложным поведением;

- я был *недостаточно требователен к студентам и тратил на них слишком мало времени, на что они отвечали взаимностью*. Пока я тратил на прием курсовой работы минуты, студенты тратили на ее выполнение ненамного больше времени.

Ни о каком качестве, а тем более развитии технологии и получении научных результатов в такой ситуации говорить не приходилось.

Поэтому, начиная с 2003 года, я изменил методику преподавания и стал проводить обучение на основе проектного метода по-новому:

- проводится обучение студентов *третьего* курса;

- студенты разбиваются на команды, состоящие из одного или двух человек (при увеличении числа студентов до трех или четырех последние не делают ничего);

- число команд, формируемых из двух учебных групп, около 25;

- команды записываются в очередь таким образом, чтобы наши встречи проис-

ходили один раз в месяц или в полтора месяца. Тем самым формируется «образовательный конвейер»;

– каждая встреча продолжается около трех часов;

– не было ни одной работы, которую удавалось довести до «товарного вида» меньше, чем за три-четыре встречи;

– если я трачу на общение с одной командой 9–12 часов, то у команды на выполнение работы уходит как минимум в 10 раз (!) больше времени;

– проект, включая вычитанную проектную программную документацию, в обязательном порядке публикуется в сети Интернет на сайте <http://is.ifmo.ru>. При этом в качестве основного используется следующий критерий: ни студентам, ни мне не должно быть стыдно за сделанное. Документация должна быть написана так, чтобы по ней заинтересовавшийся человек, который не является программистом, мог в деталях понять что сделано. В настоящее время на сайте опубликовано более 150 проектов;

– работы имели настолько «товарный вид», что около 25 из них опубликовано на дисках журнала «Мир ПК», тираж которого в то время был 50 000 экз., что не характерно для публикации студенческих работ;

– по результатам курсовых работ, в которых получены новые научные результаты, пишутся статьи и материалы на российские и международные конференции;

– во многих случаях курсовые работы «перетекают» в бакалаврские работы, а в дальнейшем и в магистерские диссертации. В отдельных случаях сотрудничество со студентами «перерастает» в написание кандидатских диссертаций, но я думаю, что некоторые из них напишут и докторские диссертации по применению автоматного подхода в программировании.

При традиционной форме организации учебного процесса отдельные преподаватели информатики также «выкладывают» в Интернет результаты студенческих работ. Однако, в силу того что у преподавателей в большинстве случаев нет времени или желания доводить работы до «нормального» вида, они пишут на сайтах, что «работы как сделаны, так и выложены» (as is). Ког-

да такая запись отсутствует, во многих случаях «качество» работы видно «невооруженным взглядом». По моему мнению, при традиционном подходе к преподаванию получить что-либо толковое от студентов, которые практически все работают, *можно только случайно*.

Именно для «отлова» в том числе и случайных работ необходим «конвейер», который позволяет «просмотреть» *всех* студентов указанных групп, для того чтобы отделить найденные ими «зерна», если они имеются, «от плевел». Например, в этом году среди студентов был спортсмен, часто отсутствующий на занятиях из-за сборов, от которого меньше всего можно было ожидать чего-то неординарного. Так вот именно он выполнил такую работу, которую практически сразу можно публиковать.

Изложенный подход во многом совпадает с «системой ИТ-образования», принятой в одном из ведущих в этой области университетов мира – Массачусетском технологическом институте (МТИ), которую называют «академической базой тренировки морских пехотинцев». В учебном процессе этого института есть лекции, но они служат лишь в качестве пояснения к постановке задачи и концепций ее решений. Для непосредственного решения задачи студенты объединяются в группы, и работа в них есть основной элемент обучения. При этом главным инструментом являются так называемые «библии», которые служат своего рода базами знаний – коллекциями решений аналогичных задач, накопленных за годы. Изучая «библии», студенты пополняют их своими результатами». Принципиальное отличие этого подхода от предлагаемого нами состоит в доступности проектов в сети Интернет. При этом наш подход позволяет проводить обучение «на проектах» не только наших студентов, но и неограниченного круга лиц, интересующихся этой тематикой.

При написании статей, бакалаврских работ, магистерских и кандидатских диссертаций вновь работает «образовательный конвейер», описанный выше. При этом каждая работа оформляется и вычитывается нами до такой степени, чтобы в конечном

счете (после публикации на «бумаге») ее не стыдно было опубликовать и на указанном выше сайте. В настоящее время номенклатура «выкладываемых» на сайте работ расширена – публикуются также результаты лабораторных работ по генерации автоматов на основе генетического программирования.

В результате изложенной организации учебного процесса студенты знакомятся с проектированием программ и технологией автоматного программирования, учатся писать «по-русски», что является уникальным явлением для технических вузов страны. Они также учатся мыслить логично, что позволяет им, в частности, в одном и том же документе не называть один и тот же предмет по-разному. Они обучаются созданию проектной документации на ПО, что обычно не вызывает большой радости у *двадцатилетних* молодых людей. Многие из них *совершенствуют технологию* автоматного программирования и *получают научные результаты* в области программной инженерии, что также не очень характерно для российских вузов.

В результате выполнения работ студенты на практике понимают, что был прав Гельвеций, который утверждал, «что для того чтобы передавать свои мысли, надо гораздо больше ума, чем для того чтобы их иметь».

Студенты, успешно выполнившие проект, начинают понимать, что мнение, бытующее среди программистов, о том, что программный код заменяет документацию, не всегда правильное: заменяет, если код написан тобой, и не заменяет, если он написан другими людьми. Я был свидетелем, как один программист призывал двух очень сильных программистов разобраться в программе, которая, по его мнению, была хорошо написана, но для которой не было документации. При этом он удивлялся тому, что у его коллег не возникало желания, «засучив рукава», срочно взяться за работу. Это было связано, видимо, с тем, что они знали, что «до сих пор, несмотря на обилие свободного кода, нормальных программ мало, и неизвестно, существуют ли какие-нибудь программы, тексты которых не вызывают отвращения?».

Опыт показывает, что те студенты, которые прошли через «конвейер», по крайней мере, учатся писать по-русски, а те из них, кто случайно избежал этой участи, с указанным пробелом остаются на всю жизнь. Это, в частности, не позволило одному из студентов защитить магистерскую диссертацию, так как рецензент не смог понять, что в ней написано. Вот что о качестве выполняемых работ писал великий русский математик Л.С. Понтрягин: «Только хорошо выполненная работа дает радость! Выполненная небрежно, она вызывает отвращение и постепенно вырабатывает в человеке аморальное отношение к труду».

Проектный подход при обучении разработке программ с обязательным выпуском проектной документации в настоящее время становится все более актуальным, так как, по словам ректора МГУ академика В. А. Садовниченко, «отказ в ряде школ от написания сочинения на выпускном экзамене под предлогом, что больше нет такого вступительного экзамена, может привести к неспособности не только правильно выражать мысли, но и вообще мыслить», и это в условиях, когда «клиповое» мышление и так становится преобладающим.

Я встречаюсь со студентами каждый день, за исключением воскресений, праздников, непредвиденных обстоятельств и двадцати дней отпуска в августе. Мой опыт показал, что *получаемые результаты прямо пропорциональны затрачиваемым усилиям*. При этом подтверждается мысль бывшего генерального директора корпорации *Hewlett-Packard* Карли Фиорины: «Чем меньше требуешь от человека, тем меньше он и достигнет».

Подтверждается также и другая мысль К. Фиорины: «Уверенность руководителя в силах подчиненных является одним из сильнейших мотивирующих факторов», а также мысль всемирно известного бизнесмена Ли Яккоки: «Все управление, в конечном счете, сводится к стимулированию активности других людей». С возрастом мне стало ясно, что сегодня следует биться не за свою возможность заниматься наукой, а за то, чтобы эту возможность имели твои лучшие ученики, в особенности те, кто находится рядом.

Если раньше без использования «конвейера» я едва «справлялся» с двумя аспирантами, то при его применении мне удается успешно работать одновременно с десятками студентов и аспирантов, причем они могут «жить», так как наши встречи происходят далеко не каждый день.

В результате весьма длительного общения в ходе выполнения указанных выше работ мне удается достаточно хорошо узнавать молодых людей. После этого становится ясно, кто хочет и может заниматься научной работой, кто ориентирован только на работу в бизнесе и промышленности, кто является разгильдяем, а с кем мы не сошлись характерами!

Эта информация становится очень важной, когда представители фирм интересуются кадрами. Кстати, образование всемирно известной компании «Google» во многом связано с рекомендациями профессорами Стэнфордского университета Р. Мотвани (один из авторов книги «Введение в теорию автоматов (!), языков и вычислений») и Т. Винограда (известный специалист по искусственному интеллекту) Сергея Брина и Ларри Пейджа инвесторам Кремниевой долины. Нам тоже удалось помочь в формировании коллектива разработчиков ПО в компании «Скартел» (торговая марка *Yota*), которые в кратчайшие сроки разработали «математику» для первого в мире коммуникатора четвертого поколения *HTC MAX 4G*. Этот коммуникатор был продемонстрирован нашими выпускниками Президенту Российской Федерации Д.А. Медведеву на Инновационном конвенте, проходившем в конце 2008 года в Москве.

Обучение не является единственным результатом образования. В Законе РФ «Об образовании» 1996 года сказано, что «образование – это целенаправленный процесс воспитания и обучения». При этом отметим, что, несмотря на то, что после этого было принято более 25 поправок в этот закон, порядок выделенных слов не изменился.

По этому поводу Генеральный конструктор подводных ракетоносцев академик С. Н. Ковалев писал: «Подрастающее поколение мы еще чему-то учим, но, к большому сожалению, никак не воспитываем». По

словам академика В. А. Садовниченко, «нынешнюю молодежь нужно заразить лучшими идеалами по сравнению с теми, что распространены в обществе».

Проводимый мною эксперимент подтверждает слова профессора Владимира Ивановича Тимофеева, у которого я учился в ЛЭТИ, о том, что «учить можно только при личном общении, а не на лекциях». В подтверждение этого он приводил свой пример и утверждал, что основное влияние на его жизнь оказал один из создателей плана ГОЭЛРО академик Г.О. Графтио во время бесед, проходивших между ними, пока они шли 40 минут от ЛЭТИ до дома академика.

При длительном общении со студентами появляется возможность их воспитания (в том числе и на личном примере). При этом делается попытка сформировать в молодых людях некоторые человеческие качества, важные для работы в коллективе (например обязательность).

На воспитании и обучении работа в университете не должна заканчиваться, так как «отделить учебное от научного нельзя, но научное без учебного все-таки светит и греет, а учебное без научного – только блесит» (хирург Н. И. Пирогов).

Во время учебы в университете кто-то должен объяснить молодым людям, что:

- «существует лишь один вид наслаждения, который превосходит то, что человеку могут дать другие радости жизни: наслаждение от сознания, что научная идея хорошо реализована» (академик В.А. Глухих);
- «очарование, сопровождающее науку, может победить свойственное людям отвращение к напряжению ума» (геометр Г. Монж);

РЕЗУЛЬТАТЫ И ПЕРСПЕКТИВЫ

Изложенный подход в научном плане позволил:

- сформулировать новую парадигму программирования, названную «автоматное программирование»;
- опубликовать первую в мире монографию по этой тематике;
- опубликовать первый в мире тематический сборник статей студентов и аспирантов по этой тематике – 28 статей в «На-

учно-техническом вестнике СПбГУ ИТМО» № 53 за 2008 год;

- провести в 2005 году секцию по автоматному программированию (20 докладов) на международной конференции по компьютерной математике;

- сделать в 2008 году семь докладов по автоматному программированию на второй молодежной Всероссийской конференции по программной инженерии, в программе которой было всего 23 доклада;

- провести в 2009 году секцию по автоматному программированию (более 20 докладов) на конференции молодых ученых и специалистов СПбГУ ИТМО;

- выиграть в 2005–2008 гг. три государственных контракта по приоритетным направлениям науки и техники страны;

- выиграть несколько грантов РФФИ и Минобрнауки РФ (в том числе и на 2009 г.);

- защитить шесть кандидатских диссертаций, что не характерно для программистов, у которых нет материальной заинтересованности в этом нелегком процессе.

В настоящее время область исследования по автоматному программированию расширяется. При этом в нашем коллективе они проводятся в таких областях как, например, искусственный интеллект, мобильные роботы, документооборот, бизнес-процессы, а самое главное – верификация программ, что крайне важно при создании ПО для ответственных объектов (самолеты, вертолеты, ядерные реакторы и т. д.).

Инновации в области автоматного программирования состоят, например:

- в разработке инструментальных средств для его поддержки;

- в разработке виртуальных лабораторий для обучения этой парадигме программирования;

- во внедрении этой парадигмы программирования в практику создания ПО для ответственных объектов управления.

В результате проводимого эксперимента в настоящее время удалось сформировать иерархическую научно-образовательную структуру из аспирантов, прошедших подготовку по изложенной методике. Они сами формулируют темы по совершенствованию автоматного программирования и

руководят студентами кафедры «Компьютерные технологии».

Кроме указанных выше целей, мы ставим и еще одну: развить у студентов нашей кафедры качества лидеров. Для этого в следующем учебном году предполагается провести эксперимент по массовому обучению проектированию программ на основе автоматного подхода. При этом студенты пятого и шестого курсов, которые обучались на «образовательном конвейере», сами должны будут организовать такой «конвейер» с пропускной способностью в несколько сотен студентов других специальностей факультета «Информационные технологии и программирование» СПбГУ ИТМО.

Занятия по автоматному проектированию программ начинают проводиться и в других университетах, например в СПбГЭТУ (ЛЭТИ).

В настоящее время нами также ведется работа по внедрению описанного подхода в школы. При этом мы предполагаем, что если язык Паскаль был предложен для обучения программированию, то *автоматное программирование* было разработано для обучения проектированию программ. С помощью преподавателя программирования Ильи Дединского во второй половине 2008 года начаты работы в этом направлении в лицее «Вторая школа» (Москва), и уже получены удивительные результаты. Так, например, *восьмиклассник* Леонид Столяров создал еще одно инструментальное средство для поддержки автоматного программирования и использует его при решении прикладных задач. Уровень выполненной работы позволил ему участвовать не только в *Пятом Балтийском научно-инженерном конкурсе (BalticSEF)*, но и во «взрослых» научных конференциях. В новом учебном году мы предполагаем также внедрить указанный подход, по крайней мере, в одной из школ Санкт-Петербурга.

Предложенная организация учебного процесса позволяет также выявить лучших, которых целесообразно пригласить на постоянную работу на кафедре. При этом они должны хотеть работать в университете и понимать, какие преимущества дает работа в университете, если она хорошо организо-

вана. Именно лучшие молодые люди требуются для того, чтобы следующие поколения студентов хотели и могли эффективно учиться на кафедре. Для этого указанные молодые люди должны быть уникальными, по крайней мере, в двух из трех видов деятельности:

– преподавание в регулярном учебном процессе;

– преподавание олимпиадного программирования для школьников и студентов и проведение олимпиад различных уровней, включая создание задач, тестов, системы тестирования и т.д.;

– проведение научных исследований в области технологий программирования.

Вот что ответил Олег Табаков на вопрос, как ему удастся формировать такой сильный состав артистов в своих театрах: «Когда моя бабушка солила помидоры на продажу, то она лучшие оставляла себе. Так поступаю и я с выпускниками Школы-студии МХАТ».

Я придерживаюсь того же мнения, что и Олег Табаков и его бабушка, и поэтому предложил инициативу для в ИТ-отрасли России «Сохраним в университетах лучших!». При этом «содержание» лучших в университете берут на себя фирмы из бизнеса, одни – из желания получить квалифицированных выпускников, воспитанных «лучшими», а другие – из благотворительных целей.

При сохранении в университете лучших появляется надежда, что когда-нибудь и про наши университеты будут писать аналогичное: «МТИ буквально кишит учеными в области космонавтики. Стоит сделать шаг, и тут же столкнешься с Нобелевским лауреатом,

гением, который собирается стать лауреатом, или, на худой конец, профессором, окруженным необыкновенно умными и продвинутыми студентами. Это создает совершенно особую атмосферу. МТИ – университет, где дисциплина и тяжелый труд обязательны» (Карли Фиорина).

У нас в университете тоже есть то, что в МТИ называются «на худой конец» – профессор, окруженный необыкновенно умными и продвинутыми студентами. Однако Нобелевские лауреаты из них почему-то не получаются, и они даже к этому не стремятся, как и не стремятся к премии Тьюринга и т.п. Большинство из них после окончания университета куда-то исчезает – их не слышно и не видно, видимо, они улетают в космос! Хотя на самом деле все не так плохо, так как на командных студенческих чемпионатах мира по программированию команды нашего университета неоднократно побеждали команды МТИ, да и Стэнфорда тоже, в котором «также много умных людей».

Итак, в результате изложенной организации учебного процесса удастся решать следующие важнейшие задачи, которые стоят перед исследовательским университетом:

- воспитание;
- обучение;
- научные исследования (генерация знаний) и инновации;
- сохранение в университете уникальных молодых преподавателей и ученых.

Изложенное явилось составной частью нашей работы, которая была отмечена премией Правительства Российской Федерации 2008 года в области образования.

Abstract

Report at Scientific Council of Saint-Petersburg State University of Information Technologies, Mechanics and Optics (ITMO). March, 24 2009. The article discusses the problems of teaching the computer science students the technologies of software design and conservation of young professionals in universities as teachers.



Наши авторы, 2009.
Our authors, 2009.

Шальто Анатолий Абрамович,
лауреат премии Правительства РФ
2008 года в области образования,
доктор технических наук,
профессор, заведующий кафедрой
«Технологии программирования»
СПбГУ ИТМО, shalyto@mail.ifmo.ru