

Парфенов В. Г., Шалыто А. А.

**ПОДГОТОВКА ВЫСОКОКВАЛИФИЦИРОВАННЫХ СПЕЦИАЛИСТОВ В ОБЛАСТИ
ПРОИЗВОДСТВА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ И РАЗВИТИЕ ТЕХНОЛОГИЙ
ПРОГРАММИРОВАНИЯ НА КАФЕДРЕ «КОМПЬЮТЕРНЫЕ ТЕХНОЛОГИИ»**

1. Развитие российской индустрии производства программного обеспечения

В настоящее время перед Россией стоит задача интенсивного развития инновационного сектора экономики, связанного с высокими технологиями, и, в частности, с информационными технологиями и программированием.

В последние годы общепризнано, что именно здесь позиции России на мировом рынке высоких технологий являются наиболее сильными и перспективными. Накопленный за годы советской власти научный, образовательный и индустриальный потенциал в этой сфере, несмотря на все трудности, удалось сохранить и реализовать в последние десять–пятнадцать лет.

Мощный импульс роста отечественная индустрия производства программного обеспечения (ПО) получила на рубеже нового тысячелетия, когда по времени практически совпали российский финансовый кризис 1998 г. и постигший развитые страны в 2001 г. «кризис доткомов» – крах надежд инвесторов на получение быстрой прибыли от развития Интернет-технологий. За падением рынка акций высокотехнологичных предприятий последовал перевод разработок ПО и компьютерных технологий в страны с более дешевой рабочей силой. Таким образом, отечественные компании получили существенные конкурентные преимущества. В результате начался бурный рост российской индустрии разработки ПО. При этом в 2005 г. объем выполненных российскими компаниями зарубежных заказов достиг миллиарда долларов. В 2011 г. он превысил 3,0 миллиарда долларов и стал сопоставим с объемом экспорта российского вооружения. В России сфера производства ПО стала лидером по темпам развития среди других направлений высоких технологий, а технологии производства ПО включены в состав *критических технологий РФ*. При этом отметим, что и в новый перечень *приоритетных направлений развития науки, технологий и техники РФ*, утвержденный Президентом РФ Д.А. Медведевым в 2009 г., включены стратегические информационные технологии, в том числе и создание ПО.

В части развития индустрии создания ПО характерен пример Санкт-Петербурга, в котором в настоящее время в этой области работают около 400 компаний. В них трудятся порядка

20000 высококвалифицированных специалистов. Среди указанных компаний – такие ведущие мировые компьютерные корпорации, как *Intel, Google, Sun, Motorola, HP, EMC, Siemens, Alcatel, Borland, Samsung, LG*, которые открыли в нашем городе центры разработки ПО. В этих центрах в настоящее время работают несколько тысяч программистов высшей квалификации.

Характерно в этом смысле высказывание посетившего Санкт-Петербург в ноябре 2005 г. президента и генерального директора компании *Sun* Скотта Мак-Нили, наиболее ярко отражающее позицию крупных корпораций: «Если бы 15–20 лет назад я сказал, что наши ведущие разработчики будут работать в России, меня сочли бы сумасшедшим».

В настоящее время объем индустрии производства ПО в Санкт-Петербурге на порядок превосходит объем любой другой отрасли высоких технологий.

Развитие в Санкт-Петербурге индустрии высоких технологий и, прежде всего, области информационных технологий и программирования, было одной из важнейших частей программы развития города, которую предложила губернатор города В.И. Матвиенко. При этом была поставлена задача позиционирования Санкт-Петербурга как столицы российского программирования. Эта цель была поддержана и Президентами РФ В.В. Путиным и Д.А. Медведевым. Выбранная стратегия активно претворяется в жизнь. Сейчас можно твердо сказать, что будущее способных молодых петербуржцев обеспечено, и у них нет необходимости уезжать на работу за границу.

В настоящее время приведенные факты являются общеизвестными. Однако в девяностых годах, когда первые немногочисленные петербургские компании-разработчики ПО начали заниматься оффшорным программированием, выполняя несложные работы для западных заказчиков, ситуация выглядела отнюдь не такой ясной. Тем более что основным конкурентным преимуществом российских компаний того времени была не высокая квалификация специалистов, а их фантастически низкая оплата труда.

В начале девяностых годов через рухнувший железный занавес в нашу страну буквально хлынул поток современных компьютерной техники и информационных технологий, которая сделала неконкурентоспособными большинство советских разработок. В этой ситуации огромное значение имел правильный выбор основных направлений развития информационных технологий в России, который позволил бы нашей стране в тяжелейших экономических условиях того времени сконцентрировать на этих направлениях лучшие силы и «вскочить на подножку уходящего и набирающего скорость поезда» мировой компьютерной науки и индустрии.

Организованная в 1991 г. в СПбГИТМО В.Н. Васильевым и В.Г. Парфеновым кафедра «Компьютерные технологии» (КТ), как показал дальнейший ход событий, блестяще справилась с этой задачей. Одно из важнейших решений начала девяностых годов, принятых ими, было связано с развитием информационных технологий и программирования. Тогда еще были далеко не ясны перспективы взрывообразного развития этих технологий и повсеместного проникнове-

ния в самые различные области человеческой деятельности, а также фантастического возрастания роли программистов и разрабатываемого ими ПО, которое нематериально и поэтому непонятно многим инвесторам старого образца.

Организованная в это время кафедра КТ правильно определила «направление главного удара», сделав ставку на концентрацию на этом направлении и **интенсивную подготовку молодой российской интеллектуальной элиты** с целью выпуска специалистов, соответствующих мировым стандартам.

Сейчас кажется невероятным, но в то время организаторов кафедры упрекали в том, что она имеет странное название, поскольку существуют оптические технологии, технологии приборостроения, технологии обработки металлов резанием и т. д., но не существует компьютерных технологий. Несмотря на все трудности, кафедра была создана, и на ней была организована подготовка разработчиков высшей квалификации, а для этого со всей страны отбирались и приглашались школьники, одаренные в области точных наук и программирования.

Сейчас **результаты этого образовательного проекта получили мировое признание**. Победы студентов кафедры на чемпионатах мира по программированию внесли огромный вклад в формирование положительного образа нашей страны в глазах мирового сообщества. Преподаватели, выпускники и студенты кафедры внесли значительный вклад в формирование международного бренда «петербургский программист», которого отличают высокий уровень теоретической подготовки, творческий подход к делу и способность решать трудные нестандартные задачи. Этот бренд существенно помог в продвижении российских компаний-разработчиков ПО на мировой рынок, и формировании отечественной индустрии производства ПО. Выпускники кафедры занимают большое число руководящих позиций в петербургских компаниях, а также являются создателями многих из них. Достижения кафедры были отмечены правительственными наградами, Премиями Президента и Правительства РФ, а также Правительства Санкт-Петербурга.

Все это способствовало совершенному в последнее десятилетие прорыву российских компаний на международный рынок разработки ПО. В настоящее время отечественная индустрия производства ПО является самым динамично развивающимся направлением в области высоких технологий не только в Санкт-Петербурге, но и в стране в целом, и, как отмечалось выше, приближается по объему экспорта к индустрии вооружений.

2. Поиск школьников, одаренных в области информатики и программирования, и подготовка высококвалифицированных специалистов в области производства программного обеспечения

В последние несколько лет развитие отрасли производства ПО стало сдерживаться огромным кадровым дефицитом, поскольку наличие квалифицированных специалистов является

решающим для ее роста. Остроту ситуации можно проиллюстрировать на примере Санкт-Петербурга. Минимальный ежегодный рост численности специалистов, работающих в петербургских компаниях, составляет 10–15%. Следовательно, требуется ежегодно «вливать» в петербургскую индустрию разработки ПО не менее полутора-двух тысяч новых высококвалифицированных специалистов. Однако четыре ведущих петербургских вуза, являющиеся победителями конкурса инновационных программ (СПбГУ, СПбГУ ИТМО, СПбПУ, СПбГЭТУ), все вместе выпускают ежегодно чуть больше 200 профессиональных программистов, которые учились по этой специальности с первого курса. В результате еще в середине 2004 г. в Санкт-Петербурге разразился кадровый кризис в рассматриваемой отрасли, что привело к резкому увеличению зарплат программистов. При этом возникла опасность потери петербургскими компаниями конкурентоспособности на мировом рынке, особенно учитывая тот факт, что сейчас в Санкт-Петербурге чуть ли не единственным источником новых кадров стали студенты, в том числе, к сожалению, и младших курсов. Как показывают данные социологических исследований, в настоящее время в Санкт-Петербурге после окончания трех-четырех курсов вузов работают практически все студенты, обладающие более или менее выраженными способностями в области производства ПО.

Мировая практика показывает, что к работе в области информационных технологий подходит только порядка 3 % от численности населения. Если это применить к Санкт-Петербургу, то в 2007 г. из 36000 выпускников петербургских школ (в 2011 г. выпускников будет значительно меньше) только примерно 1000 были перспективны для работы в области информационных технологий, что в несколько раз ниже потребностей отрасли. Аналогичная ситуация складывается в Москве и других промышленно-развитых регионах. Таким образом, в стране возникает опасность резкого замедления темпов роста этого направления. Неслучайно, что для улучшения кадровой ситуации в Санкт-Петербурге в городской программе развития инноваций образование было поставлено на первое место.

С учетом демографического кризиса и роста зарплат, **в области решения «простых» задач Россия стала неконкурентоспособной по сравнению с Индией и Китаем.** Поэтому российская «ставка» на мировом рынке производства ПО – это **выполнение сложных проектов** с использованием высококвалифицированных специалистов, развитие и доведение инновационных идей до коммерческого использования, а также проведение научно-исследовательских работ. Таким образом, для нашей страны **особую ценность представляют наиболее талантливые специалисты**, способные стать лидерами проектов и научно-исследовательских работ в качестве руководителей и «генераторов идей».

Однако кадровый дефицит руководителей проектов проявляется в настоящее время в России еще в более острой форме по сравнению с описанным выше дефицитом разработчиков ПО. Одна из причин такой ситуации состоит в последствиях сложного социально-

экономического кризиса, пережитого нашей страной в 90-ые годы, и финансового кризиса, переживаемого с 2008 г. В результате первого кризиса из индустрии разработки ПО, как и из многих других областей высоких технологий, оказалась выведено целое поколение «сорокалетних», которые закончили вузы в конце восьмидесятых – начале девяностых годов. Аналогичный, но еще более острый характер имеют кадровые проблемы в области научных исследований. Последний кризис привел к сокращению объемов производства ПО, но потребность в высококвалифицированных кадрах в этой области не только не уменьшилась, а даже возросла, так как на рынке требуется все больше инновационных продуктов.

Ситуация осложняется тем, что в последнее десятилетие формирование научно-технической элиты и кадров высокой квалификации в области компьютерных технологий и программирования в России, также как и в других развитых странах, сталкивается с трудностями, вызванными **негативными социально-психологическими процессами**. Они обусловлены, в частности, сильным оттоком наиболее активных и способных университетских преподавателей в промышленность, общим падением интереса молодежи к занятиям точными науками, негативным воздействием на молодежь клиповой и интернет-культуры, компьютерных игр, телевидения и т. д., приводящих к сдвигам в психике молодых людей, препятствующим их долгосрочной сосредоточенной умственной деятельности (так называемый синдром перманентного частичного внимания), и общим уменьшением настроения молодежи на напряженный труд. В российских условиях описанные факторы и отмеченный выше демографический спад резко снижают уровень конкуренции между молодыми специалистами на рынке труда и их стремление к затрате усилий для повышения своей квалификации. Некоторые из указанных факторов лежат вне сферы образования, однако, несмотря на это, **в работу образовательных учреждений всех уровней должны быть внесены изменения**, которые позволят (хотя бы частично) компенсировать негативное влияние внешних условий.

Изменения должны быть внесены в учебный процесс также и потому, что программирование в последние годы превратилось в индустрию, а разработка ПО – в **производство**, которое неразрывно связано с процессом его **проектирования**.

Отметим, что в области производства ПО огромную роль играет возрастной фактор, который необходимо учитывать при выборе учебно-методических и организационных подходов. Для специалистов в области производства ПО характерны весьма ранняя профессиональная подготовка и раннее начало профессиональной трудовой деятельности. **Программирование – это занятие для очень молодых**. Опыт показывает, что изучение ряда фундаментальных курсов теоретической информатики и программирования надо начинать еще в школьные годы, и подчас даже в весьма молодом по общепринятым стандартам возрасте весьма трудно наверстать упущенное в школе, так как с годами резко падает способность к обучению программиро-

ванию. Молодой человек даже в двадцать с небольшим лет может оказаться слишком «старым» для начала целевой программистской подготовки.

Как показывает практика, малоэффективной является и переподготовка для работы в области производства ПО даже недавних выпускников вузов, окончивших математические или физические специальности, которые не получили интенсивной программистской подготовки в студенческие годы. Такая переподготовка была возможна в семидесятых – начале восьмидесятых годов, когда программирование еще не превратилось в индустрию. Однако за последние десятилетия технологии производства ПО (**software engineering**) получили огромное развитие как самостоятельное инженерное и научное направление, включающее большое число специальных дисциплин и технологических компетенций, предусмотренных соответствующими международными стандартами. **Знания, умения и навыки, полученные в двух-трех последних классах средней школы и на первых трех-четырех курсах вузов при изучении фундаментальных дисциплин в области теоретической информатики и технологий программирования, играют огромную (если не решающую) роль в становлении высококвалифицированных разработчиков, исследователей и руководителей в области создания ПО.**

В настоящее время в силу указанных выше причин, а также чрезвычайно высокой скорости развития информационных технологий, срок, за который разработчик может стать руководителем проекта в области создания ПО, существенно сократился по сравнению со сроками, существующими в традиционных инженерных отраслях. Можно привести много примеров, когда спустя всего два-три года после окончания вуза молодые специалисты становились руководителями технических и технологических направлений программистских компаний на позициях не только руководителей проектов, но и технических директоров. Такие сжатые сроки профессионального становления обуславливают необходимость проведения раннего, начиная со старших классов средней школы, поиска и подготовки будущих руководителей таких компаний.

Все это приводит к тому, что элементы программной инженерии должны вводиться в учебный процесс, начиная со старших классов средней школы и младших курсов вуза.

В связи с изложенным весьма актуальным является построение системы «школа – вуз – научные исследования – индустрия», обеспечивающей поиск, профориентацию, отбор школьников, одаренных в области информатики и программирования, формирование за счет дополнительного обучения, предпрофессиональной и профессиональной подготовки студентов (с последующим обучением в аспирантуре и докторантуре) высококвалифицированных специалистов в области производства ПО, которые могут выполнять функции разработчиков, исследователей и руководителей широкого круга научно-технических, научно-исследовательских и инновационных проектов.

Такая система должна использовать учебно-методические, технологические и организационные подходы, позволяющие нейтрализовать указанные выше негативные социально-психологические и демографические процессы.

Эти подходы, во-первых, должны обеспечить максимальную стандартизацию, формализацию, автоматизацию и унификацию соответствующих процедур, позволяющих **сократить потребность в высококвалифицированных педагогических кадрах**, что особенно важно в связи с их сильным дефицитом.

Во-вторых, должны быть разработаны современные формы самостоятельной работы студентов, которые «позволят молодым людям научиться учиться» [1] и будут использоваться наряду с традиционными формами, поскольку самостоятельная работа чрезвычайно важна при подготовке специалистов в области производства ПО. Эти формы должны отвечать психологическому настрою современного молодого человека на применение в учебном процессе сетевых технологий.

В-третьих, применяемые образовательные подходы должны быть активными, обеспечивая эффективное **взаимодействие** не только преподавателя с учащимися, но и учащихся между собой.

И, наконец, учащихся необходимо готовить к инновационной и научной деятельности в условиях сильной конкуренции, характерной для современного мирового рынка разработок и научных исследований в области информационных технологий.

Реализация изложенного потребовало создания в СПбГУ ИТМО специальной **системы** поиска перспективных молодых людей и новой организации учебного процесса для них.

3. Уровни «соударения умов»

Со времен пушкинского лицея известно, что главное в подготовке и сохранении талантов является атмосфера, обеспечивающая «соударение умов».

До сих пор в нашей стране такие условия создавались в основном только для *одаренных школьников* в специализированных учебно-научных центрах при известных университетах и физико-математических лицеях и школах. Назовем условия, созданные в этих учебных заведениях, **первым уровнем** «соударения умов». При этом, правда, в большинстве из них основное внимание уделялось и уделяется в настоящее время не информатике и программированию, а другим дисциплинам, в основном математике и физике.

С 1991 г. в Ленинградском институте точной механики и оптики (ЛИТМО), ныне Санкт-Петербургском государственном университете информационных технологий, механики и оптики (СПбГУ ИТМО), В.Н. Васильевым и В.Г. Парфеновым был сформирован **второй уровень** «соударения умов» – организована указанная выше кафедра КТ, в которую для обучения отби-

рали школьников, талантливых в области точных наук, информатики и программирования. Это обеспечило возможность «соударения умов» студентов.

С 2008 г. в рамках инициативы «Сохраним в университетах лучших!» (<http://www.savethebest.ru>), предложенной А.А. Шалыто, формируется **третий уровень** «соударения умов», на котором взаимодействуют выпускники кафедры КТ – молодые преподаватели и аспиранты, и наиболее сильные студенты, которые **работают на кафедре на постоянной основе**.

При этом опытные преподаватели делают **все возможное** для обеспечения «соударения умов» молодежи.

В настоящей работе описываются идеи и подходы, использованные нами в ходе формирования второго и третьего уровней указанной «пирамиды». В ходе ее создания осуществлялась подготовка высококвалифицированных специалистов и развитие технологий программирования на кафедре КТ, а в дальнейшем и на выделившейся из нее по предложению В.Н. Васильева и В.Г. Парфенова кафедре «Технологии программирования» (ТП), которой руководит А.А. Шалыто.

4. Применение проектного и соревновательного подходов в учебном процессе

При реализации указанной системы на кафедре КТ в качестве основных **совместно использовались и развивались проектный и соревновательный подходы**.

Целесообразность и эффективность использования **проектного подхода** в учебном процессе при подготовке высококвалифицированных специалистов в области производства ПО связаны с технологическим характером профессии программиста и изучаемых дисциплин. Компетенции в области программирования приобретаются только в ходе решения проблем, возникающих при реализации и доведении программ до работоспособного состояния.

«Практика работы в проектом подходе позволит выявить и передать современные способы организации мыслительной работы человека, что и является современным содержанием образования» [1].

Проектный подход воспитывает у учащихся аккуратность, умение доводить начатое дело до конца, самостоятельно мыслить, работать в коллективе, грамотно писать и правильно оформлять проектную документацию, дает возможность реализовывать различные формы самостоятельной работы студентов.

Важность использования **проектного подхода** в учебном процессе определяется также и тем обстоятельством, что если **разработка ПО** может выполняться без его проектирования, то **производство ПО** без проектирования невозможно.

Отметим, что **применение проектного подхода особенно актуально при обучении школьников**, и является в настоящее время принципиально важным и необходимым, так как в противном случае молодые программисты впервые сталкиваются с жесткими требованиями обязательной разработки проектной документации только в начале своей трудовой деятельности в «зрелом» для программистов возрасте. При этом весьма часто возникают острые конфликтные ситуации с заказчиками и коллегами, а времени для перестройки взглядов на процесс создания ПО и дополнительного обучения непосредственно в ходе трудовой деятельности практически не остается. Раннее обучение молодых людей выполнению проектов представляется в настоящее время тем более актуальным, что упомянутый выше синдром частичного перманентного внимания находится в противоречии с характером и духом требований к качественным программным проектам.

При реализации проектного подхода при подготовке специалистов в области производства ПО не удастся использовать методики проектного обучения, применяемые в классических инженерных областях, в которых общий цикл подготовки высококвалифицированного специалиста составляет не менее десяти-пятнадцати лет. В рассматриваемой области этот цикл составляет шесть-восемь лет, за который используемые технологии обычно изменяются. Поэтому учебный проектный процесс в области производства ПО должен быть организован иначе, что требует разработки новых научно-методических подходов к организации образовательного процесса.

Вторым эффективным средством для решения указанных задач является развиваемый на кафедре КТ **соревновательный подход**, суть которого состоит во введении элементов коллективных и индивидуальных интеллектуальных соревнований в учебный процесс, причем не только в форме предметных олимпиад.

Отметим, что в настоящее время осталось не так много средств, позволяющих **мотивировать молодых людей** на построение карьеры в области разработки ПО, поскольку кардинально изменилась система мотивации молодежи при выборе профессии. Практика последних почти двадцати лет показала, что одним из наиболее результативных из таких средств, которое доказало свою работоспособность даже в экстремальных политико-экономических условиях девяностых годов и в период настоящего финансового кризиса, является проведение олимпиад по информатике и программированию.

Широкое распространение в России олимпиад по этим предметам позволяет одновременно решать задачи профориентации, поиска, отбора и подготовки школьников и студентов, а также ряд важных методических и организационных вопросов.

В частности, на базе подготовки к олимпиадам в школах и вузах под руководством наиболее квалифицированных преподавателей (в том числе и весьма молодых) формируются коллективы способных, увлеченных и трудолюбивых студентов и школьников, ориентированных

на построение серьезной профессиональной карьеры в области информационных технологий. Во многом благодаря объединяющей и организующей роли олимпиад в вузах, средних школах и учреждениях дополнительного образования возникают неформальные сообщества преподавателей и учащихся, интересующихся фундаментальными и прикладными вопросами в области компьютерных технологий. Регулярно проводятся семинары и тренировочные занятия, ведется отбор и большая подготовительная работа.

Соревновательный подход воспитывает у учащихся способность работать в условиях конкуренции и дефицита времени, дисциплинированность, умение общаться и работать в коллективе (при проведении получивших широкое распространение командных олимпиад), проявлять инициативу и брать ответственность на себя, быстро ориентироваться в новой предметной области и быстро решать возникающие в ней задачи. Участие в тренировках и соревнованиях (особенно индивидуальных) – одна из наиболее эффективных современных форм самостоятельной работы учащихся.

Из изложенного следует, что **проектный и соревновательный подходы формируют у обучающихся дополняющие друг друга наборы качеств**, необходимых для высококвалифицированных специалистов в области производства ПО. При этом их совокупность образует **оптимальный набор качеств**, позволяющих специалистам, обладающим ими, эффективно выполнять **инновационные программные проекты**.

Совместное применение указанных подходов приводит к синергетическому эффекту в подготовке высококвалифицированных специалистов в области производства ПО, которые становятся способными ярко проявлять многие достоинства и полезные качества в своей профессиональной деятельности.

Таким образом, в результате применения развиваемых на кафедре КТ подходов образовательный процесс применительно к подготовке высококвалифицированных специалистов в области производства ПО кардинально изменяется в соответствии с основными тенденциями в развитии отечественного образования, и достигается главная в настоящее время цель образования – **не столько давать знания, сколько повышать обучаемость** [1].

Отметим, что разработка ПО относится к области человеческой деятельности, в которой производительность труда работников может отличаться в десятки раз. Поэтому **задача поиска и подготовки молодых людей, имеющих перспективы стать лучшими, является задачей первостепенной важности для создания инновационной экономики России**.

Исследования показали, что корреляция между результатами тестирований и показателями на рабочем месте близка к нулю [2]. Поэтому для построения надежной системы поиска, отбора и подготовки программистских кадров, а также получения обоснованных выводов об их способностях, В.Н. Васильевым и нами было предложено проводить специальным образом организованное обучение студентов и школьников старших классов, базирующееся на проектном

и соревновательном подходах. Так как необходимо обучать большое число учащихся, а надежные методики быстрого определения их способностей в области программирования, как отмечалось выше, отсутствуют, то существенно возрастает роль стандартизации, автоматизации и унификации учебного процесса, обеспечивающих требуемый уровень массовости при ограничениях на возможность привлечения высококвалифицированных преподавателей.

Ниже описывается созданная **В.Н. Васильевым, В.Г. Парфеновым, А.А. Шалыто и сотрудниками кафедры КТ инновационная система поиска и подготовки высококвалифицированных специалистов ПО**, основанная на использовании проектного и соревновательного подходов.

5. Основы проектного подхода

При реализации проектного подхода для подготовки специалистов в области производства ПО возникают трудности, связанные с необходимостью воспроизведения в учебном процессе реального процесса создания ПО, включающего такие этапы, как формирование архитектуры, кодирование, тестирование, верификация и документирование, а также этапов выполнения научно-исследовательских работ. Полная и качественная реализация этих этапов ограничиваются выделяемым на учебные проекты временем и трудностями привлечения к учебному процессу высококвалифицированных специалистов, имеющих опыт практической и/или научной работы в рассматриваемой области. Задача еще более усложняется, если принять во внимание наличие большого числа подходов к организации процесса создания ПО, парадигм программирования, инструментальных средств и т. д. Как было отмечено выше, элементы программной инженерии должны вводиться в учебный процесс, начиная со старших классов средней школы и младших курсов вуза. Поэтому необходимо обеспечить адаптацию реальных процессов создания ПО, имеющих место в различных компаниях, к уровню учащихся на ранних стадиях обучения. До последнего времени концепция, инструменты и педагогические приемы, позволяющие решить поставленную задачу при указанных ограничениях, отсутствовали.

На кафедре КТ разработана и реализована концепция сквозного непрерывного («школа – вуз») проектного обучения, базирующаяся на **специально разработанных** компонентах: парадигме автоматного программирования, инструментальных средствах и виртуальных лабораториях для его поддержки, интернет-библиотеках проектов и едином методическом подходе. Эти компоненты образуют базовый набор, позволяющий обеспечить достижение в учебном процессе компетентности обучающихся в области проектирования ПО в форме, доступной и понятной даже школьникам старших классов и студентам младших курсов. Предлагаемый подход позволяет, в частности, **ввести в учебный процесс этап создания качественной проектной документации**, который практически не удается реализовать при других подходах.

Одной из основных задач, возникающих при применении проектного подхода, является выбор парадигмы программирования, на базе которой должен строиться учебный процесс. При этом необходимо учитывать, в частности, такие факторы, как:

- сравнительная простота и доступность для школьников и студентов технологических аспектов выполнения проектов;
- возможность формулировки заданий с широкой и интересной для учащихся тематикой, требующих рассмотрения объектов из реальной жизни, в том числе и по тематике, им близкой;
- степень формализации процедуры разработки проектной документации;
- возможность построения сравнительно простых процедур контроля выполнения, сдачи и проверки законченных проектов;
- возможность унификации, стандартизации и автоматизации наиболее сложных процедур проектного учебного процесса.

Опыт работы кафедры КТ позволяет сделать вывод о том, что указанным требованиям отвечает учебный процесс, основанный на проектном подходе с применением **технологии автоматного программирования** [3], которая, кроме образования, имеет и самостоятельное значение, что будет показано в следующем разделе.

Эта технология, первоначально предложенная для систем логического управления, требовала своего развития применительно к другим классам программных систем. Она развивалась в течение последних двенадцати лет в рамках различных работ, проводимых студентами, аспирантами и сотрудниками кафедры КТ.

Одним из существенных результатов этих работ стала создание методологии применения проектного подхода на основе автоматного программирования при обучении студентов путем выполнения курсовых проектов [4].

В настоящее время трудовая деятельность большинства студентов, ориентированных на работу в области производства ПО, начинается, в основном, с четвертого курса. При этом старшекурсник оказывается вовлеченным в реальную деятельность одной из компаний программистского профиля, и его проектная подготовка проводится обычно в индивидуальном порядке в рамках выполнения реальных проектов в этой компании.

Поэтому, как было сказано выше, основной фундамент подготовки к проектной работе должен быть заложен у молодого человека к окончанию третьего курса, а вводить элементы проектного обучения необходимо еще в старших классах средней школы.

При этом отметим, что имеют место существенные различия между реализацией проектного подхода при обучении школьников старших классов, студентов младших курсов и студентов старших курсов.

В рамках реализуемой концепции обучения студенты выполняют проекты на первом и третьем курсах, а на втором курсе получают знания в области основных технологий программирования. Подчеркнем, что выполнение курсовых проектов включает не только написание работающего кода, как это обычно принято, но и **проектирование программ**, и разработку качественной **проектной** документации для них. Отметим, что получение проектной документации от студентов – весьма сложный процесс, так как они еще недостаточно зрелы для того, чтобы оценить важность документации при создании программ [5].

Задание **на первом курсе** состоит в разработке визуализатора одного из алгоритмов дискретной математики и документации к нему. **На третьем курсе** студенты **сами** предлагают темы курсовых работ из различных предметных областей, в рамках которых требуется не только разработать и отладить программный код со сложной логикой, но и подготовить качественную проектную документацию, а также выложить программу и документацию в открытый доступ в Интернет.

Эти проекты имеют общую «идеологическую базу»: проектирование, документирование и автоматный подход. При этом на первом курсе студенты применяют инструментарий, построенный на основе автоматного подхода, а на третьем – разрабатывают проект с применением автоматного программирования [6].

Опишем особенности реализации проектного подхода на первом курсе. Одной из основных целей выполняемого курсового проекта является обучение молодых людей очень несвойственному и неинтересному для них делу – проектированию, а самое главное, разработке и выпуску документации. На этом этапе учащиеся должны осознать то, что в инженерной практике **проектов без проектной документации не бывает**.

Разрабатываемый курсовой проект тесно связан с такими программистскими дисциплинами, как «Алгоритмы программирования и структуры данных» и «Дискретная математика», которые читаются на первом курсе. При этом каждый студент разрабатывает **проект** визуализатора одного из алгоритмов дискретной математики. Таким образом, студент должен не только продемонстрировать знания в области программирования, но и реализовать визуализатор, который наглядно показывает, как работает алгоритм. При этом формируются не только графические образы, но и текстовые комментарии. Это, в частности, позволяет приобрести знания в области создания пользовательских интерфейсов.

Для того чтобы визуализаторы **не писались** так, как это делается традиционно «в режиме вольной импровизации», **а проектировались**, на кафедре КТ был **разработан метод построения визуализаторов алгоритмов дискретной математики** [7], основанный на автоматном подходе. Построение визуализаторов для простых алгоритмов на основе этого метода может выполняться вручную [8], а для сложных – с помощью инструментального средства *Vizi* [9, 10]. Несмотря на то, что визуализаторы алгоритмов дискретной математики используются в

учебном процессе в ряде университетов мира, формализованный метод их построения не был известен, и обычно студенты не проектировали визуализаторы, а либо просто писали их, либо только использовали.

Многие из разработанных на основе предлагаемого подхода визуализаторы опубликованы на сайте <http://rain.ifmo.ru/cat/>, который в 2005 г. стал лауреатом конкурса «ИТ-образование в РУНЕТЕ», а визуализаторы **совместно с проектной документацией** опубликованы по адресу <http://is.ifmo.ru/vis/>.

Перейдем к изложению **методологии применения проектного подхода на третьем курсе**. Проекты могут выполняться как в индивидуальном порядке, так и командой, состоящей обычно из двух студентов.

Выбор темы проекта осуществляется самими студентами на основе предварительного знакомства с библиотекой проектов, представленных в Интернете на сайте <http://is.ifmo.ru>, разработанном на кафедре КТ. При этом студент может либо развить и усложнить задание уже выполненного проекта, либо предложить оригинальную задачу, которая его интересует. Без использования такой библиотеки практически невозможно сформулировать требования к качеству проектов, поскольку эту задачу можно решить эффективно только в результате знакомства с аналогами. Отметим, что в настоящее время отсутствуют другие интернет-библиотеки проектов, которые качественны и доступны по уровню сложности школьникам и студентам младших курсов.

Как отмечалось выше, при выполнении проектов используется автоматное программирование [6]. Применение этого подхода позволяет при разумных трудозатратах студентов качественно выполнять все этапы создания программ. При этом на этапе проектирования предлагается использовать автоматную модель для описания поведения программ. В процессе выполнения заданий студенты: проектируют программу в целом; строят автоматную модель ее поведения; проверяют путем валидации и верификации модель на корректность; по модели вручную или автоматически генерируют код; реализуют функции входных и выходных воздействий; интегрируют эти функции и сгенерированный код; при необходимости отлаживают программу; разрабатывают проектную документацию.

Практика показала, что для эффективного руководства проектом с целью обеспечения его качества и разработки документации на него преподаватель должен провести с каждой командой студентов три-четыре встречи, каждая продолжительностью около трех часов. При этом на индивидуальную работу студент тратит в среднем около ста часов.

Разработанная по проекту документация, исходные и исполняемые коды программы *в обязательном порядке* публикуются на сайте <http://is.ifmo.ru>.

Из изложенного следует, что применение автоматного подхода позволяет студентам **сконцентрироваться на проектировании программ, а не на их реализации**. Отметим, что

при традиционном обучении за время, отведенное на реализацию курсовой работы, студенты успевают только написать код, и, изредка, фрагменты документации. В описываемом подходе основное время уходит на проектирование, что позволяет студентам получить необходимые навыки в этой области, обеспечивает высокое качество работ и дает возможность и другим людям применять полученные результаты, так как в ходе выполнения работы автор создает **публикуемые в сети Интернет** текст программы, программную документацию, а также и **проектную документацию**, содержащую различные диаграммы, являющиеся не картинками, а математическими моделями, по которым может генерироваться код.

Практика показывает, что при соблюдении высоких требований к качеству **проектов** преподавателю, действующему по традиционной технологии, не удастся «справиться» даже с двумя-тремя обучающимися. Если же применять описываемый подход, базирующийся на автоматном программировании и одном из инструментальных средств для его поддержки, то преподаватель может успешно руководить несколькими десятками студентов. Таким образом, удастся организовать **«конвейерный способ» руководства курсовыми проектами**, что позволяет увеличить «производительность труда» преподавателя не менее чем на порядок. В результате работы «конвейера» преподаватель оказывается загруженным полностью, а студенты встречаются с ним, как показывает опыт, в среднем не чаще одного раза в месяц.

«Конвейер» необходим также и для выявления ребят, способных заниматься научной работой, так как позволяет достаточно внимательно «просмотреть» *всех* студентов одного года приема, обучающихся на кафедре КТ. Это иногда дает неожиданные результаты. Так, например, в 2008 г. среди студентов третьего курса был студент, часто отсутствующий на занятиях из-за спортивных сборов, от которого меньше всего можно было ожидать чего-то неординарного. Несмотря на это, именно он и выполнил такую работу, которую практически сразу можно было опубликовать.

В настоящее время на кафедре, **кроме курсовых работ**, студенты третьего курса **выполняют также и лабораторные работы** по верификации автоматных программ и генерации автоматов на основе генетического программирования, причем для проведения последних разработаны виртуальные лаборатории. У многих студентов эти работы перерастают в курсовые, которые, в свою очередь, часто превращаются сначала в бакалаврские работы, а затем и в магистерские диссертации. Те, кто выдерживает этот многолетний «марафон», потом защищают и кандидатские диссертации.

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 рассматриваются лишь для пояснения постановки задачи и выбора концепций ее решения. При решении задачи студенты объединя-

ются в группы, и работа в них и является основным элементом обучения. При этом главным инструментом являются так называемые «библии», которые являются своего рода базами знаний – коллекциями решений аналогичных задач, накопленных за годы. Изучая «библии», студенты пополняют их своими результатами. Принципиальное отличие этого подхода от используемого на кафедре КТ состоит в доступности наших проектов в сети Интернет. При этом предлагаемый подход позволяет проводить «проектное» обучение не только наших студентов, но и **дистанционно 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 (обычно после публикации на «бумаге»), ее не стыдно было опубликовать также и на указанном выше сайте.

За время использования проектного подхода на основе автоматного программирования на кафедре КТ студентами в 2002–2010 гг. было реализовано около 150 проектов, опубликованных в разделах «Проекты», «UniMod-проекты» и «Визуализаторы» сайта <http://is.ifmo.ru>. На сайте опубликованы также лабораторные работы, бакалаврские работы, магистерские и кандидатские диссертации, статьи, книги, отчеты по научно-исследовательским работам. Эта коллекция постоянно пополняется. Более двадцати из указанных проектов и десять студенческих статей опубликованы на дисках, являющихся приложениями к журналу «Мир ПК», тираж которого превышает 50 000 экземпляров, что нетипично для работ студентов, да и мало кого вообще.

Для повышения мотивации студентов к созданию качественно документированных проектов, нами организовано «Движение за открытую проектную документацию» [11], которое дополняет широко известные в мире инициативы в области создания сводного и открытого ПО (*Free Software Foundation* и *Open Source Initiative*). Оно может рассматриваться как одна из составляющих «Движения за открытое образование», организованного в мире в начале 2008 г. (*The Cape Town Open Education Declaration*).

При использовании описанного подхода, как отмечалось выше, можно решить проблему привлечения молодежи к научным исследованиям в области создания ПО. В настоящее время в эту область (особенно у нас в стране) по указанным выше причинам идут лишь единицы увлеченных молодых людей, а основная масса студентов даже не имеет представления о характере исследовательской работы. Поэтому во время учебы в университете мы стараемся объяснить молодым людям, что «существует лишь один вид наслаждения, который превосходит то, что человеку могут дать другие радости жизни: наслаждение от сознания, что идея хорошо реализована» (академик РАН В.А. Глухих) и что «**очарование, сопровождающее науку, может победить свойственное людям отвращение к напряжению ума**» (Г. Монж).

Из изложенного следует, что весьма важной является задача практического знакомства большого числа студентов с научно-исследовательской работой с целью повышения их заинте-

ресованности, мотивации к научной деятельности и проведения в дальнейшем отбора для работы в университете наиболее способных из них. И эту задачу можно решить при использовании описанного подхода. Практика показала, что в ходе выполнения проектов проявляются студенты, имеющие склонность к научной работе, а главное – желание заниматься наукой. Естественно, что такие учащиеся встречаются с преподавателем значительно чаще, чем остальные студенты.

В результате изложенной организации учебного процесса студенты не только знакомятся с проектированием программ и технологией автоматного программирования, но и учатся писать по-русски [12], что является уникальным явлением для технических вузов страны. Они обучаются созданию проектной документации на ПО, что обычно не вызывает большой радости у **двадцатилетних** молодых людей. Некоторые из них **совершенствуют технологию** автоматного программирования и **получают научные результаты** по программной инженерии, что также не очень характерно для современных российских вузов. Большинство из этих молодых людей после этого продолжают заниматься научной работой и защищают кандидатские диссертации в этой области.

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. Один из авторов настоящей работы был свидетелем того, как известный 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 наш «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 ни по форме, ни по содержанию. 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 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.

При реализации описанного подхода преподаватель встречается со студентами каждый день, за исключением воскресений, праздников, непредвиденных обстоятельств и двадцати дней отпуска в августе. Эксперимент в очередной раз 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 меньшего он и достигнет». Подтверждается также и другая ее мысль: «Уверенность руководителя в силах подчиненных является одним из сильнейших мотивирующих факторов», а также высказывание всемирно известного бизнесмена Ли Яккоки: «Все управление, в конечном счете, сводится к стимулированию активности других людей». С возрастом нам стало ясно, что **сегодня следует биться не за свою возможность заниматься наукой, а за то, чтобы эту возможность имели наши лучшие ученики, в особенности те, которые работают на постоянной основе на кафедре**.

В результате использования предлагаемого подхода удалось **сформировать иерархическую научно-образовательную структуру** из руководителей кафедр факультета информационных технологий и программирования, опытных и молодых преподавателей, аспирантов и студентов, прошедших подготовку по изложенной методике.

В заключение раздела отметим, что при выполнении проектов, как на первом, так и на третьем курсах, преподаватель выступает не в роли основного носителя знаний, а в качестве помощника (тьютора), что соответствует идеологии Болонского процесса. При этом повышается роль и степень участия студентов в собственном образовании.

В последние годы, как отмечено выше, большое значение приобрела задача использования **проектного подхода в средних школах**. Актуальность этой задачи обусловлена введением профильного обучения, требованием ориентации значительной части школьников, имеющих способности к точным наукам на дальнейшую работу в области производства ПО, а также прогрессирующим развитием негативных социально-психологических процессов. Организовать такую работу довольно трудно даже в наиболее сильных специализированных физико-математических школах в связи с тем, что указанные выше вузовские проблемы с высококвалифицированными педагогическими кадрами в средней школе носят еще более острый характер. Однако иногда нам это удается делать в качестве эксперимента в рамках внеклассной работы.

Расширение круга учащихся связано с созданием интернет-практикума [13], который позволяет школьникам освоить проектный подход на основе автоматного программирования [14]. При этом темы проектов школьников могут быть как специально ориентированными на их интересы и возможности, так и сформированными в результате адаптации студенческих проектов. Особое внимание уделено введению элементов игр и состязаний, которые позволяют школьникам в наглядном виде наблюдать и сравнивать результаты своих проектных работ.

Для первоначального знакомства с проектированием программ написана книга [15], в которой излагаются основы автоматного программирования.

Опыт внедрения автоматного подхода в лицее «Вторая школа» (Москва) свидетельствует о том, что **восьмиклассники** могут выполнять такие проекты, качество которых удивляет многих [16].

6. Применение соревновательного подхода в учебном процессе

Как отмечалось выше, соревновательный подход (в частности, участие в тренировочных занятиях и соревнованиях) формирует положительные качества, которые не могут быть развиты за счет других видов занятий (например, умение быстро решать задачи на сообразительность). Так, например, такие компании, как *Google* и *Microsoft*, которые добились выдающихся результатов в индустрии разработки ПО, широко используют проверку способностей кандидатов при приеме на позиции разработчиков при помощи испытаний, построенных на основе решения олимпиадных задач.

Для формирования у студентов указанных выше положительных качеств на кафедре КТ соревновательный подход введен в учебный процесс (по крайней мере, факультативно). В силу того, что этот метод обучения в нашей системе носит массовый характер, он требует модернизации по сравнению с его использованием для небольшого числа учащихся.

При традиционном применении соревновательного подхода необходимой процедурой является просмотр и обсуждение с преподавателем текстов программ. Реализация этой процедуры является дорогостоящей, поскольку требует больших затрат времени высококвалифицированных специалистов, которых, как отмечалось выше, в школах и вузах становится все меньше. В рамках предлагаемого подхода ручная процедура проверки правильности программных решений заменяется автоматической проверкой на системе тестов [17]. На кафедре КТ такой подход одним из первых в стране и мире был использован в 1993 г. при проведении первой в стране командной олимпиады школьников Санкт-Петербурга по информатике и программированию [18]. В дальнейшем развитие соревновательного подхода проходило в направлении совершенствования методики использования автоматического тестирования, развития форм ее применения в учебном процессе и совершенствования программной реализации.

Для развития методики автоматического тестирования были проведены работы по созданию спецификаций задач; разработке подходов к построению тестовых систем; определению сценариев тестирования; выбору структуры и программной реализации соответствующих сетевых аппаратно-программных комплексов; классификации и формированию банка заданий, структурированного по уровню сложности, тематике и другим классификационным признакам.

Традиционно наиболее широко соревновательный подход реализовывался в форме олимпиад. Поэтому одно из направлений работы было связано с организацией и проведением студентами, аспирантами и сотрудниками кафедры КТ в течение последних почти двадцати лет Санкт-Петербургских (в дальнейшем и Всероссийских) командных олимпиад школьников и студентов по информатике и программированию, четвертьфинальных и полуфинальных соревнований командного студенческого чемпионата мира по программированию. За это время под руководством В.Н. Васильева была создана получившая широкую известность **система интернет-поддержки** (<http://neerc.ifmo.ru>) полуфинальных соревнований Северо-Восточного Европейского региона [19], директором которых является В.Г. Парфенов, а также олимпиад школьников, которая сначала выполняла только информационную функцию. Затем ее функциональность была расширена путем добавления доступа к архивам задач, историческим справкам, ссылкам на региональные сайты, персоналиям и т. д. Система интернет-поддержки олимпиад позволяет осуществлять как автоматическую проверку задач, так и автоматическое управление соревнованиями.

Отметим, что в настоящее время требуется совершенно иной масштаб охвата контингента учащихся, так как из-за демографического кризиса и негативных социальных процессов «нельзя потерять ни одного перспективного молодого человека». В последнее десятилетие требования, обусловленные, с одной стороны, расширением масштаба олимпиад, а с другой – формированием у современного молодого человека психологического настроя на использование в учебном процессе современных технологий, привели к применению интернет-технологий [20], вплоть до появления интернет-олимпиад, в которых очные туры отсутствуют полностью. Интернет-технологии также стали широко использоваться при проведении тренировочного процесса. Олимпиады и интернет-олимпиады по информатике и программированию, включающие большое число отборочных соревнований, позволяют по существу организовать процесс поиска, отбора и обучения способных молодых людей в масштабе региона, страны и ближнего зарубежья.

Соревновательный подход позволяет в несколько раз увеличить число обучаемых молодых программистов за счет значительной интенсификации процесса обучения и приближения его к современным требованиям. Силами молодых преподавателей, аспирантов и студентов кафедры КТ создана система Всероссийских личных и командных интернет-олимпиад школьников по информатике и программированию, регулярно проводящихся, начиная с 2003 г. Санкт-

Петербургские интернет-олимпиады проводятся с 2006 г. [21]. Олимпиады предусматривают различные уровни сложности и подведение итогов для каждого уровня в отдельности. Указанные соревнования проводятся через интернет-представительства <http://neerc.ifmo.ru/school/io/> и <http://olymp.ifmo.ru>.

Внедрение соревновательного подхода для широкого контингента учащихся потребовало не только автоматизировать процесс проведения соревновательных туров, но и создать сетевую инфраструктуру их организации и проведения. При этом участвовать в Интернет-туре может любой человек, зарегистрировавшийся в системе. За счет автоматизации стало возможным проводить туры с числом участников до нескольких тысяч. Отметим, что, в отличие от официальных соревнований, участников Интернет-тура достаточно сложно идентифицировать, что приводит к необходимости принятия дополнительных мер безопасности. В частности, вероятность того, что в рамках официального соревнования на проверку будет отправлена вредоносная программа, чрезвычайно мала, в то время как для Интернет-туров эта ситуация не является необычной. Таким образом, система проведения соревнований через Интернет должна иметь существенно более высокую степень надежности, что определяет сложность ее разработки, которая также была выполнена на кафедре КТ.

В течение многих лет на кафедре проводятся факультативные тренировочные занятия для подготовки к олимпиадам по программированию, в ходе которых расширяются знания в области алгоритмов дискретной математики и развиваются практические навыки быстрой разработки и отладки программ. В этих занятиях участвуют не только наиболее сильные студенты с исключительными способностями, претендующие на победы международного уровня, но и значительно более широкий круг учащихся первого и второго курсов. Обычно в ходе тренировок проводятся занятия двух типов: практические (не менее двух занятий в неделю по пять часов) и лекционные, на которых излагаются особенности решения олимпиадных задач и избранные главы дискретной математики.

В рамках описываемой системы все более широкое применение находят различные формы тестирования – контрольные проверки знаний, умений и навыков учащихся, которые проводятся в соревновательной форме в учебных целях. Тестирование практически всегда проводится автоматически. В такой же форме могут проводиться и тренинги учащихся, и их самостоятельная работа. При проведении самостоятельной работы также могут быть введены соревновательные элементы, когда в таблицу результатов тестирований вводятся данные о результатах, показанных другими учащимися.

Автоматическое тестирование использовалось и для проверки уровня знаний и компетенций учащихся при составлении программ при компьютерной реализации части «С» единого государственного экзамена по информатике [22].

7. Инновационная система поиска школьников, одаренных в области информатики и программирования, и подготовки высококвалифицированных специалистов в области производства программного обеспечения

Как отмечалось выше, построение системы поиска одаренных школьников и подготовки высококвалифицированных специалистов в области производства ПО было начато В.Н. Васильевым и В.Г. Парфеновым в СПбГИТМО в 1991 г. Организационной формой для реализации этой системы стала специально организованная в том году кафедра КТ, которую уже в течение двадцати лет возглавляет В.Н. Васильев. На кафедре практически **с самого начала (с 1992 г.) проводилась подготовка бакалавров и магистров** по направлению «Прикладная математика и информатика», государственный стандарт для которого в наибольшей степени отвечал целям создаваемой системы.

Основной целью этой системы, как уже отмечалось, является не просто подготовка высококвалифицированных разработчиков ПО, а воспитание нового поколения специалистов, которые могли бы стать научными работниками, имеющими желание внедрить результаты своих исследований на практике, и руководителями инновационных проектов и компаний. Характер подготовки специалистов обеспечивает необходимую гибкость, выражающуюся в их способности быстро оценивать конъюнктуру рынка научно-технических достижений, изменять в соответствии с ней направления своей деятельности, а также проводить работы системного характера, лежащие на стыке ряда областей науки и техники. Отметим, что специалисты подобного профиля относятся к научно-технической элите государства и представители кадровых служб крупнейших компаний и корпораций ведут их поиск среди студентов вузов за два-три года до выпуска.

Для реализации указанной цели студентам дается широкая фундаментальная подготовка по математике, физике, теоретической информатике, программированию, технологиям программирования и производства ПО с использованием в учебном процессе проектного и соревновательного подходов. Преподают также предметы инженерного профиля.

Учебный план обеспечивает гармоничную подготовку бакалавров по дисциплинам математического, физического и компьютерного циклов, каждый из которых реализуется по усложненным учебным программам. По информационной насыщенности и сложности подготовка на кафедре технического университета не уступает уровню соответствующих специализированных факультетов классических университетов. Особое внимание уделяется изучению английского языка.

Студенты, получившие степень бакалавра, могут продолжать свое образование по магистерским программам «Технологии программирования» и «Технологии производства программного обеспечения», разработанным в рамках инновационной программы СПбГУ ИТМО. При этом обучение в магистратуре в обязательном порядке сочетается с работой в компании

или на кафедре. Магистратура позволяет не ограничиваться подготовкой разработчиков ПО, а обеспечивает возможность обучения специалистов, способных через сравнительно короткое время защитить диссертации, стать преподавателями или руководить компьютерными компаниями и инновационными проектами.

Высокий уровень научных работ студентов и аспирантов кафедры был, в частности, отмечен такими ведущими специалистами мирового класса в области информатики и программной инженерии как Никлаус Вирт [23], Бертран Мейер [24] и Джон Хопкрофт [25], которые по инициативе факультета информационных технологий и программирования были избраны почетными докторами СПбГУ ИТМО.

Отметим, что применение проектного подхода на младших курсах обеспечивает в дальнейшем качественное выполнение бакалаврских работ и магистерских диссертаций.

Реализация описываемой системы потребовала организации поиска и дополнительного обучения школьников, имеющих хорошие способности в области точных наук и разработки ПО, как среди петербургских школьников, так и школьников всех регионов России.

Для поиска талантливых школьников в Санкт-Петербурге использовалось два основных источника информации. Один из них был связан с установлением контактов практически со всеми учителями математики, физики и информатики ведущих специализированных физико-математических школ города. В первые годы рекомендации о наиболее способных школьниках собиралась в конце десятого класса и работа с рекомендованными ребятами велась в течение одиннадцатого класса. В последние годы информация стала собираться гораздо раньше, и работа ведется уже в течение нескольких лет – со школьниками восьмого-одиннадцатого классов и даже более младшими ребятами. При ее проведении **используются соревновательный и проектный подходы по описанным выше методологиям**. При реализации соревновательного подхода для школьников проводятся лекционные и практические занятия по базовым алгоритмам программирования, выдаются домашние задания, которые учащиеся выполняют дистанционно на базе системы автоматического тестирования программных решений, доступной через соответствующее Интернет-представительство (<http://neerc.ifmo.ru/freshman/>). Примерно один раз в месяц проводится очный соревновательный тур, на котором школьникам предлагаются для решения с использованием автоматического тестирования несколько задач. По результатам этих туров выводится общая оценка школьника по соревновательной части. Одновременно учащиеся выполняют проектную работу, которую защищают перед комиссией с соответствующей оценкой в конце одиннадцатого класса. Сумма оценок по соревновательной и проектной частям дает общую оценку по информатике и программированию, которая учитывается при зачислении абитуриентов на кафедру КТ по результатам ЕГЭ.

Второй путь получения информации о способных петербургских школьниках, включаемых в описанную схему подготовки, связан с привлечением дипломантов городских петербург-

ских олимпиад по точным наукам. В частности, широко используются проводимые сотрудниками, аспирантами и студентами кафедры КТ совместно с Комитетом по образованию Правительства Санкт-Петербурга городская олимпиада школьников по информатике, командная олимпиада школьников по информатике и программированию и интернет-олимпиады по информатике и математике.

Для поиска талантливых школьников из регионов используется сочетание соревновательного подхода и дистанционного обучения алгоритмам и методам программирования. Для проведения дистанционного обучения в течение всего учебного года дважды в месяц на сайте <http://neerc.ifmo.ru/school/io/> проводятся командные и личные интернет-олимпиады с двумя уровнями сложности, после окончания которых на этом сайте публикуются подробные разборы задач.

Абитуриенты, показавшие хорошие результаты на Всероссийской олимпиаде школьников по информатике, Всероссийской командной олимпиаде школьников по информатике и программированию, учебно-тренировочных сборах и летней компьютерной школе, в проведении которых традиционно принимают активное участие молодые преподаватели, аспиранты и студенты кафедры КТ, получают преимущество при поступлении на кафедру КТ в силу своей лучшей подготовленности к олимпиадам Российского Союза ректоров и ЕГЭ.

Организация отбора школьников, используемая на кафедре в течение последних почти двадцати лет, позволяет ежегодно формировать достаточно сильный и ровный состав первокурсников. Среди более двухсот студентов, обучающихся ежегодно на кафедре КТ, большинство отмечено дипломами региональных и городских олимпиад по точным наукам, а около четверти – всероссийских и международных. В отдельные годы на кафедру поступали до 40% от общего числа дипломантов Всероссийской олимпиады школьников по информатике, являющихся учащимися выпускных классов.

В результате применения описываемой в данной работе системы на базе кафедры КТ сформировался получивший международное признание центр подготовки одаренных программистов (**достижения этого центра приведены в Приложении**). Мировую известность принесли университету победы студентов в чемпионатах мира по программированию, которые в настоящее время имеют «вселенский» охват – в 2009 г. на стадии отборочных соревнований участвовало 7109 команд из 1838 университетов 88 стран мира, а в финале – 100 команд победителей полуфинальных соревнований, среди которых команды 27-ми американских и 16-ти китайских университетов. **Студенты кафедры КТ семь раз выигрывали полуфинальные соревнования чемпионата мира по программированию (в 1995 году в Амстердаме команда нашего университета не только победила, но и первой из команд российских вузов пробилась в финал чемпионата мира), шесть раз (в 1996, 2001, 2003, 2004, 2007 и 2010 гг.) становились чемпионами России по программированию, и три раза (в 2000, 2005 и 2006 гг.) –**

вице-чемпионами. В течение многих лет, начиная с 1995 года, студенты университета ИТМО неизменно выходили в финал чемпионата мира, где **в 2000 году завоевали серебряные медали, в 1999, 2001, 2003, 2005 и 2007 гг. – золотые, а в 2004, 2008 и 2009 гг. стали чемпионами мира и Европы по программированию.** При этом необходимо отметить, что эти выдающиеся результаты на чемпионате мира достигнуты **не одной, а семью различными командами университета по три студента в каждой.**

На сайте <http://snarknews.info> о победе СПбГУ ИТМО в 2009 г. сказано следующее: **«СПбГУ ИТМО после этой победы установил или повторил сразу несколько абсолютных достижений чемпионатов мира по программированию.** Во-первых, это единственный вуз, становившийся чемпионом мира три раза (также трижды *АСМ ICPC* выигрывал Стэнфордский университет в 1985, 1987 и 1991 гг., но два первых раза это было до того, как турнир был объявлен чемпионатом мира). Во-вторых, СПбГУ ИТМО стал единоличным лидером по числу завоеванных на чемпионатах мира золотых медалей – восемь. В-третьих, СПбГУ ИТМО во второй раз в истории (после побед СПбГУ в 2000 и 2001 гг.) удержал завоеванный титул чемпиона мира, но впервые это сделала команда, не пересекающаяся по составу с предыдущим чемпионом».

Благодаря успехам студентов кафедры КТ университет ИТМО занимает первое место в мировом и российском рейтингах университетов, составленном по результатам выступлений в чемпионате мира по программированию за последние годы. В 2004 г. тренер команд университета ИТМО А.С. Станкевич за выдающиеся успехи был награжден премией АСМ как лучший тренер Европы, а в 2008 г. – как один из лучших тренеров мира. Решением оргкомитета чемпионата мира ежегодная премия *DeBlasi Award* 2009 г. за большой вклад в развитие этих соревнований вручена представителям СПбГУ ИТМО **В.Н. Васильеву, В.Г. Парфенову и Р.А. Елизарову.**

Команду университета ИТМО, победившую на чемпионате мира по программированию в 2004 г., принял Президент России В.В. Путин. С командой университета, занявшей третье место на чемпионате мира по программированию 2007 г., встретился первый вице-премьер Правительства РФ Д.А. Медведев, который в ранге Президента РФ принял команду университета, победившую на чемпионате мира в 2009 г. С командами университета, победившими на чемпионатах мира 2004, 2008 и 2009 г.г., встречалась губернатор Санкт-Петербурга В.И. Матвиенко, которая награждала членов и руководителей команды почетными грамотами Правительства Санкт-Петербурга. Достижения команд университета, сформированных из студентов кафедры КТ, в финалах чемпионата мира по программированию, как уже отмечалось, внесли существенный вклад в формирование положительного образа России и Санкт-Петербурга в мировом обществе.

В 2003 г. за создание системы Всероссийских и Международных олимпиад по информатике и программированию и достигнутые в этих олимпиадах успехи ректору СПбГУ ИТМО, заведующему кафедрой КТ, профессору В.Н. Васильеву, декану факультета «Информационные технологии и программирование», профессору В.Г. Парфенову, старшим преподавателям кафедры КТ Р.А. Елизарову и А.С. Станкевичу была присуждена Премия Президента РФ в области образования.

В последние годы в мире стали также проводиться и индивидуальные соревнования по программированию, которые названы спортивным программированием. Методика подготовки к командным соревнованиям по программированию, разработанная на кафедре, позволила и в этих соревнованиях добиваться выдающихся результатов на международном уровне студентам, аспирантам и выпускникам кафедры КТ [26].

Например, в 2006 г. (см. Приложение) на соревнованиях *Google Code Jam Europe* (Дублин) представители кафедры КТ заняли третье, шестое и десятое места; на соревнованиях *Google Code Jam* (Нью-Йорк) в финал вышли пять представителей нашей кафедры, двое из которых заняли третье и седьмое места; на соревнованиях *TopCoder Open* (Лас-Вегас) в финале участвовали два представителя кафедры, один из которых занял шестое место; на соревнованиях *TopCoder Collegiate Challenge* (Сан-Диего) в финале принимали участие три студента нашей кафедры, один из которых занял четвертое место.

Представители кафедры КТ добивались успехов и в последующие годы. Так, например, в финале соревнований *TopCoder Collegiate Challenge*, который проходил в ноябре 2007 г. в Орландо, участвовали два наших студента.

В 2011 г. А.С. Станкевич, участвуя в предварительных соревнованиях, в которых принимало участие более 3000 программистов со всего мира, попал в первую десятку участников очного тура олимпиады, проводимой компанией *Facebook*.

В результате успехов на соревнованиях *TopCoder* целый ряд молодых людей, связанных с нашей кафедрой, имеют высокий рейтинг в мировой классификации, что еще больше повышает авторитет кафедры КТ и СПбГУ ИТМО в целом в области подготовки высококвалифицированных программистов, как в стране, так и в мире.

Еще одно международное достижение кафедры КТ состоит в том, что молодые ее сотрудники с 2010 г. тренируют швейцарских школьников в Давосе, а студентов – в Цюрихской высшей политехнической школе (ETH), в которой учились и/или работали 21 Нобелевский лауреат. Эти тренировки привели к тому, что команда ETH впервые в своей истории заняла первое место в своем полуфинале и получила **единственную путевку** в этом регионе на финал чемпионата мира по программированию. **Команду СПбГУ ИТМО на этих соревнованиях «выведет» А.С. Станкевич, а команду ETH – Ф.Н. Царев**, причем из-за особенностей названий на английском языке эти команды на соревнованиях будут располагаться рядом.

За последние почти двадцать лет на кафедре КТ сформировался методический, технологический и организационный Центр проведения интернет-олимпиад по информатике и программированию, Всероссийских студенческих и школьных олимпиад, а также четвертьфинальных и полуфинальных соревнований Северо-Восточного Европейского региона чемпионата мира по программированию.

При создании и развитии этого центра большую роль сыграла описанная система. В критических социально-экономических условиях девяностых годов сложилась ситуация, когда весьма сложные технологии и методология проведения олимпиад по информатике и программированию и подготовки их участников на уровне, соответствующем международным стандартам, поддерживались в стране в значительной мере студентами, прошедшими в свое время школу олимпиад российского и международного уровней. Благодаря высокой «концентрации» одаренных молодых людей на кафедре КТ, в эти годы под руководством В.Н. Васильева удалось организовать молодежную «цепочку», по которой проводилась передача знаний, методологии и технологий проведения олимпиад по информатике и программированию, а также по подготовке их участников.

Эти же молодые люди обеспечивали в то время всю «интеллектуальную» часть городских и районных олимпиад для петербургских школьников, подготовку сборных команд школьников Санкт-Петербурга, сборных команд СПбГУ ИТМО, участвовали в подготовке сборных команд школьников России.

В результате удалось преодолеть трудный период и сохранить указанные методики и технологии до начала существенного улучшения социально-экономического положения страны, позволившего победителям международных олимпиад Р.А. Елизарову, М.А. Казакову, А.С. Станкевичу, Г.А. Корнееву, П.Ю. Маврину, Ф.Н. Цареву и М.В. Буздалову занять ведущие позиции в проведении указанных олимпиад в последние годы.

В 2008 г. за создание инновационной системы подготовки высококвалифицированных программистских кадров, базирующейся на использовании проектного и соревновательного подходов, ректору СПбГУ ИТМО, заведующему кафедрой КТ, профессору В.Н. Васильеву, декану факультета «Информационные технологии и программирование», профессору В.Г. Парфенову, заведующему кафедрой ТП, профессору А.А. Шалыто, доцентам кафедры КТ М.А. Казакову и Г.А. Корнееву была присуждена Премия Правительства России в области образования.

В 2010 г. за разработку учебно-методического обеспечения для подготовки высококвалифицированных программистов доценту кафедры КТ Г.А. Корнееву и старшему преподавателю этой кафедры С.Е. Столяру была присуждена Премия Правительства Санкт-Петербурга в области образования.

8. Инициатива «Сохраним в университетах лучших!»

Эффективность работы описанной выше системы определяется не только отбором талантливых школьников и их отличной подготовкой при обучении на кафедре, но и за счет **инициативы «Сохраним в университетах лучших!»**, которую поддержал ряд отечественных компаний. Это позволяет оставлять для работы в университете молодых талантливых выпускников и студентов, что было бы невозможно при использовании только бюджетного финансирования. При этом надежда только на молодых совместителей не выдерживает критики, так как «на бегу» нельзя обучать даже бегу. Вот как охарактеризовал образование «на бегу» академик РАН, генеральный конструктор атомных подводных ракетноносцев С.Н. Ковалев: «В этой ситуации еще как-то можно обучать, но нельзя воспитывать».

Последнее замечание является чрезвычайно важным, так как обучение не является единственным результатом образования. В Законе РФ «Об образовании» 1996 г. сказано, что «образование – это целенаправленный процесс **воспитания и обучения**». При этом, несмотря на то, что после этого было принято более 25 поправок в этот закон, порядок выделенных слов не изменился.

В настоящее время на постоянной основе на кафедре КТ работают:

- **Андрей Станкевич** (год рождения – 1981) – выпускник кафедры КТ 2004 г., лауреат премии Президента РФ 2003 г. в области образования, лауреат премии Правительства Санкт-Петербурга педагогам-наставникам, подготовившим победителей и призеров Всероссийских олимпиад школьников 2009 г., лауреат Молодежной премии Санкт-Петербурга 2010 г., обладатель серебряной и золотой медалей чемпионатов мира по программированию 2000 и 2001 гг., **тренер всех команд университета ИТМО**, начиная с 2000 г., доцент кафедры.
- **Георгий Корнеев** (1981 г.) – выпускник кафедры КТ 2004 г., лауреат премий Правительства РФ 2008 г. и Правительства Санкт-Петербурга 2010 г. в области образования, выпускник кафедры КТ 2004 г., обладатель серебряной и золотой медалей чемпионатов мира по программированию 2000 и 2001 гг., кандидат технических наук, доцент кафедры.
- **Павел Маврин** (1984 г.) – выпускник кафедры КТ 2008 г., лауреат премии Президента РФ 2002 г. за успехи на международной школьной олимпиаде по информатике, лауреат Молодежной премии Санкт-Петербурга 2003 г., лауреат премии Правительства Санкт-Петербурга педагогам-наставникам, подготовившим победителей и призеров Всероссийских олимпиад школьников 2009 г., чемпион мира по программированию 2004 г.
- **Федор Царев** (1986 г.) – выпускник кафедры КТ 2009 г., лауреат Молодежной премии Санкт-Петербурга 2007 г. и премии Правительства Санкт-Петербурга в области инноваций 2009 г., чемпион мира по программированию 2008 г.

- **Максим Буздалов** (1987 г.) – чемпион мира по программированию 2009 г., закончил бакалавриат кафедры КТ в 2009 г.
- **Михаил Царев** (1988 г.) – лауреат премии Правительства Санкт-Петербурга в области инноваций 2009 г., закончил бакалавриат кафедры КТ в 2009 г.

6 мая 2009 года на встрече с победителями очередного чемпионата мира студент четвертого курса кафедры КТ М. Буздалов изложил Президенту РФ Д. А. Медведеву идею предлагаемой инициативы, которая была одобрена Президентом: **«Это, кстати, хорошая штука. То есть задача заключается в том, чтобы не вытаскивать хорошо подготовленных, одаренных студентов, а просто, чтобы они финансировались за счет компаний и не уходили из университетов, чтобы там эта микросреда сохранялась, как я понимаю. Это хорошая идея. Главное, чтобы к этому были компании готовы».** (<http://kremlin.ru>).

На сегодня нас поддерживают: группа компаний *Транзас* (президент – Николай Лебедев), ООО *Скартел* (генеральный директор – Денис Свердлов), компания *JetBrains* (генеральный директор – Сергей Дмитриев), компания *SPB Software* (исполнительный директор – Василий Филиппов), компания *developers* (генеральный директор – Андрей Нарвский), компания *ДевиноСМС* (генеральный директор – Павел Ушанов), компания *Одноклассники.ru* (президент – Илья Широков), бизнес-центр «Мартышкино» (управляющий – Олег Давыдов). Более подробно с изложенной инициативой можно ознакомиться на сайте <http://www.savethebest.ru>

Выпускники нашей кафедры могут иметь склонность как для работы в промышленности, так и для работы в университете. Естественно, что руководители кафедр факультета информационных технологий и программирования стремятся сохранить в университете молодых людей, ориентированных на университетскую карьеру. Это в основном те, кто хочет и может бесконфликтно работать в коллективе, добился выдающихся достижений в студенческие годы и/или обладают незаурядными способностями, по крайней мере, в двух из четырех областей:

- преподавание дискретной математики, информатики и программирования студентам и школьникам;
- проведение олимпиад студентов и школьников всех типов и уровней по информатике и программированию в Санкт-Петербурге, России и мире (создание задач, тестов, систем проведения и автоматического тестирования олимпиадных задач, в том числе, и для соревнований, в которых принимают участие сотни и тысячи участников);
- подготовка студентов и школьников к соревнованиям по информатике и программированию, как командным, так и личным, всех уровней, включая чемпионаты мира, в том числе и таких молодых людей, которые не связаны с СПбГУ ИТМО (например, в ходе летних компьютерных школ);
- проведение научных исследований.

Предпринимаются усилия для привлечения в университет на работу не только выдающихся выпускников, но и талантливых студентов. Материальная поддержка молодых талантов компаниями в рамках этой инициативы является основой для сохранения молодежи на постоянной работе в университете, так как учебный процесс должен происходить постоянно, а любая грантовая поддержка имеет ограниченный срок действия. Сказанное не исключает участия кафедр факультета в конкурсах на выполнение государственных контрактов по научным исследованиям, а наличие выдающихся молодых людей, постоянно работающих в университете, позволяет, как отмечалось выше, выигрывать конкурсы на получение этих контрактов.

Научные исследования, проводимые на кафедре

9.1. Автоматное программирование

Основы автоматного программирования, которое также называется «SWITCH-технология» или «программирование с явным выделением состояний», были разработаны в 1991 г. [27]. Термин «автоматное программирование» родился в 1997 г. в ходе беседы одного из авторов настоящей статьи с Д.А. Поспеловым на конференции по мультиагентным системам, проходившей в поселке Ольгино под Санкт-Петербургом, и был впервые использован во введении к работе [6]. На английский язык этот термин переводится как «Automata-Based Programming», который был предложен в работе [28].

Автоматное программирование было активно поддержано В.Н. Васильевым, который продолжает внимательно следить за развитием этого направления, оказывая ему всемерную поддержку.

9.1.1. Автоматное программирование как стиль программирования

Автоматное программирование является одним из стилей программирования. В работе [29] отмечается: «Термин **«автоматное программирование»** принадлежит, насколько нам известно, А.А. Шалыто. Во всяком случае, ему принадлежит заслуга в его развитии **вопреки моде и мнению большинства»**».

Упрощенная трактовка автоматного программирования состоит в том, что это стиль программирования, при использовании которого поведение программ предлагается описывать конечными автоматами, которые в дальнейшем изоморфно преобразуются в код.

При этом отметим, что автоматы давно применяются в программировании при построении компиляторов и протоколов. Автоматы используются от случая к случаю также и при решении других задач.

В работе [29] подход с использованием автоматов для описания поведения программ был определен как стиль программирования, названный «программирование от состояний». В

указанной работе отмечалось, что «это, пожалуй, самый старый стиль программирования, так как на него наталкивает само устройство существующих вычислительных машин, которые представляют собой гигантские конечные автоматы».

На выделение указанного подхода в качестве самостоятельного стиля программирования, на авторов работы [30] повлияла работа [6], на которую они ссылаются, как на современную методику программирования от состояний. В последней было предложено применять автоматы в программировании не от случая к случаю, как одну из моделей дискретной математики, а как универсальный подход, который целесообразно использовать практически во всех случаях, когда программа должна обладать достаточно сложным поведением, и, в особенности, реактивным [31] – реагировать по-разному на события в зависимости от состояний, в которых находится программа.

В заключение раздела отметим, что «программирование с автоматами» нельзя рассматривать как парадигму программирования, так как при этом остается неясным, как с использованием автоматов проектировать и реализовывать программы в целом.

9.1.2. Автоматное программирование как парадигма программирования

Очень многие системы, которые для внешнего наблюдателя ведут себя достаточно осмысленно, являются автоматизированными объектами.

Автоматизированный объект управления представляет собой совокупность объекта управления и системы управления, охваченных обратными связями.

Задача построения автоматизированных объектов рассматривается в любом курсе теории автоматического управления применительно к объектам различных типов.

Удивительно, что это почти никак не коснулось практики программирования, несмотря на то, что в теории алгоритмов в качестве одной из основных моделей используется машина Тьюринга, которая, по сути, является автоматизированным объектом [32], в котором объект управления – лента (ее ячейки памяти), а система управления – конечный автомат. Сложность программирования на машине Тьюринга [33] определяется тем, что ней используются чрезвычайно простые объекты управления (ячейки памяти), которые могут выполнять только простейшие действия (операции) по записи и стиранию отдельных символов. В этой ситуации вычисления, в некотором смысле, приходится выполнять конечному автомату, который для этой цели не приспособлен, так как его предназначение – управление. Другая особенность машины Тьюринга, которая может резко усложнять программы, это использование только одного автомата, которого достаточно для проведения теоретических исследований, но часто бывает мало для практического применения.

Переход от тьюрингова программирования к автоматному программированию осуществляется за счет усложнения объектов управления, которые могут выполнять сколь угодно

сложные действия (операции), и применения в системе управления системы взаимодействующих автоматов. Из изложенного следует, что универсальность предлагаемого подхода определяется тем, что он основан на расширении машины Тьюринга, которая позволяет реализовать произвольные алгоритмы.

При этом теоретические положения о том, что автоматы позволяют распознавать только регулярные языки, а магазинные автоматы – только языки с контекстно-свободными грамматиками, отходят на второй план, так как **в рассматриваемом подходе используются не автоматы, а автоматизированные объекты управления**, в которых объекты управления и их сложность не фиксированы.

Учитывая изложенное, в работе [34] была сформулирована основная особенность автоматного программирования – **программы предлагается создавать так же, как производится автоматизация технологических (и не только) процессов**.

При этом на основе анализа предметной области выделяются источники входных воздействий и автоматизированные объекты, каждый из которых, как сказано выше, представляет собой охваченные обратными связями объект управления и систему управления. Эти объекты реализуют выходные воздействия и формируют значения еще одной разновидности входных воздействий, которые по обратным связям передаются системе управления.

Объекты управления могут быть реальными или виртуальными – реализованными программно. В первом случае их логика изменена быть не может, а во втором – она, при необходимости, практически вся может быть вынесена в автоматы.

Парадигма автоматного программирования состоит в представлении и реализации программ как систем автоматизированных объектов управления [34].

В соответствии с этой парадигмой программирования, при проектировании программ предлагается выделять источники информации, систему управления и объекты управления. При этом в качестве системы управления используется система детерминированных конечных автоматов, взаимодействующих между собой, например, за счет вложенности. Автоматы, находясь в определенных состояниях, реагируют на входные воздействия и переходят в новые состояния, формируя при этом выходные воздействия, «указывающие» объектам управления, что они должны делать.

9.1.3. Основные положения автоматного программирования

Основным понятием в автоматном программировании является **состояние** [6]. При написании автоматных программ предлагается разделять состояния на два класса: **управляющие** и **вычислительные**. При этом с помощью небольшого числа управляющих состояний, как и в машине Тьюринга, можно управлять сколь угодно большим числом вычислительных состояний [35]. Во введенной классификации управляющие состояния могут быть названы **качественны-**

ми, а вычислительные – **количественными**. Поэтому в рамках автоматного программирования основное внимание уделяется управляющим состояниям, которые, если это не оговаривается особо, и рассматриваются в дальнейшем.

При этом справедливо соотношение: «Состояния + входные воздействия = конечный (детерминированный) автомат без выхода». Справедливо также: «Автомат без выхода + выходные воздействия = автомат».

Автоматы могут быть абстрактными (входные и выходные воздействия формируются последовательно) и структурными (входные и выходные воздействия формируются «параллельно»). В автоматном программировании, в отличие, например, от программирования компиляторов, обычно применяются структурные автоматы.

Время в автоматах в явном виде не используется. При необходимости применения элементов задержки, они рассматриваются как объекты управления. При этом задержки запускаются и сбрасываются из автоматов, а информация об истечении времени поступает в них в виде входных воздействий.

Автоматы могут задаваться в различном виде, однако при их проектировании и использовании человеком, они должны обладать когнитивными свойствами [36], что достигается при задании поведения автоматов графами переходов (диаграммами состояний).

Понятность графов переходов достигается во многом за счет того, что состояния в каждом из них декомпозируют множество всех входных воздействий соответствующего автомата на группы, каждая из которых определяет переходы из рассматриваемого состояния.

9.1.4. Достоинства автоматного программирования

В рамках автоматного программирования предполагается, что собственно написание (генерация) программы начинается только после ее проектирования. При этом в инженерной практике проект или его этап обязательно завершается выпуском проектной документации. Поэтому при автоматном программировании, основной областью использования которого являются встроенные системы (чисто инженерная область), должна выпускаться проектная документация [11], а не только документацию пользователя, как это обычно принято в программных проектах. При этом для автоматных программ необходимой компонентой, входящей в состав проектной документации, являются графы переходов.

Проектирование автоматов, описывающих логику программ, которая при традиционном программировании неупорядочена и потому сложна, а также формальный и изоморфный переход от автоматов к реализующим их программам, приводит к тому, что после завершения написания вручную текстов, соответствующих входным и выходным воздействиям, которые при излагаемом подходе не должны содержать сложной логики, программы либо сразу работают, либо требуют минимальной отладки.

При необходимости проведения отладки автоматные программы могут выполняться с применением отладочных протоколов [37], которые отражают поведение программ в терминах автоматов (состояний, переходов, значений входных и выходных воздействий), так как в описываемом подходе к программированию автоматы являются не картинками, как считают многие, а могут быть частью программ, представленных в нетрадиционной для многих программистов визуальной, а не текстовой форме.

При этом отметим, что проектирование автоматных программ, увеличивая время их создания, позволяет резко сократить время отладки. Это приводит к тому, что для программ средней сложности время их разработки на основе автоматного и традиционных подходов практически совпадает, но в первом случае «в остатке» остаются модели, понятные человеку, по которым программа была построена формально, а во втором – имеется только программа, понимание логики которой для представителей заказчика или даже для ее автора через некоторое время часто представляет большую проблему.

Создание программ со сложным поведением без использования автоматов приводит к трудностям на всех этапах жизненного цикла. Особенно сложно писать такие программы, так как все особенности их поведения приходится все время держать в голове вместо того, чтобы отобразить их на модели и на время забыть. Еще одним преимуществом автоматных программ является простота внесения изменений в них людьми, для которых **автоматный стиль программирования** привычен [6].

Следующее, видимо, самое важное достоинство – это эффективность верификации автоматных программ на основе использования моделей, что и отмечено в работе [38].

В заключение раздела отметим, что для автоматных программ естественен параллелизм, что особенно важно при применении многоядерных процессоров.

9.1.5. Разновидности автоматного программирования

С 1991 г. автоматное программирование развивалось в пяти направлениях: логическое управление, программирование с явным выделением состояний, объектно-ориентированное программирование с явным выделением состояний, функциональное программирование с явным выделением состояний и реализация алгоритмов дискретной математики на основе автоматного подхода.

9.1.5.1. Логическое управление

Наиболее просто и естественно применять автоматы в системах логического управления, в которых входные и выходные переменные двоичны. Естественность применения автоматов при программировании этого класса систем объясняется тем, что программы в них заменяют

схемы, проектирование которых с использованием автоматов распространено и развивается давно, начиная с релейно-контактных схем.

Однако проста и естественность применения автоматов даже для этого класса систем, видимо, далеко не очевидна, так как даже при программировании логических контроллеров, практически никто не предлагал сначала проектировать автоматы, а затем их реализовать на выбранном языке программирования.

В работе [39] было показано, что плохого в неавтоматном программировании контроллеров, а работах [40, 41] описано применение технологии автоматного программирования для систем логического управления. Эта технология была использована при создании ряда систем управления, которые в основном предназначались для управления судовыми техническими средствами [6].

В работах [42–44] в качестве примеров показано как применять предложенную технологию применительно к языкам функциональных блоков, широко используемым в программируемых логических контроллерах, а также к инструментальному средству *LabVIEW*, которое может быть применено и для программной реализации систем логического управления.

9.1.5.2. Программирование с явным выделением состояний

Значительно более широким классом по сравнению с системами логического управления являются реактивные системы. Для описания поведения этого класса систем, к которым относятся, большинство встроенных систем, применение автоматов также естественно, как и для систем логического управления. Однако далеко не все программисты используют автоматы для таких систем, приобретая множество проблем, о которых говорилось выше.

Реактивные системы являются более сложными по сравнению с системами логического управления. В них:

- в качестве входных воздействий, наряду с входными переменными, используются события;
- запуск программ осуществляется по событиям, а не циклически;
- в качестве выходных воздействий могут использоваться не только двоичные, а любые функции, что позволяет называть автоматы в этом случае **гибридными**;
- автоматы могут содержать не только вложенные состояния [31], но и вложенные автоматы;
- автоматы могут взаимодействовать не только за счет проверки номеров состояний, как было предложено для систем логического управления [6], но и путем вызываемости и вложенности.

Рассматриваемый класс систем обычно реализуется на процедурных языках программирования, и поэтому традиционно используемый процесс написания программ в этом случае

может быть назван **процедурным программированием** или просто **программированием**. В таких программах состояния существуют, но обычно явно не выделяются, что отличает их от автоматных программ, создание которых в этом случае может быть названо **программированием с явным выделением состояний**.

Технология программирования с явном выделением состояний создавалась в ходе выполнения работ по разработке системы управления судовыми дизель-генераторами [45]. Фрагмент проектной программной документации приведен в работе [46]. Эта технология подробно описана в работах [47–51].

9.1.5.3. Объектно-ориентированное программирование с явным выделением состояний

Уже два десятилетия объектно-ориентированное программирование является наиболее широко используемым стилем программирования в мире. При применении объектной парадигмы программы строят из объектов, взаимодействующих за счет обмена событиями.

В рамках этой парадигмы ничего конструктивного относительно описания и реализации поведения объектов не было сказано. Более того, с развитием методов проектирования таких программ в этот вопрос ясность не была внесена, а применять автоматы предлагалось лишь от случая к случаю, а не «как руководство» к действию, пригодное для использования во многих системах при описании их сложного поведения. Даже в тех редких случаях, когда автоматы применялись для описания поведения программ, это делалось крайне неубедительно.

Появление унифицированного языка моделирования *UML*, и даже языка *UML 2.0*, эту проблему не решило, так как, во-первых, в этом языке, кроме диаграмм состояний для описания поведения предлагается использовать и другие типы диаграмм и не говорится когда и какие диаграммы следует применять, во-вторых, в рамках унифицированного процесса разработки программ *RUP*, как впрочем, и при использовании других методологий их создания, например, экстремального программирования, не было предложено подходов для совместного использования диаграмм, описывающих статические и динамические свойства программ, и, наконец, в-третьих, диаграммы для описания поведения в основном использовались как язык общения между участниками разработки и для документирования программ, в то время как для кодогенерации использовались только диаграммы классов. Лишь в последние годы в рамках концепции *исполняемый UML* вопрос о кодогенерации был поставлен шире.

Для решения указанной проблемы СПбГУ ИТМО были выполнены исследования по совместному использованию объектной и автоматной парадигм программирования. При этом такой стиль программирования был назван **объектно-ориентированное программирование с явным выделением состояний** [52].

Возможны различные подходы к решению этой проблемы. Автоматы можно использовать, например, как методы классов [53, 54], или как классы [55, 56].

Более «глубокое проникновение» автоматов в объектно-ориентированное программирование происходит при применении паттернов проектирования. Для устранения недостатков, присущих известному паттерну *State*, в работе [57] был предложен паттерн *State Machine*, на базе которого создан язык с тем же названием [58], являющийся расширением языка *Java* и позволяющий достаточно эффективно реализовывать автоматы.

В работе [59] был предложен еще один подход к реализации объектно-ориентированных программ с явным выделением состояний, который позволяет обеспечить повторное использование программных компонентов, параллельные вычисления, автоматическое протоколирование работы системы и повышение ее отказоустойчивости. В качестве основы для разработки «автоматной» части программ в этой работе была предложена библиотека, реализованная на языке *C++*. При использовании этой библиотеки остальная часть программ (контекст) разрабатывается традиционным образом.

Особенности проектирования и свойства автоматных программ позволяют отнести автоматное программирование к одной из разновидностей синхронного программирования [60], которое активно развивается в Западной Европе для создания программного обеспечения ответственных систем.

Высокое качество автоматных программ может быть достигнуто не только за счет автоматного расширения универсальных языков программирования, но в тех случаях, когда для написания автоматных программ разрабатываются языки, ориентированные на эту предметную область [61]. Одной из наиболее интересных в этом направлении является среда языково-ориентированного программирования *Meta Programming Systems (MPS)*, разрабатываемая компанией *JetBrains* (Санкт-Петербург) [62]. Эта среда позволяет эффективно создавать проблемно-ориентированные языки, которые строятся на базе задания абстрактного синтаксического дерева.

В ходе работ по этой тематике, выполняемых на кафедре КТ, были разработаны текстовый язык автоматного программирования [63] и язык и инструментальное средство для многопоточного автоматного программирования [64].

Также на кафедре разработан метод интеграции кода и формализованных требований к автоматным объектно-ориентированным программам [65, 66], а также метод внесения изменений в автоматные программы на основе рефакторинга [67].

Существуют и другие подходы к совместному использованию объектной и автоматной парадигм программирования. Классификация таких подходов приведена в работе [68].

Исследования в указанном направлении продолжаются. При этом, например, в работе [69] предложена удобная графическая нотация для отображения наследования автоматных классов, а в работе [70] рассмотрен декларативный подход к вложению и наследованию автоматных классов при использовании императивных языков программирования.

9.1.5.4. Функциональное программирование с явным выделением состояний

В работах [71–74] рассмотрены вопросы реализации конечных автоматов на функциональных языках программирования.

9.1.5.5. Применение автоматов при реализации вычислительных алгоритмов дискретной математики

Среди алгоритмов дискретной математики известно лишь несколько алгоритмов, которые могут быть эффективно специфицированы с использованием автоматов, например, поиск подстрок, обход деревьев [75] или подсчет длины слов в строке [76].

Поэтому для автоматного представления итеративных [77, 78] и рекурсивных [79] алгоритмов на кафедре были разработаны соответствующие методы. Был также разработан метод преобразования программы в систему взаимодействующих автоматов [80].

9.1.6. Разработка визуализаторов алгоритмов дискретной математики

Технология автоматного программирования продемонстрировала свою эффективность при решении различных задач, но, как отмечено в предыдущем разделе, при реализации алгоритмов дискретной математики автоматы используются крайне редко.

Однако оказалось, что если реализовывать алгоритмы дискретной математики на автоматах часто нецелесообразно, то строить их визуализаторы с применением автоматов, как уже отмечалось в разд. 5, крайне полезно всегда, так как при этом визуализацию можно проводить, например, в состояниях, а не там, где захочется разработчику. При решении задачи построения таких визуализаторов был выполнен ряд работ как теоретического [7], так и прикладного характера [8, 9]. В результате в СПбГУ ИТМО было разработано инструментальное средство *Vizi* для автоматизированного построения визуализаторов рассматриваемого класса [10]. Несмотря на то, что лекции по курсу «Алгоритмы дискретной математики» читаются практически в каждом университете мира, где обучают информационным технологиям, а в некоторых из них в учебном процессе используются визуализаторы, эффективную методику их построения без применения автоматного подхода создать не удавалось никому.

9.1.7. Инструментальные средства для поддержки автоматного программирования

Рассмотрим инструментальные средства для поддержки автоматного программирования. Для **процедурных автоматных программ** в работе [81] было описано инструментальное средство, которое по графам переходов, представленным в нотации, предложенной в работе [48], и изображенным с помощью пакета *Visio*, генерирует код, изоморфный реализуемым графам переходов, который основан на конструкциях `switch` языка *C*.

Это средство используется в настоящее время при создании программного обеспечения одного класса ответственных систем реального времени [82]. При этом в качестве программ применяются реализованные вручную на языке C функции входных и выходных воздействий, которые практически не содержат логики, а также графы переходов, по которым исходный код генерируется автоматически. По отзывам разработчиков этих систем, их уже долгое время не покидает удивление оттого, что в каждой новой системе программы, спроектированные указанным образом, работают практически без отладки, а расширение функциональности обычно обеспечивается «малой кровью», и все это достигается при использовании процедурного, а не объектно-ориентированного программирования.

Еще одно инструментальное средство этого типа было разработано восьмикласником лица «Вторая школа» Леонидом Столяровым [83].

Развитие это направление исследований получило в работе [84], в которой показано, что аналогичный подход может быть использован для реализации автоматов на любом наперед заданном языке программирования, и было создано инструментальное средство *MetaAuto* для поддержки такого подхода.

Переходя к инструментальному средству для поддержки построения **объектно-ориентированных автоматных программ**, отметим, что если для генерации программ по автоматам, кроме средств, рассмотренных выше, известны также и многие другие, то решение задачи об автоматизации построения объектно-ориентированных программ в целом в открытых источниках не излагалось.

Решение этой задачи было предложено в СПбГУ ИТМО в ходе проведения опытно-конструкторской работы по теме «Автоматное программирование: применение и инструментальные средства», которая выполнялась в рамках Федеральной целевой научно-технической программы «Исследования и разработки по приоритетным направлениям науки и техники» на 2002 – 2006 гг. [85]. Это средство, названное *UniMod* [86–91], использует нотацию *UML* и является плагином к среде разработки *Eclipse*. Эта работа вошла в число 15 наиболее перспективных научных проектов, которые находились в 2005 и 2006 гг. в распоряжении Федерального агентства по науке и инновациям.

При использовании инструментального средства *UniMod* структура программ задается диаграммами классов, которые изображаются не традиционным путем, а в форме схемы связей автоматов с поставщиками событий и объектами управления. Динамика программ описывается с помощью диаграмм состояний в нотации языка *UML*, в которых могут использоваться не только вложенные состояния, но и вложенные автоматы. При этом имеется возможность проверить ряд свойств этих диаграмм, например, полноту и непротиворечивость. Функции входных и выходных воздействий, которые практически не содержат логики, пишутся на языке *Java* вручную. После этого автоматически может быть скомпилирован код программы в целом или про-

грамма может выполняться в режиме интерпретации. Описанное инструментальное средство находится в свободном доступе, и является открытым и бесплатным средством, поддерживающим концепцию *исполняемого UML*. Пример использования указанного инструментального средства описан в работе [92]. Другие примеры – курсовые работы студентов третьего курса опубликованы по адресу <http://is.ifmo.ru/unimod-projects/>.

9.1.8. Технологии автоматного программирования

На основании указанных выше работ была разработана технология автоматного программирования, которая охватывает все этапы жизненного цикла программ рассматриваемого класса. Она описана в работе [93], а в работе [94] изложена для массового читателя.

В соответствии с этой технологией при проектировании программы создается модель из двух **типов** диаграмм: схемы связей (при объектно-ориентированном программировании – в виде диаграммы классов) и графа переходов. Они полностью описывают статические и динамические свойства программы. Возможность применения только двух типов диаграмм отличает автоматный подход от других подходов к *проектированию объектно-ориентированных программ*. В рамках предлагаемого подхода иногда предлагается строить третий тип диаграмм – диаграмму объектов.

Генерация кода по модели может производиться как вручную, так и автоматически – с помощью одного из инструментальных средств. При генерации код по модели строится формально и сохраняет ее структуру. Функции входных и выходных воздействий, вызываемых из модели, обычно реализуются вручную. Эти функции весьма просты, так как вся сложная логика реализуется кодом, сгенерированным по модели. Таким образом, в отличие от традиционного подхода, в описываемой технологии этап кодирования занимает сравнительно мало времени даже при его выполнении вручную и обычно не приводит к появлению трудно обнаруживаемых ошибок в программе. Ввиду качественного проектирования, получаемые программы во многих случаях практически не требуют отладки. При ее необходимости, отладка упрощается за счет анализа протоколов работы программы, которые ведутся в автоматных терминах.

Разработанные диаграммы составляют основную часть проектной документации. Схемы связей и графы переходов дополняются подробными комментариями и описанием их работы, а также обоснованием принятых решений. Разработаны требования к проектной документации (<http://is.ifmo.ru/projects/req/>).

Описанная технология обладает следующими особенностями: предполагается, что вся логика централизована и реализуется автоматами; автоматные программы с так построенной логикой, хорошо приспособлены для верификации с помощью метода *Model Checking*; удобна при разработке процедурных программ; удобна при разработке объектно-ориентированных программ одним-двумя программистами.

Если объектно-ориентированная программа разрабатывается командой программистов, то в главе 3 работы [15] предлагается другая технология автоматного программирования. При ее использовании, как обычно, выделяются классы. Если класс может быть представлен в виде автоматизированного объекта управления, система управления которым реализуется одним автоматом, то он так и представляется. Вложенность автоматов обеспечивается средствами объектно-ориентированного программирования. Пример применения такой технологии описан в работе [95]. Эта технология более естественна для программистов, которые пишут объектно-ориентированные программы традиционным образом, но верификация программ с такой структурой на основе метода *Model Checking* усложняется по сравнению с программами, в которых логика централизована, как это имеет место при использовании технологии, описанной первой.

9.1.9. Апробация технологии автоматного программирования

Апробация первой технологии автоматного программирования осуществлялась при разработке ответственных систем и показала свою эффективность [96–99].

Автоматный подход применялся также при программировании игр [100, 101], информационных систем [102], построении обработчиков XML-документов [103] и использовании FLASH-технологии [104].

Автоматы использовались и при выполнении исследований в области искусственного интеллекта. Так, например, в работах [105, 106] рассматривалось применение автоматов при построении мультиагентных систем, а в работах [107, 108] – их применение при моделировании функционирования агрегатов жидкостного ракетного двигателя.

9.2. Автоматное управление

Автоматное программирование является весьма важной при создании ответственных систем парадигмой программирования [109]. Однако не менее важным для таких систем является автоматное управление – применение автоматов в управлении. Этот термин был введен на стр. 27 работы [6]. Автоматное управление важно для различных объектов, но особенно – при создании мобильных роботов [110–114]. При этом очень важно мнение аспиранта А.А. Шальито Виталия Клебана по поводу автоматного программирования, которое он высказал после сдачи одной из систем: «В системе часть программ была написано традиционно, а часть – автоматно. Про программы первого типа можно сказать, что они работают почти непредсказуемо – то работают, то не работают. Автоматные программы или работают, или не работают. И если они начали работать, то они так работать будут всегда. Это мечта всех программистов – поведение программы должно быть предсказуемо».

Термин «автоматное управление» начинает применяться все шире, например, он был использован в названии выигранного при участии кафедры КТ гранта «Адаптивное и автоматное управление мобильными роботами», выполняемого в рамках Аналитической ведомственной целевой программе «Развитие научного потенциала высшей школы».

Автоматное управление может быть реализовано не только программно, но и аппаратно, причем в настоящее время аппаратная реализация тоже может быть программируемой при использовании программируемых логических интегральных схем (ПЛИС). В работе [115] было предложено применять автоматное управление для шаговых двигателей, причем устройство управления реализовывалось на ПЛИС. Эффект внедрения превзошел все ожидания: устройство практически не потребовало настройки и сразу работало, и в него легко вносить изменения.

Это открывает возможность ко-дизайна, при котором и в программной, и программируемой аппаратной частях системы применяется автоматное управление.

9.3. Верификация автоматных программ

Автоматные программы целесообразно использовать при создании ответственных систем, так как они не только легко документируются, но и **приспособлены к верификации**. Поэтому дальнейшие исследования в области автоматного программирования на кафедре КТ проводились в направлении **верификации автоматных программ на основе метода *Model Checking*** [116–122]. Эти работы выполнялись в рамках Федеральной целевой программы «Исследования и разработки по приоритетным направлениям развития научно-технологического комплекса России на 2007–2012 гг.» по теме «Разработка технологии верификации управляющих программ со сложным поведением, построенных на основе автоматного подхода».

Выбор этого направления исследования связан с тем, что в настоящее время метод верификации модели (*Model Checking*) является наиболее практическим методом верификации программ. В этом методе по программе строится модель с конечным числом состояний, и проверяемые свойства записываются в виде формул темпоральной логики. В результате верификации либо подтверждается соответствие программы формуле, либо приводится контрпример.

Для программ, построенных традиционным образом, верификация моделей связана с большими временными и интеллектуальными затратами, так как модель обычно содержит большое число состояний и не может быть построена по тексту программы формально, так как в традиционных программах обычно не выделяются управляющие состояния, которых обычно немного. Эти недостатки устраняются в автоматных программах, в которых при их проектировании состояния делятся на управляющие и вычислительные, что позволяет производить по спецификации, задаваемой в виде системы графов переходов, **автоматическое** построение мо-

дели для верификации. Для автоматных программ возможен и **автоматический** обратный переход от модели к спецификации при построении контрпримера.

В ходе работ по теме разработаны:

1. Методы верификации автоматных моделей управляющих программ, позволяющие автоматизировать построение моделей Крипке по автоматным моделям.
2. Инструментальные средства, построенные на основе разработанных методов.
3. Технология верификации автоматных моделей управляющих программ, основанная на разработанных методах.

Внедрение разработанной технологии позволяет производить верификацию управляющих программ со сложным поведением. Это приводит к повышению качества такого рода программ при сокращении времени их разработки. В отличие от широко применяемого метода экспертной оценки соответствия программ спецификации, разработанные методы позволяют значительно уменьшить влияние человеческого фактора, что особенно важно для ответственных систем [123].

9. 4. Генерация автоматов на основе методов искусственного интеллекта

В рамках той же Федеральной целевой программы («Исследования и разработки по приоритетным направлениям развития научно-технологического комплекса России на 2007–2012 гг.») на кафедре КТ были выполнены работы по теме «Технология генетического программирования для генерации автоматов управления системами со сложным поведением».

Выполнение исследований по этой теме связано с тем, что существуют задачи, которые могут быть решены за счет построения конечных автоматов или их систем. Однако, во многих случаях их ручное построение либо затруднено, либо невозможно. Для построения таких автоматов могут быть использованы методы генетического программирования. Решения этой задачи особенно актуально в рамках автоматного программирования, так как ядро таких программ составляют автоматы. Поэтому целью этой работы была разработка методов генетического программирования для генерации управляющих автоматов систем со сложным поведением.

При этом были разработаны:

1. Методы эффективного представления автоматов в виде хромосом, а также соответствующие операции скрещивания и мутаций.
2. Инструментальные средства, построенные на основе разработанных методов.
3. Технология генетического программирования для генерации автоматов управления системами со сложным поведением.

Разработанные методы и технология на их основе позволяют существенно сократить время создания автоматных программ, управляющих системами со сложным поведением за

счет автоматизированной генерации требуемых автоматов по выбранной функции приспособленности. При этом время разработки автоматов сокращается в десятки раз, а качество получаемого решения сравнимо, а часто и выше решения, предлагаемого человеком. Изложенный подход позволяет резко повысить уровень автоматизация автоматных программ, так как для них поведенческая часть может достигать до 80 % кода, который, в свою очередь, может строиться автоматически, так как, во-первых, при использовании генетического программирования сами автоматы могут генерироваться автоматически, а, во-вторых, по автоматам автоматически может быть построен код.

При этом были предложены три метода представления автоматов в виде хромосом для генетического программирования: на основе сокращенных таблиц [124, 125], деревьев решений и линейных бинарных графов [126 – 128].

Эффективность метода сокращенных таблиц была продемонстрирована на примере генерации автомата для автопилота верхнего уровня модели самолета. При этом автомат генерировался с помощью алгоритма генетического программирования, использующего вычисление функции приспособленности на основе моделирования поведения самолета в некоторой внешней среде [124].

Основной недостаток этого подхода – функцию приспособленности необходимо для каждой задачи разрабатывать «с нуля». Кроме того, подход к вычислению функции приспособленности на основе моделирования связан с большими затратами вычислительных ресурсов.

Этих недостатков лишен метод генерации автоматов с помощью генетического программирования на основе тестов [129, 130]. Однако такой метод не всегда генерирует автомат, удовлетворяющий заданной спецификации. Для устранения этого недостатка в работе [131] было предложено в ходе реализации алгоритма генетического программирования проводить верификацию.

Дальнейшее развитие метода генетического программирования на основе тестов было выполнено в работе [132], в которой в ходе его реализации выполнялось сравнение поведения автоматов с поведением, обеспечиваемым за счет управления человеком-оператором. Особенность рассматриваемого подхода состоит в том, что он позволяет использовать объекты управления не только с дискретными, но и с непрерывными параметрами. Применение метода было проиллюстрировано на примере создания автомата, управляющего моделью самолета в режиме «мертвая петля».

Еще один метод, позволяющий осуществлять генерацию автоматов с помощью генетического программирования для случая, когда значения входных переменных могут быть вещественными числами, предложен в работе [133]. Этот метод основан на том, что система управления строится как совокупность искусственной нейронной сети и автомата. Нейронная сеть преобразует значения входных вещественных переменных в логические, которые подаются на

вход автомата. Применение метода иллюстрируется на примере моделирования командного соревнования беспилотных летательных аппаратов. При этом для учета взаимодействия в команде хромосома алгоритма генетического программирования описывает системы управления двух аппаратов.

На основе разработанных подходов и методов был решен ряд задач по генерации автоматов, описанных в работах [134–136].

Для обучения методам генерации автоматов на кафедре КТ были разработаны виртуальные лаборатории [138–140], которые используются в учебном процессе.

Автоматы могут строиться и с помощью других формализованных методов. Например, в работе [141] было предложено использовать **динамическое программирование для построения автоматов**. Однако такой подход имеет ограниченное использование.

Естественно, что не все задачи, которые эффективно решаются с помощью генетических алгоритмов связаны с генерацией автоматов, так, например, в работе [142] генетические алгоритмы применялись для определения неэффективных решений олимпиадных задач по программированию. Существуют задачи, которые могут на основе генетических алгоритмов решаться как с помощью генерации автоматов, так и без их использования. Решение одной из таких задач описано в работе [143].

Вопросы применения генетических алгоритмов для генерации тестов для автоматных программ рассмотрены в работе [144].

9. 5. Клеточные автоматы

Первоначально были написаны две научно-популярные работы по этой тематике [145, 146], вторая из которых весьма интересна. После этого была выполнена работа [147], которая попала в библиотеку С. Вольфрама по клеточным автоматам.

Инструментальное средство для создания клеточных автоматов описано в работе [148], в работе [149] клеточные автоматы используются для распознавания текстов, а в работе [150] для генерации клеточных автоматов применяются генетические алгоритмы.

9. 6. Другие работы в области программирования и не только

В первые годы существования кафедры основные научные исследования студентов были связаны с работами в области физики и оптики, проводимых под руководством профессоров С.А. Козлова и И.П. Гурова. При этом студенты кафедры КТ выигрывали значительную долю грантов таких оптических организаций, как *OSA* и *SPIE*.

Кроме этих направлений исследований в годы становления кафедры ее студентами проводились серьезные исследования в области математики. При этом, частности, в 1997 г. под ре-

дакцией В.Н. Васильева был издан на английском языке сборник трудов студентов кафедры КТ [151].

В дальнейшем на кафедре стали проводиться исследования в области программирования, что соответствует профилю кафедры КТ. При этом исследования в этой области можно разбить на два класса: *Software Engineering* и *Computer Science*.

К первому классу могут быть отнесены исследования по автоматному программированию, указанные выше. Результаты этих исследований нашли свое отражение в первом в мире сборнике статей по автоматному программированию [152], а результаты дальнейших исследований по технологиям программирования и искусственного интеллекта, выполненных на кафедре КТ – в сборнике [153]. При этом отметим, что статья «Автоматное программирование» стала одним из победителей Всероссийского конкурса обзорно-аналитических статей по приоритетному направлению «Информационно-телекоммуникационные системы (<http://www.ict.edu.ru/itkonkurs2008/2224/>).

К этому же классу относятся и следующие работы, как относящиеся к автоматной тематике [154], так и не относящиеся к ней [155–157].

Ко второму классу относятся как работы, связанные с автоматами [158–160], так не связанные с ними [161].

С работами, указанными в настоящей работе, а также с многими другими работами можно ознакомиться на сайте <http://is.ifmo.ru/>.

9. 7. «Расшифровка» генома

В настоящее время отсутствуют российские программы, обеспечивающие «расшифровку» генома. В результате общения руководителя центра «Биоинженерия» РАН академика К.Г. Скрябина и заведующего кафедрой «Технологии программирования» профессора А.А. Шалыто была достигнута договоренность о проведении под их руководством совместных работ по созданию комплекса программ для решения этой задачи.

«Расшифровка» генома состоит из следующих этапов: секвенирование молекул ДНК, сборка генома, анализ и сравнение геномов.

Для *секвенирования молекул ДНК* применяются специальные приборы – секвенаторы. В настоящее время наиболее «продвинутыми» являются секвенаторы, производимые компанией *Illumina* (США). Результатом секвенирования являются прочитанные «кусочки» ДНК относительного небольшого размера (порядка 100–150 нуклеотидов). При этом вся последовательность ДНК покрыта этими кусочками несколько десятков раз. В России секвенаторы последнего поколения имеются в центре «Биоинженерии» и в МГУ.

Задачей *сборки генома* является восстановление всей последовательности ДНК (ее длина составляет от миллионов до миллиардов нуклеотидов у разных живых существ) на основании информации, полученной в результате секвенирования. Решение этой задачи можно сравнить со сборкой головоломки-пазла, состоящей из миллиарда фрагментов. Для сборки генома в мире разрабатываются специальные алгоритмы, которые запускаются на суперкомпьютерах или кластерах.

Задачей *анализа и сравнения геномов* является выявление различий и сходств как между геномами живых существ одного вида, так и между различными видами. Этот этап, как и второй, требует разработки специальных алгоритмов, оперирующих с очень большими объемами данных.

Задачей сотрудничества кафедры КТ и центра «Биоинженерия» является разработка технологии сборки генома, превосходящей по качеству или производительности (в идеале – по обоим критериям) мировой уровень. Эта технология будет включать в себя как алгоритмы сборки генома, так и реализующее их программное обеспечение для кластеров или суперкомпьютеров.

Работа проводится в два этапа. На первом из них были написаны магистерская диссертация [162] и две бакалаврские работы [163, 164] по этой тематике, что позволило осуществить быстрый «вход в геном» [165], что весьма необычно. Ответственным исполнителем этих работ на кафедре был Г.А. Корнеев.

На втором этапе на кафедре КТ разрабатывается алгоритм сборки генома, состоящий из четырех этапов:

- исправление ошибок в *ридах* – данных, поступивших с секвенирующей машины;
- восстановление фрагментов геномной последовательности длиной примерно по 500 нуклеотидов на основе исправленных ридов;
- сборка *контигов* – длинных непрерывных фрагментов геномной последовательности. Для этого этапа пока использовался сборщик *Newbler*, предоставленный центром «Биоинженерия»;
- определение взаимного расположения контигов друг относительно друга – для этой цели предполагается использовать один из модулей open-source сборщика *Abyss*.

К середине января 2011 г. был разработан и реализован алгоритм исправления ошибок в ридов. Запуск программы, реализующей этот алгоритм, проводился на суперкомпьютере (24 гигабайта оперативной памяти, два четырехъядерных процессора) центра «Биоинженерия», объем входных данных составлял порядка 250 гигабайт. В результате трех запусков программы, занявших суммарно примерно 60 часов, было исправлено примерно 18,5% исходных ридов, и появилась возможность перейти ко второму этапу.

К началу февраля 2011 г. был разработан и реализован алгоритм восстановления фрагментов геномной последовательности, основанный на построении и обходе графа де Брюина. Запуск программы, реализующей этот алгоритм, проводился на суперкомпьютере, предоставленном компанией «Т-Платформы» (64 гигабайта оперативной памяти, два двенадцатиядерных процессора). В результате работы программы в течение 40 часов были сформированы данные, которые были поданы на вход сборщика *Newbler*. Работа этого сборщика на этих данных заняла 60 часов.

Полученные результаты показывают работоспособность и перспективность разработанного метода сборки генома. Ответственными исполнителями проекта являются: от кафедры КТ – Ф.Н. Царев, от центра «Биоинженерия» – Е.Б. Прохорчук. Исполнители от кафедры КТ – А.В. Александров, В.В. Исенбаев, С.В. Казаков, С.В. Мельников, А.А. Сергушичев.

В апреле 2011 г. наша команда из пяти человек (трое из них ездили за счет нашего университета) приняла участие в семинаре в Барселоне, на котором подводились предварительные итоги выполнения проекта *de novo genome assembly project (dnGASP)*, организованного национальным центром геномного анализа (Барселона, Испания). В работе семинара участвовали девять исследовательских центров, таких как, например, *Beijing Genomics Institute* (Пекинский геномный институт, Китай), *European Bioinformatics Institute* (Европейский институт биоинформатики, Великобритания) и *Canada's Michel Smith Genome Science Center* (Канадский институт геномных исследований, Канада). Сборкой генома, кроме организаций, принявших участие в указанном проекте, занимаются еще несколько американских университетов.

Наше участие в этой проекте (<http://cnag.bsc.es/>) позволяет говорить о том, что **СПбГУ ИТМО является одним из немногих университетов и исследовательских центров мира, обладающих технологией сборки геномных последовательностей на основе данных о чтениях на секвенаторах второго поколения.**

Так как срок подачи результатов сборки генома в Барселону был 01.03.2011 г., то у наших ребят было всего полтора – два месяца на проведение сборки, включая разработку новых алгоритмов и программ. Поэтому есть все основания полагать, что на следующей стадии проекта разрабатываемая нами технология сборки генома значительно улучшится, так как работы в этом направлении нашей командой активно продолжаются.

В ходе работ по сборке генома осуществляется эксперимент по созданию **новой разновидности экстремального программирования – программирование олимпиадными командами**, при котором одна или несколько команд, успешно выступающих на чемпионате мира по программированию, на «тренировках» и после них совместно с тренерами и представителями предметной области **решают и обсуждают после многочасового мозгового штурма решения не многих задач, как принято на олимпиадах, а только одной**, которая является научной. Есть основания надеяться, что многолетние тренировки, командный дух и наличие у

участников «одной крови», могут позволить более эффективно решать задачи, основанные на знании алгоритмов дискретной математики, чем при традиционной форме работы.

Результаты первых двух месяцев проведения эксперимента, в которые входили все новогодние праздники и сессия, свидетельствует об эффективности предложенного подхода по решению сложных задач указанного класса.

10. Другие достижения кафедры КТ и ее выпускников

Выпускники кафедры добиваются выдающихся результатов не только в олимпиадном, но и в промышленном программировании. О всех не написать. Вот двое из таких выпускников:

- **Роман Елизаров** (1977 г.) – лауреат премии Президента Российской Федерации 2003 г. в области образования, выпускник кафедры КТ 2000 г., участник финалов чемпионатов мира по программированию среди студентов 1996, 1997 гг., третье место на всемирной олимпиаде по информатике среди школьников 1994 г.
- **Матвей Казаков** (1979 г.) – лауреат премии Правительства РФ 2008 г. в области образования, выпускник кафедры КТ 2002 г., призер чемпионата мира по программированию 1999 г.

Теперь об одном из достижений, относящихся к области инноваций. Под руководством чемпионов России 2001 г. и призеров чемпионата мира по программированию 2003 г. **Александра Штучкина, Евгения Южакова и Тимофея Бородина** за десять месяцев в компании «Скартел» было создано программное обеспечение для сотового телефона четвертого поколения (торговая марка «Yota»). За это достижение в 2009 г. А. Штучкин, Е. Южаков и Ф. Царев получили **Гран-при «Прорыв» Года молодежи в России**, который им вручил в спорткомплексе «Олимпийский» Президент РФ Д.А. Медведев.

Многие одаренные школьники, попав в созданные на кафедре КТ условия «для соударения умов», сохраняют свой научный потенциал и после ее окончания. При этом выпускниками кафедры защищено более 20 кандидатских диссертаций, и не только по технологиям программирования, по которым защитились **М.А. Казаков, В.С. Гуров, Г.А. Корнеев, Е.Г. Князев, П.Г. Лобанов, О.Г. Степанов и М.А. Мазин**. По этой же тематике защитились аспиранты кафедры призер чемпионата мира по программированию **Н.Н. Шамгунов**, окончивший Уральский государственный университет им. А.М. Горького, и **Д.Г. Шопырин**, окончивший Оренбургский государственный университет.

Выпускники кафедры КТ сравнительно быстро взяли и следующий рубеж, правда, пока не в области технологий программирования, а в математике и в физике – **стали докторами физико-математических наук**: в 2008 г. – **Е.О. Степанов** (первый выпускник кафедры КТ, закончивший кафедру по индивидуальному плану в 1994 г.), а в 2010 г. – **Павел Белов и Юрий**

Шполянский, закончившие кафедру в 2000 г.! До этого Павлу Белову была присуждена **премия Президента РФ** для молодых ученых по науке и инновациям за 2009 г.

Число грантов и стипендий разного уровня (включая гранты и стипендии Президента и Правительства РФ), которые за все годы получили студенты и аспиранты кафедры КТ достигает многих десятков и определить их точно уже трудно.

Имея мощный студенческий и аспирантский научный потенциал, кафедра КТ активно участвует в различных конкурсах на получение грантов и государственных контрактов, часто побеждает в них, а потом их успешно выполняет.

Кроме перечисленных выше грантов, сотрудники кафедры КТ выиграли также четыре гранта по Федеральной целевой программе «Научные и научно-педагогические кадры инновационной России» на 2009–2013 гг. на проведение научных исследований:

- научными группами под руководством докторов наук – тема «Применение методов искусственного интеллекта в разработке управляющих программных систем»;
- научными группами под руководством кандидатов наук – тема «Методы повышения качества при разработке автоматных программ с использованием функциональных и объектно-ориентированных языков программирования»;
- молодыми кандидатами наук – тема «Разработка методов совместного применения генетического и автоматного программирования для построения систем управления беспилотными летательными объектами»;
- целевыми аспирантами – «Разработка методов машинного обучения на основе генетических алгоритмов для построения управляющих конечных автоматов».

Кроме того, при участии кафедры КТ, как отмечено выше, был выигран также грант «Адаптивное и автоматное управление мобильными роботами» в рамках «Аналитической ведомственной целевой программы «Развитие научного потенциала высшей школы».

Сотрудники кафедры КТ неоднократно выигрывали гранты Российского фонда фундаментальных исследований. Например, в настоящее время выполняется работа по теме «Разработка методов машинного обучения на основе генетического программирования для построения управляющих конечных автоматов».

В апреле 2011 г. в рамках Федеральной целевой программы «Научные и научно-педагогические кадры инновационной России» на 2009 – 2013 годы мы выиграли два конкурса на заключение на государственных контрактов на проведение исследований научными группами под руководством докторов наук. Первый из них в области «Информатика» по теме **«Разработка метода машинного обучения на основе алгоритмов решения задачи о выполнимости булевой формулы для построения управляющих конечных автоматов»** (<http://kadryedu.ru/concurs.php?file=0a31d87241b1507641978519f8cb5878>), а второй – в следую-

щих областях: биокаталитические, биосинтетические и биосенсорные технологии; биомедицинские и ветеринарные технологии жизнеобеспечения и защиты человека и животных; геномные и постгеномные технологии создания лекарственных средств; клеточные технологии; биоинженерия; биоинформационные технологии (!) по теме «Разработка метода сборки геномных последовательностей на основе восстановления фрагментов по парным чтениям» (<http://www.kadryedu.ru/concurs.php?file=d3a5b8151d8bb37a35978c170bdeacbd>).

Первом случае контрактов было три, а заявок – 73 (по одной от организации), а во втором – контрактов шесть, а заявок – 103, причем и там и там **мы выиграли первое место!**

11. Результаты и эффективность работы кафедры КТ

Результаты и эффективность многолетней работы описанной системы, созданной на кафедре КТ, по поиску школьников, одаренных в области информатики и программирования, и подготовке высококвалифицированных специалистов в области производства ПО [166, 167] можно охарактеризовать следующим образом.

1. Разработана система «школа – вуз – научные исследования – индустрия», обеспечивающая поиск, профориентацию, отбор школьников, одаренных в области информатики и программирования, формирование за счет дополнительного обучения, предпрофессиональной и профессиональной подготовки студентов (включая обучение в аспирантуре и докторантуре) высококвалифицированных специалистов в области производства ПО, которые могут выполнять функции разработчиков, исследователей и руководителей широкого круга научно-технических, научно-исследовательских и инновационных проектов.
2. Анализ профессиональных карьер выпускников всех выпусков, которые проходили подготовку в рамках описанной системы, показал, что все они работают по специальности. При этом примерно три четверти из них спустя два-три года после окончания вуза становятся руководителями проектов в области производства ПО, и даже руководителями предприятий в этой области.
3. Создан получивший всероссийское и международное признание центр подготовки высококвалифицированных программистов. Выпускники и студенты кафедры разработали методическое и технологическое обеспечение и поддерживают организационную работу по проведению центром Всероссийских студенческих и школьных олимпиад, российских интернет-олимпиад по информатике и программированию, а также полуфинальные и четвертьфинальные соревнования Северо-Восточного Европейского региона чемпионата мира по программированию, начиная с 1996 г. и по настоящее время.
4. Обоснована целесообразность совместного использования в данной системе проектного и соревновательного подходов, обеспечивающих формирование у обучающихся дополняю-

щих друг друга набора качеств, необходимых для разработки инновационных программных проектов.

5. Разработаны и реализованы концепция и подход к сквозному непрерывному проектному обучению, которые носят оригинальный характер и соответствуют, а по многим параметрам и превосходят уровень лучших мировых достижений в этой области. Концепция и подход базируются на специально разработанных компонентах: технологии автоматного программирования, инструментальных средствах, интернет-библиотеках проектов, виртуальных лабораториях и едином методическом подходе.
6. В части развития соревновательного подхода проведенные работы, которые были начаты в СПбГУ ИТМО одними из первых в стране и в мире, получили широкую известность и все-российское и международное признание. Сегодня СПбГУ ИТМО является общепризнанным российским и мировым лидером в этой области. В ходе совершенствования соревновательного подхода были разработаны и реализованы принципы и формы его использования при поиске, отборе и обучении школьников и студентов, одаренных в области информатики и программирования, базирующиеся на автоматическом тестировании программных решений и соответствующем учебно-методическом и аппаратно-программном обеспечении.
7. Проведена многолетняя апробация разработанной системы на кафедре, которая показала свою эффективность при решении задачи поиска и подготовки разработчиков, исследователей и руководителей компаний и инновационных проектов в области производства ПО. Эта система, базирующаяся на совместном использовании проектного и соревновательного подходов, по эффективности подготовки кадров соответствует уровню лучших мировых университетов, а по некоторым показателям и превышает их уровень.
8. Описанный подход являлся одной из составляющих, позволивших СПбГУ ИТМО в 2007 г. победить в конкурсе инновационных образовательных программ вузов России, а в 2009 г. – в конкурсе на присвоение категории «Национальный исследовательский университет».
9. При сохранении для работы в университете на постоянной основе лучших выпускников и студентов, как это делается на кафедре КТ в течение уже ряда лет, появляется надежда, что когда-нибудь и про наш университет будут писать слова, аналогичные следующим: «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)». Эта надежда базируется и на том, что на чемпионатах мира по программированию команды нашего университета неоднократно побеждали команды МТИ и Стэнфорда, в котором, по словам Карли Фиорины, «также много умных людей».

10. В рамках **национального исследовательского университета СПбГУ ИТМО** на базе кафедры КТ создан **научно-исследовательский центр «Технологии программирования и искусственного интеллекта»**. При этом особое внимание уделяется не только технологиям автоматного программирования, но и работам по верификации автоматных программ и применению эволюционных стратегий для генерации автоматов, являющихся ядром указанного класса программ. Также проводятся исследования для других парадигм программирования.
11. В результате работы системы, созданной на кафедре КТ, удается решать следующие важнейшие задачи, которые стоят перед национальным исследовательским университетом: **воспитание и мотивация к творчеству [168], обучение, научные исследования (генерация знаний в области создания технологий программирования и искусственного интеллекта), создание инноваций, а также сохранение в университете талантливых молодых преподавателей и ученых. Последнее вселяет надежду, что традиции, заложенные на кафедре КТ под руководством В.Н. Васильева, будут сохранены и в будущем.**
12. В 2003 г. за создание системы Всероссийских и Международных олимпиад по информатике и программированию и достигнутые в этих олимпиадах успехи ректору СПбГУ ИТМО, заведующему кафедрой КТ, профессору В.Н. Васильеву, декану факультета «Информационные технологии и программирование», профессору В.Г. Парфенову, старшим преподавателям кафедры КТ Р.А. Елизарову и А.С. Станкевичу была присуждена Премия Президента РФ в области образования.
13. Научно-практическая и методическая разработка «Инновационная система поиска и подготовки высококвалифицированных специалистов в области производства программного обеспечения на основе проектного и соревновательного подходов» удостоена **премии Правительства РФ в области образования за 2008 г.** Авторский коллектив – В.Н. Васильев, В.Г. Парфенов, А.А. Шалыто, М.А. Казаков, Г.А. Корнеев. При этом отметим, что **двум последним авторам было меньше 30 лет!**
14. В 2010 г. за разработку учебно-методического обеспечения для подготовки высококвалифицированных программистов доценту кафедры КТ Г.А. Корнееву и старшему преподавателю этой кафедры С.Е. Столяру была присуждена Премия Правительства Санкт-Петербурга в области образования.
15. В марте 2011 г. произошло событие, о котором можно сказать, что **«мечты сбываются»**. Как отмечено выше в рамках инициативы «Сохраним в университетах лучших!» нам помогают разные компании, в том числе компания *Одноклассники. ru*, входящая в корпорацию *Mail.ru Group*. Руководство этой корпорации с целью развития в России науки в области программной инженерии предложило финансировать в СПбГУ ИТМО кафедру «Программная инженерия и верификация программ», которую возглавит упомянутый выше ученый с

мировым именем **Бертран Мейер**. Это позволит еще восьми молодым ученым на постоянной основе совместно работать с молодыми людьми, которые работают на кафедре КТ в рамках указанной выше инициативы, образуя достаточно большой коллектив, состоящий из талантливой молодежи.

16. В 2011 г. кафедра проведет олимпиаду по программированию для корпорации *Mail.ru Group* «Российский кубок по программированию» (<http://russiancodecup.ru/>) по аналогии с олимпиадами, проводимыми компаниями *Google* и *Facebook*.

Литература

1. Волков А., Ливанов Д., Фурсенко А. Высшее образование: повестка 2008 – 2016 // Эксперт. 2007. № 32, с. 42 – 47.
2. Глас Р. Факты и заблуждения профессионального программирования. СПб.: Символ-Плюс, 2007. – 240 с.
3. Васильев В. Н., Казаков М. А., Корнеев Г. А., Парфенов В. Г., Шалыто А. А. Применение проектного подхода на основе автоматного программирования при подготовке разработчиков программного обеспечения / Труды Первого Санкт-Петербургского конгресса «Профессиональное образование, наука, инновации в XXI веке». СПбГУ ИТМО. 2007, с. 98 – 100.
4. Шалыто А. А. Трехдиагональная задача одного педагогического эксперимента в области ИТ-образования // Инженерное образование. 2007. № 4, с. 208 – 213.
5. Ван Влиет Х. О преподавании программной инженерии // Открытые системы. 2006. № 6, с. 58 – 63.
6. Шалыто А. А. Switch-технология. Алгоритмизация и программирование задач логического управления. СПб.: Наука, 1998. – 628 с.
7. Казаков М. А., Корнеев Г. А., Шалыто А. А. Разработка логики визуализаторов алгоритмов на основе конечных автоматов // Телекоммуникации и информатизация образования. 2003. № 6, с. 27 – 58.
8. Казаков М. А., Шалыто А. А. Использование автоматного программирования для реализации визуализаторов // Компьютерные инструменты в образовании. 2004. № 2, с. 19–33.
9. Корнеев Г. А., Шалыто А. А. Автоматизированное построение визуализаторов алгоритмов дискретной математики // Компьютерные инструменты в образовании. 2006. № 5, с. 16 – 26.
10. Корнеев Г. А., Шалыто А. А. *Vizi* – язык описания логики визуализаторов алгоритмов // Научно-технический вестник СПбГУ ИТМО. Вып. 23. 2005, с.130 – 138.
11. Шалыто А. А. Новая инициатива в программировании «Движение за открытую проектную документацию» // Открытое образование. 2003. № 6, с. 69 – 76.
12. Шалыто А. А. Писать по-русски // PC WEEK/RE. 2006. № 46, с. 52, 53.

13. Красильников Н. Н., Парфенов В. Г., Царев Ф. Н., Шалыто А. А. Виртуальная лаборатория для первоначального обучения проектированию программ // Компьютерные инструменты в образовании. 2007. № 5, с. 62 – 67.
14. Шалыто А. А. Проектный подход при обучении разработке программ // Компьютерные инструменты в образовании. 2009. № 4, с. 30 – 38.
15. Поликарпова Н. И., Шалыто А. А. Автоматное программирование. СПб.: Питер, 2010. – 176 с.
16. Столяров Л. В., Дединский И. Р., Шалыто А. А. Трансляция описаний автоматов, представленных в формате *Microsoft Visio*, в исходный код на языке С // Прикладная дискретная математика. Приложение. 2009. № 1, с. 81 – 83.
17. Елизаров Р. А., Корнеев Г. А. Автоматическое тестирование решений на соревнованиях по программированию // Телекоммуникации и информатизация образования. 2003. № 1, с. 61 – 73.
18. Восьмая Всероссийская олимпиада школьников по информатике и программированию / Под ред. В. Н. Васильева, В. Г. Парфенова и А. С. Станкевича. СПбГУ ИТМО, 2007. – 204 с.
19. Командный чемпионат мира по программированию АСМ 2007/2008. Северо-восточный европейский регион / Под ред. В. Н. Васильева и В. Г. Парфенова. СПбГУ ИТМО, 2007. – 242 с.
20. Казаков М. А. Создание системы проведения интернет-соревнований и дистанционного обучения программированию // Телекоммуникации и информатизация образования. 2002. № 6, с. 81 – 100.
21. Вторая Санкт-Петербургская Интернет-олимпиада / Под ред. В. Н. Васильева, В. Г. Парфенова, С. К. Стафеева. СПбГУ ИТМО, 2007. – 75 с.
22. Васильев В. Н., Лисицына Л. С., Лямин А. В. Результаты апробации технологии сетевой информационной системы при проведении в 2007 г. ЕГЭ по информатике в компьютерной форме / XXXVII научная и учебно-методическая конференция СПбГУ ИТМО. 2008, с. 52 – 54.
23. Никлаус Вирт – почетный доктор СПбГУ ИТМО // Информационно-управляющие системы. 2005. № 5, с. 56 – 58.
24. Бертран Мейер – почетный доктор СПбГУ ИТМО // Информационно-управляющие системы. 2007. № 1, с. 55, 56.
25. Джон Хопкрофт – почетный доктор СПбГУ ИТМО. http://is.ifmo.ru/misc/hopcroft_itmo.pdf
26. Шалыто А. А. Победы и проблемы российской школы программирования // PC WEEK/RE. 2006. № 47, с. 42, 45.
27. Шалыто А. А. Программная реализация управляющих автоматов // Судостроительная промышленность. Серия «Автоматика и телемеханика». 1991. Вып. 13, с. 41, 42.

28. *Shalyto A.* Technology of Automata-Based Programming.
<http://www.codeproject.com/KB/architecture/abp.aspx>
29. *Ненейвода Н. Н.* Стили и методы программирования. М.: Интернет-университет информационных технологий, 2005. – 316 с.
30. *Ненейвода Н. Н., Скопин И. Н.* Основания программирования. М.-Ижевск: Институт компьютерных исследований, 2003. – 812 с.
31. *Harel D. et al.* Statemate: A Working Environment for the Development of Complex Reactive Systems //IEEE Trans. Software Eng. 1990. № 4, pp. 23 – 34.
32. *Шалыто А. А., Туккель Н. И.* От тьюрингова программирования к автоматному //Мир ПК. 2002. № 2, с. 144 – 149.
33. *Хопкрофт Д.* Машины Тьюринга //В мире науки. 1984. № 7, с. 24 – 28.
34. *Шалыто А. А.* Парадигма автоматного программирования // Научно-технический вестник СПбГУ ИТМО. 2008. Вып. 53. Автоматное программирование, с. 3 – 23.
35. *Туккель Н. И., Шамгунов Н. Н., Шалыто А. А.* Ханойские башни и автоматы //Программист. 2002. № 8, с. 82 – 92.
36. *Shalyto A. A.* Cognitive Properties of Hierarchical Representations of Complex Logical Structures /Proceeding of the 1995 International Symposium on Intelligent Control (ISIC). Workshop. 1995. Monterey. California, p. 53.
37. *Шалыто А. А.* Логическое управление. Методы аппаратной и программной реализации алгоритмов. СПб.: Наука, 2000. – 780 с.
38. *Кузьмин Е. В., Соколов В. А.* Моделирование, спецификация и верификация «автоматных» программ //Программирование. 2008. № 1, с. 38 – 60.
39. *Вавилов В. К., Шалыто А. А.* Что плохого в неавтоматном подходе к программированию контроллеров? //Промышленные АСУ и контроллеры. 2007. № 1, с. 49 – 51.
40. *Шалыто А. А.* SWITCH-технология. Алгоритмизация и программирование задач логического управления //Промышленные АСУ и контроллеры. 1999. № 9, с. 33 – 37.
41. *Шалыто А. А.* Автоматное проектирование программ. Алгоритмизация и программирование задач логического управления //Известия РАН. Теория и системы управления. 2000. № 6, с. 63 – 81.
42. *Шалыто А. А.* Реализация алгоритмов логического управления программами на языке функциональных блоков //Промышленные АСУ и контроллеры. 2000. № 4, с. 45 – 50.
43. *Альтерман И. З., Шалыто А. А.* Формальные методы программирования логических контроллеров //Промышленные АСУ и контроллеры. 2005. № 10, с. 49 – 52.
44. *Вавилов В. К., Шалыто А. А.* LabVIEW и SWITCH-технология //Промышленные АСУ и контроллеры. 2006. № 6, с. 43 – 45.

45. Туккель Н. И., Шалыто А. А. Проектирование программного обеспечения системы управления дизель-генераторами на основе автоматного подхода //Системы управления и обработки информации. 2003. Вып. 5, с. 66 – 82.
46. Туккель Н. И., Шалыто А. А. Система дистанционного управления судовым дизель-генератором (Фрагмент). Программирование с явным выделением состояний. 2000. – 51с. http://is.ifmo.ru/download/short_dg.pdf
47. Туккель Н. И., Шалыто А. А. SWITCH-технология – автоматный подход к созданию программного обеспечения «реактивных» систем //Промышленные АСУ и контроллеры. 2000. № 10, с. 44 – 48.
48. Шалыто А. А. Алгоритмизация и программирование для систем логического управления и «реактивных» систем //Автоматика и телемеханика. 2001. № 1, с. 3 – 39.
49. Туккель Н. И., Шалыто А. А. SWITCH-технология – автоматный подход к созданию программного обеспечения «реактивных» систем //Программирование. 2001. № 5, с. 45 – 62.
50. Туккель Н. И., Шалыто А. А. Программирование с явным выделением состояний //Мир ПК. 2001. № 8, с. 116 – 121; № 9, с. 132 – 138.
51. Туккель Н. И., Шалыто А. А. Реализация автоматов при программировании событийных систем //Программист. 2002. № 4, с. 74 – 80.
52. Туккель Н. И., Шалыто А. А. Объектно-ориентированное программирование с явным выделением состояний /Материалы Международной научно-технической конференции «Искусственный интеллект». Т.1. Таганрог. ТГРУ; Донецк. Донецкий гос. ин-т искусств. интеллекта. 2002, с. 198 – 202.
53. Туккель Н. И., Шалыто А. А. Система управления танком для игры *Robocode*. Вариант 1. Объектно-ориентированное программирование с явным выделением состояний. Проектная документация. СПб.: 2001. – 52 с. <http://is.ifmo.ru/download/tanks.pdf>.
54. Туккель Н. И., Шалыто А. А. Танки и автоматы //BYTE/Россия. 2003. № 2, с. 69 – 73.
55. Кузнецов Д. В., Шалыто А. А. Система управления танком для игры *Robocode*. Вариант 2. СПбГ ИТМО (ТУ). 2003. – 86 с. <http://is.ifmo.ru/download/robocode2.pdf>
56. Наумов Л. А., Шалыто А. А. Искусство программирования лифта. Объектно-ориентированное программирование с явным выделением состояний //Информационно-управляющие системы. 2003. № 6, с. 38 – 49.
57. Shamgunov N., Korneev G., Shalyto A. State Machine Design Pattern /.NET Technologies 2006 – Shot communication papers conference proceedings. 4-th International Conference in Central Europe on .Net Technologies. University of West Bohemia. 2006, pp. 51 – 58.
58. Корнеев Г. А., Шамгунов Н. Н., Шалыто А. А. Язык *State Machine* – расширение языка *Java* для эффективной реализации автоматов //Информационно-управляющие системы. 2005. № 1, с. 16 – 24.

59. Шопырин Д. Г., Шалыто А. А. Объектно-ориентированный подход к автоматному программированию //Информационно-управляющие системы. 2003. № 5, с. 29 – 39.
60. Шопырин Д. Г., Шалыто А. А. Синхронное программирование //Информационно-управляющие системы. 2004. № 3, с. 35 – 42.
61. Степанов О. Г., Шопырин Д. Г., Шалыто А. А. Предметно-ориентированный язык автоматного программирования на базе динамического языка RUBY //Информационно-управляющие системы. 2007. № 4, с. 22 – 27.
62. Дмитриев С. Языково-ориентированное программирование //RSDN Magazine. 2005. № 5, с. 23 – 27.
63. Гуров В. С., Мазин М. А., Шалыто А. А. Текстовый язык для автоматного программирования // Научно-технический вестник СПбГУ ИТМО. 2007. Вып. 42, с. 29 – 32.
64. Жукова А. Р., Мазин М. А. Акторное расширение языка Java в среде MPS //Научно-технический вестник СПбГУ ИТМО. 2010. № 2, с. 72 – 77.
65. Stepanov O., Shalyto A. Method for Automatic Runtime Verification of Automata-Based Programs //Proceeding of Spring/Summer Young Researchers` Colloquims on Software Engineering. SPb.: 2008. Vol.2, pp. 19 – 23.
66. Борисенко А. А., Шалыто А. А. Программное средство для автоматической проверки контрактов и темпоральных спецификаций в среде MPS. // Свидетельство о государственной регистрации программы для ЭВМ. № 2010 615076. Дата регистрации – 05.08.2010.
67. Федотов П. В., Степанов О. Г. Внесение изменений в автоматные программы //Научно-технический вестник СПбГУ ИТМО. 2011. Вып. 1, с. 77 – 88.
68. Naumov L., Korneev G., Shalyto A. Methods of Object-Oriented Reactive Agents Implementation on the Basis of Finite Automata /2005 International Conference on «Integration of Knowledge Intensive Multi-Agent Systems: Modeling, Exploration and Engineering» (KIMAS-05). Boston: IEEE. 2005, pp. 460 – 465.
69. Шопырин Д. Г., Шалыто А. А. Графическая нотация наследования автоматных классов //Программирование. 2007. № 4, с. 62 – 74.
70. Астафуров А.А. Декларативный подход к вложению и наследованию автоматных классов при использовании императивных языков программирования //Научно-технический вестник СПбГУ ИТМО. 2008. Вып. 53. Автоматное программирование, с. 230 – 237.
71. Малаховски Я. М., Шалыто А. А. Конечные автоматы на чистых функциональных языках программирования (Автоматы и Haskell) //RSDN Magazine. 2009. № 3, с. 20 – 26.
72. Малаховски Я. М., Шалыто А. А. Реализация конечных автоматов на функциональных языках программирования //Информационно-управляющие системы. 2009. № 6, с. 31 – 33.

73. Малаховски Я. М., Корнеев Г. А. Валидация автоматов с переменными на функциональных языках программирования //Научно-технический вестник СПбГУ ИТМО. 2010. № 8, с. 73 – 77.
74. Малаховски Я. М., Шалыто А. А. Библиотека для поддержки автоматного программирования на языке *Haskell*. // Свидетельство о государственной регистрации программы для ЭВМ. № 2010 614196. Дата регистрации – 29.06.2010.
75. Корнеев Г. А., Шамгунов Н. Н., Шалыто А. А. Обход деревьев на основе автоматного подхода //Компьютерные инструменты в образовании. 2004. № 3, с. 32 – 37.
76. Лобанов П. Г., Шалыто А. А. Подсчет длины слов в строке //Мир ПК. 2005. № 7, с. 66 – 70.
77. Туккель Н. И., Шалыто А. А. Реализация вычислительных алгоритмов на основе автоматного подхода //Телекоммуникации и информатизация образования. 2001. № 6, с. 35 – 53.
78. Туккель Н. И., Шалыто А. А. Преобразование итеративных алгоритмов в автоматные //Программирование. 2002. № 5, с. 45 – 62.
79. Туккель Н. И., Шамгунов Н. Н., Шалыто А. А. Реализация рекурсивных алгоритмов на основе автоматного подхода //Телекоммуникации и информатизация образования. 2002. № 5, с. 72 – 99.
80. Корнеев Г. А. Метод преобразования программы в систему взаимодействующих автоматов //Труды II межвузовской конференции молодых ученых СПбГУ ИТМО. 2005, с. 18 – 23.
81. Головешин А. Использование конвертора Visio2Switch. <http://is.ifmo.ru/progeny/visio2switch/>
82. Волобуев В. Н., Калачинский А. В. Опыт использования автоматного подхода при разработке программного обеспечения систем боевого управления //Системы управления и обработки информации. 2009. Вып. 18, с. 88 – 92.
83. Столяров Л. В. Трансляция описаний автоматов, представленных в формате *Microsoft Visio* в исходный код на языке *C* //Компьютерные инструменты в образовании. 2009. № 5, с. 35 – 44.
84. Канжелев С. Ю., Шалыто А. А. Автоматическая генерация автоматного кода //Информационно-управляющие системы. 2006. № 6, с. 35 – 42.
85. О проекте «Технология автоматного программирования: применение и инструментальные средства» //Информационные технологии. 2006. № 2, с. 79.
86. Гуров В. С., Мазин М. А., Нарвский А. С., Шалыто А. А. Разработка UML. SWITCH-технология. Eclipse //Информационно-управляющие системы. 2004. № 6, с. 12 – 17.
87. Gurov V. S., Mazin M. A., Narvsky A. S., Shalyto A. A. UniMod: Method and Development of Reactive Object-Oriented Programs with Explicit States Emphasis /Proceedings 2005 of St. Petersburg IEEE Chapters. International Conference «10 Anniversary of Radio Invention SPb ETU «LETI». 2005, pp. 106 – 110.

88. Гуров В. С., Мазин М. А., Шалыто А. А. Ядро автоматного программирования //Свидетельство об официальной регистрации программы для ЭВМ. № 2006 613249 от 14.09.2006.
89. Гуров В. С., Мазин М. А., Шалыто А. А. Встраиваемый модуль автоматного программирования для среды разработки //Свидетельство об официальной регистрации программы для ЭВМ. № 2006 613817 от 07.11.2006.
90. Гуров В. С., Мазин М. А., Нарвский А. С., Шалыто А. А. Инструментальное средство для поддержки автоматного программирования //Программирование. 2007. № 6, с. 65 – 80.
91. Гуров В. С., Нарвский А. С., Шалыто А. А. Исполняемый UML из России //PC WEEK/RE. 2005. № 26, с. 18, 19.
92. Paraschenko D., Tsarev F., Shalyto A. Modeling Technology for One Class of Multi-Agent Systems with Automata Based Programming /Proceedings of 2006 IEEE International Conference on Computational Intelligence for Measurement Systems and Application (IEEE CIMSA–2006). Spain, 2006, pp. 15 – 20.
93. Шалыто А. А. Технология автоматного программирования /Труды первой Всероссийской научной конференции «Методы и средства обработки информации». М.: МГУ. 2003, с. 528 – 535.
94. Шалыто А. А. Технология автоматного программирования //Мир ПК. 2003. № 10, с. 74 – 78.
95. Кочелаев Д. Ю. Проектирование, спецификация и реализация автоматизированных классов. Магистерская диссертация. СПбГУ ИТМО. 2009. – 113 с.
96. Ремизов А. О., Шалыто А. А. Применение автоматного подхода при создании программного обеспечения //Системы управления и обработки информации. 2009. Вып. 18, с. 82 – 87.
97. Вавилов К. В. LabVIEW и SWITCH-технология. Методика алгоритмизации и программирования задач логического управления. СПб.: 2005. – 68 с. http://is.ifmo.ru/progeny/_vavilov2.pdf.zip
98. Вавилов К. В. Программируемые логические контроллеры SIMATIC S7-200 (SIEMENS). Методика алгоритмизации и программирования задач логического управления. СПб.: 2005. – 68 с. http://is.ifmo.ru/progeny/_metod065.pdf
99. Вавилов К. В. Контроллеры SIMATIC S7-300 (SIEMENS). Организация взаимодействия независимых локальных систем управления на основе автоматного подхода. СПб.: 2005. – 50 с. http://is.ifmo.ru/progeny/_s7300.pdf
100. Беляев А. В., Суясов Д. И., Шалыто А. А. Компьютерная игра «Космонавт». Проектирование и реализация //Компьютерные инструменты в образовании. 2004. № 4, с. 75 – 84.

101. Корнеев Г. А., Петрошенко П. А., Шалыто А. А. Реализация игры «Морской бой» на основе автоматного подхода //Компьютерные инструменты в образовании. 2005. № 6, с. 72–82.
102. Гуров В. С., Нарвский А. С., Шалыто А. А. ICQ и автоматы //Технология «Клиент-Сервер». 2004. № 3, с. 3 – 11.
103. Владыкин А. А., Шалыто А. А. Непроцедурный текстовый язык описания автоматных обработчиков XML-документов и его применение // Информационные технологии. 2009. № 12, с. 38 – 45.
104. Мазин М. А., Парфенов В. Г., Шалыто А. А. Анимация. FLASH-технология. Автоматы //Компьютерные инструменты в образовании. 2003. № 4, с. 39 – 47.
105. Naumov L., Shalyto A. Automata Theory for Multi-Agent Systems Implementation /International Conference on «Integration of Knowledge Intensive Multi-Agent Systems: Modeling, Exploration and Engineering». (KIMAS-03). Boston: IEEE. 2003, pp. 65 – 70.
106. Yartsev B., Korneev G., Kotov V., Shalyto A. Automata-Based Programming of the Reactive Multi-Agent Control Systems /2005 International Conference on «Integration of Knowledge Intensive Multi-Agent Systems: Modeling, Exploration and Engineering» (KIMAS-05). Boston: IEEE. 2005, pp. 449 – 453.
107. Кретинин А. В., Солдатов Д. В., Шостак А. В., Шалыто А. А. Ракеты. Автоматы. Нейронные сети //Нейрокомпьютеры: разработка и применение. 2005. № 5, с. 50 – 58.
108. Кретинин А. В. Солдатов Д. В. Шостак А. В., Шалыто А. А. Использование нейросетевых конечных автоматов для моделирования функционирования агрегатов жидкостного ракетного двигателя // Информационные технологии. 2005. № 8, с. 47 – 53.
109. Поликарпова Н. И., Шалыто А. А. Парадигма автоматного программирования //RSDN Magazine. 2009. № 1, с. 53 – 58.
110. Клебан В. О., Шалыто А. А., Парфенов В. Г. Построение системы автоматического управления мобильным роботом на основе автоматного подхода //Научно-технический вестник СПбГУ ИТМО. 2008. Вып. 53. Автоматное программирование, с. 281 – 285.
111. Клебан В. О., Шалыто А. А. Разработка системы управления малоразмерным вертолетом //Научно-технический вестник СПбГУ ИТМО. 2011. Вып. 2 (72). Технологии автоматного программирования и искусственного интеллекта, с. 12 – 16.
112. Клебан В. О., Шалыто А. А. Средство для обеспечения взаимодействия автоматных программ, реализующих управление мобильными роботами. //Свидетельство о государственной регистрации программы для ЭВМ. №2009 615456. Дата регистрации – 1.10.2009.
113. Клебан В. О., Шалыто А. А. Система автоматного управления макетом безэкипажного танка. //Свидетельство о государственной регистрации программы для ЭВМ. №2009 615622. Дата регистрации – 8.10.2009.

114. *Алексеев С. А., Калинин А. И., Клебан В. О., Шалыто А. А.* Программный комплекс для исследования автоматного управления роботами. //Свидетельство о государственной регистрации программы для ЭВМ. №2010 614264. Дата регистрации – 30.06.2010.
115. *Янкин Ю. Ю., Шалыто А. А.* Автоматное программирование ПЛИС в задачах управления электроприводом // Информационно-управляющие системы. 2011. № 1, с. 50 – 56.
116. *Вельдер С. Э., Шалыто А. А.* О верификации простых автоматных программ на основе метода *Model Checking* //Информационно-управляющие системы. 2007. № 3, с. 27 – 38.
117. *Яминов Б. Р., Шалыто А. А.* Расширение верификатора *Vogor* для верификации автоматных *UniMod*-моделей //Свидетельство об официальной регистрации программы для ЭВМ. № 2008 611055 от 28.02.2008.
118. *Лукин М. А., Шалыто А. А.* Транслятор автоматной *UniMod*-программы во входной язык верификатора *SPIN* //Свидетельство об официальной регистрации программы для ЭВМ. № 2008 610473 от 25.01.2008.
119. *Курбацкий Е. А., Шалыто А. А.* Транслятор автоматной во входной язык верификатора *SMV* //Свидетельство об официальной регистрации программы для ЭВМ. № 2008 614 235 от 4.09.2008.
120. *Вельдер С. Э., Шалыто А. А.* Система преобразования автоматной модели в модель *Крипке* и верификации *CTL*-свойств на ней //Свидетельство об официальной регистрации программы для ЭВМ. № 2008 614 385 от 11.09.2008.
121. *Вельдер С. Э., Лукин М. А., Шалыто А. А., Яминов Б. Р.* Верификация автоматных программ. СПбГУ ИТМО. 2011. – 241 с.
122. *Клебанов А. А., Степанов О. Г., Шалыто А. А.* Применение шаблонов требований к формальной спецификации и верификации автоматных программ //Научно-технический вестник СПбГУ ИТМО. 2010. № 5, с. 91 – 95.
123. *Ремизов А. О., Шалыто А. А.* Верификация автоматных программ на основе метода *Model Checking* //Системы управления и обработки информации. 2010. Вып. 21, с. 84 – 94.
124. *Поликарпова Н. И., Точилин В.Н., Шалыто А.А.* Метод сокращенных таблиц для генерации автоматов с большим числом входных переменных на основе генетического программирования //Известия РАН. Теория и системы управления. 2010. № 2, с. 100 – 117.
125. *Поликарпова Н.И., Точилин В.Н.* Генетический генератор автоматов //Свидетельство об официальной регистрации программы для ЭВМ. № 2008 610473 от 25.01.2008.
126. *Данилов В. Р., Шалыто А. А.* Метод представления функций переходов деревьями решений для генерации автоматов /Сборник научных трудов V-я Международной научно-практической конференции «Интегрированные модели и мягкие вычисления в искусственном интеллекте». Т. 2. Коломна: Физматлит. 2009, с. 589 – 595.

127. Данилов В. Р., Шалыто А. А. Представление функций переходов линейными бинарными графами при генерации управляющих автоматов с помощью генетического программирования / Труды Третьей Всероссийской научной конференции «Методы и средства обработки информации». МГУ. 2009, с. 110 – 115.
128. Данилов В. Р., Шалыто А. А. Программное средство для генерации на основе генетического программирования автоматов, представленных линейными бинарными графами. //Свидетельство о государственной регистрации программы для ЭВМ. № 2010 615014. Дата регистрации – 03.08.2010.
129. Царев Ф. Н. Метод построения управляющих конечных автоматов на основе тестовых примеров с помощью генетического программирования // Информационно-управляющие системы. 2010. № 5, с. 31 – 36.
130. Царев Ф. Н. Программное средство для построения управляющих автоматов на основе обучающих примеров с использованием генетических алгоритмов. //Свидетельство о государственной регистрации программы для ЭВМ. № 2010 614197. Дата регистрации – 29.06.2010.
131. Егоров К.В., Царев Ф. Н., Шалыто А. А. Совместное применение генетического программирования для построения автоматов управления системами со сложным поведением // Труды СПИИРАН. 2010. Вып. 15, с. 123 – 135.
132. Александров А. В., Казаков С. В., Сергушичев А. А., Царев Ф. Н., Шалыто А. А. Генерация конечных автоматов для управления моделью беспилотного самолета //Научно-технический вестник СПбГУ ИТМО. 2011. Вып. 2 (72). Технологии автоматного программирования и искусственного интеллекта, с. 3 – 12.
133. Царев Ф. Н. Совместное применение генетического программирования, конечных автоматов и искусственных нейронных сетей для построения системы управления беспилотным летательным аппаратом //Научно-технический вестник СПбГУ ИТМО. 2008. Вып. 53. Автоматное программирование, с. 42 – 59.
134. Лобанов П. Г., Шалыто А. А. Использование генетических алгоритмов для автоматического построения конечных автоматов в задаче о флибах //Известия РАН. Теория и системы управления. 2007. № 5, с. 127 – 136.
135. Мандриков Е. А., Кулев В. А., Шалыто А. А. Применение генетического программирования при решении задачи о флибах //Информационные технологии. 2007. № 12, с. 42 – 45, 89.
136. Царев Ф. Н., Шалыто А. А. О построении автоматов с минимальным числом состояний для задачи об «умном муравье» /Сборник докладов X международной конференции по мягким вычислениям и измерениям. СПбГУ ЭТУ «ЛЭТИ». 2007. Т. 2, с. 209 – 215.
137. Попов С. И., Попов Ю. И., Шалыто А. А. Задача о муравьеде и муравьях //Информационные технологии. 2010. № 8, с. 18 – 22.

138. *Davydov A., Sokolov D., Tsarev F., Shalyto A.* Application of Genetic Programming for Generation of Controllers represented by Automata / Preprints of the 13th IFAC Symposium on Information Control Problems in Manufacturing. Moscow. 2009, pp. 684 – 689.
139. *Тяhti А. С., Чебатуркин А. А., Царев Ф. Н. Шалыто А. А.* Виртуальная лаборатория для обучения методам искусственного интеллекта для генерации управляющих конечных автоматов / Сборник докладов IV Международной научно-практической конференции «Современные информационные технологии и ИТ-образование». МГУ. 2009, с. 222 – 227.
140. *Чащин Д. А., Шалыто А. А.* Распределенная виртуальная лаборатория для обучения генетическим алгоритмам //Сборник трудов V международной научно-практической конференции «Современные информационные технологии и ИТ-образование». МГУ. 2010, с. 245 – 252.
141. *Оршанский С. А., Шалыто А. А.* Применение динамического программирования при решении задач на конечных автоматах //Компьютерные инструменты в образовании. 2006. № 4, с. 26 – 35.
142. *Буздалов М. В.* Применение генетических алгоритмов для определения неэффективных решений олимпиадных задач по программированию (на примере задачи о рюкзаке) / Сборник статей третьей Всероссийской научной конференции «Нечеткие системы и мягкие вычисления». Т.2. Волгоград: ВолгГТУ. 2009, с. 16 – 24.
143. *Бедный Ю. Д., Шалыто А. А.* Применение генетических алгоритмов для создания системы управления танком в игре Robocode / Труды четвертой международной конференции по проблемам управления. М.: Институт проблем управления. 2009, с. 1521 – 1529.
144. *Zakonov A., Stepanov O., Shalyto A.* GA-Based and Design by Contract Approach to Test Generation for EFSMs /Proceedings of IEEE East-West Design & Test Symposium (EWDTS'10). St. Petersburg. 2010, pp. 152 – 155.
145. *Наумов Л. А., Шалыто А. А.* Клеточные автоматы. Реализация и эксперименты //Мир ПК. 2003. № 8, с. 71 – 78.
146. *Наумов Л. А., Шалыто А. А.* Цветные клеточные автоматы, или клонирование Мона Лизы //Мир ПК. 2004. № 5, с. 64 – 71.
147. *Наумов Л. А., Шалыто А. А.* Клеточные структуры, порождаемые одномерными двоичными клеточными автоматами из точечного зародыша //Известия РАН. Теория и системы управления. 2005. № 5, с. 137 – 145.
148. *Наумов Л. А.* Решение задач с помощью клеточных автоматов посредством программного обеспечения *CAMEL* //Информационно-управляющие системы. 2005. № 5, с. 22 – 30; № 6, с. 30 – 38.

149. *Суясов Д. И.* Выделение структурных признаков изображения символов на основе клеточных автоматов с метками //Информационно-управляющие системы. 2010. № 4, с. 39 – 45.
150. *Тихомиров А. В., Шальто А. А.* Применение генетического подхода для генерации клеточных автоматов //Научно-технический вестник СПбГУ ИТМО. 2011. Вып. 2 (72), с. 62 – 66.
151. *Proceedings of Student Conferences in Mathematics* (Ed. by V.N. Vasiliev). St. Petersburg Institute of Fine Mechanics and Optics (Technical University). 1997.
http://is.ifmo.ru/works/_vas.pdf
152. Научно-технический вестник СПбГУ ИТМО. 2008. Вып. 53. Автоматное программирование. – 314 с.
153. Научно-технический вестник СПбГУ ИТМО. 2011. Вып. 2 (72). Технологии автоматного программирования и искусственного интеллекта. – 85 с.
154. *Поликарпова Н. И.* Отношение наследования для типов со сложным поведением //Научно-технический вестник СПбГУ ИТМО. 2006. № 25, с. 44 – 56.
155. *Shopyrin D.* Multimethods in C++ using recursive deferred dispatching //IEEE Software. 2006. № 3, pp. 62 – 73.
156. *Князев Е. Г., Шопырин Д. Г.* Автоматизированная классификация изменений программного кода методами многомерного статистического анализа //Информационные технологии. 2008. № 5. с. 48 – 53.
157. *Князев Е. Г., Шопырин Д. Г.* Использование автоматизированной классификации изменений программного кода в управлении процессом разработки программного обеспечения //Информационно-управляющие системы. 2010. № 9, с. 15 – 21.
158. *Паращенко Д. А.* Обработка строк на основе суффиксных автоматов. Бакалаврская работа. СПбГУ ИТМО. 2007. – 36 с. <http://is.ifmo.ru/papers/paraschenko/>
159. *Паращенко Д. А.* Суффиксные автоматы с сохранением промежуточных версий и их приложения. Магистерская диссертация. СПбГУ ИТМО. 2009. – 59 с.
http://is.ifmo.ru/papers/_paraschenko-masters.pdf
160. *Акишев И. Р., Дворкин М. Э.* О построении минимальных детерминированных конечных автоматов, распознающих префиксный код заданной мощности //Прикладная дискретная математика. 2010. № 2, с. 104 – 116.
161. *Станкевич А. С.* Использование алгоритмов анализа левоконтекстных терминальных грамматик в задачах автоматического тестирования программ //Труды СПИИРАН. 2010. Вып. 13, с. 106 – 121.

162. *Дворкин М. Э.* Методы минимизации необходимого числа цепей для секвенирования ДНК. Магистерская диссертация. СПбГУ ИТМО. 2010. – 44 с.
http://is.ifmo.ru/papers/_dvorkin_genom.pdf
163. *Исенбаев В. В.* Разработка системы секвенирования ДНК с использованием paired-end данных. Бакалаврская работа. СПбГУ ИТМО. 2010. – 30 с.
http://is.ifmo.ru/genom/_isenbaev_thesis.pdf
164. *Капун Е. Д.* Разработка метода сравнения нуклеотидных последовательностей путем разбиения на фрагменты. Бакалаврская работа. СПбГУ ИТМО. 2010. – 26 с.
http://is.ifmo.ru/genom/_kapun_thesis.pdf
165. *Dvorkin M., Isenbaev V., Kapun E.* Genomics Quick Start. SPbSU ИТМО. 21.06.2010.
http://is.ifmo.ru/present/_mit.ppt
166. *Васильев В. Н., Казаков М. А., Корнеев Г. А., Парфенов В. Г., Шалыто А. А.* Инновационная система поиска и подготовки высококвалифицированных разработчиков программного обеспечения на основе проектного и соревновательных подходов / Труды Первого Санкт-Петербургского конгресса «Профессиональное образование, наука, инновации в XXI веке». СПбГУ ИТМО. 2007, с. 84 – 97.
167. *Васильев В. Н., Казаков М. А., Корнеев Г. А., Парфенов В. Г., Шалыто А. А.* Три кита подготовки программистов //Открытые системы. 2009. № 3, с. 54 – 56.
168. *Шалыто А. А.* Заметки о мотивации. СПб.: Мозаика НК, 2011. – 72 с.

Приложение. Успехи студентов и выпускников кафедры «Компьютерные технологии» СПбГУ ИТМО на соревнованиях по программированию

1. Чемпионат мира ACM ICPC (ACM ICPC – Международная студенческая командная олимпиада по программированию, проводящаяся под эгидой международной ассоциации Association for Computing Machinery)

- 2010 год, Антон Ахи, Антон Банных, Сергей Поромов – 14 место
- **2009 год, Максим Буздалов, Владислав Исенбаев, Евгений Капун – 1 место, золотая медаль, чемпионы мира и Европы**
- **2008 год, Дмитрий Абдрашитов, Дмитрий Парашенко, Федор Царев – 1 место, золотая медаль, чемпионы мира и Европы**
- 2007 год, *Искандер Акишев, Михаил Дворкин, Роман Сатюков – 3 место, золотая медаль*
- 2006 год, Искандер Акишев, Михаил Дворкин, Роман Сатюков – 13 место
- 2005 год, *Павел Маврин, Сергей Оршанский, Дмитрий Павлов – 3 место, золотая медаль*
- **2004 год, Павел Маврин, Сергей Оршанский, Дмитрий Павлов – 1 место, золотая медаль, чемпионы мира и Европы**

- 2003 год, Тимофей Бородин, Александр Штучкин, Евгений Южаков – 3 место, золотая медаль
- 2002 год, Тимофей Бородин, Александр Штучкин, Евгений Южаков – 11 место
- 2001 год, Георгий Корнеев, Денис Кузнецов, Андрей Станкевич – 3 место, золотая медаль
- 2000 год, Георгий Корнеев, Денис Кузнецов, Андрей Станкевич – 4 место, серебряная медаль
- 1999 год, Александр Волков, Матвей Казаков, Владимир Левкин – 3 место, золотая медаль
- 1998 год, Александр Волков, Матвей Казаков, Марк Сандлер – 11 место
- 1997 год, Роман Елизаров, Денис Кисловский, Марк Сандлер – 11 место
- 1996 год, Александр Аникин, Денис Кисловский, Роман Елизаров – 19 место

2. Чемпионат России

- **2010 год, Антон Ахи, Антон Банных, Сергей Поромов – 1 место, чемпионы России**
- 2010 год, Сергей Мельников, Владислав Исенбаев, Евгений Капун – 4 место
- 2009 год, Антон Ахи, Антон Банных, Сергей Поромов – 3 место
- 2009 год, Максим Буздалов, Владислав Исенбаев, Евгений Капун – 4 место
- 2008 год, Максим Буздалов, Владислав Исенбаев, Евгений Капун – 6 место
- **2007 год, Дмитрий Абдрашитов, Дмитрий Паращенко, Федор Царев – 1 место, чемпионы России**
- 2006 год, Искандер Акишев, Михаил Дворкин, Роман Сатюков – 2 место
- 2005 год, Искандер Акишев, Михаил Дворкин, Роман Сатюков – 2 место
- **2004 год, Павел Маврин, Сергей Оршанский, Дмитрий Павлов – 1 место, чемпионы России**
- **2003 год, Павел Маврин, Сергей Оршанский, Дмитрий Павлов – 1 место, чемпионы России**
- 2002 год, Тимофей Бородин, Александр Штучкин, Евгений Южаков – 6 место
- **2001 год, Тимофей Бородин, Александр Штучкин, Евгений Южаков – 1 место, чемпионы России**
- 2001 год, Олег Пестов, Андрей Пестов, Иван Прокушкин – 3 место
- 2000 год, Георгий Корнеев, Денис Кузнецов, Андрей Станкевич – 2 место
- 1999 год, Андрей Беломутский, Денис Кузнецов, Андрей Станкевич – 9 место
- 1998 год, Александр Волков, Матвей Казаков, Владимир Левкин – 4 место
- 1997 год, Александр Волков, Матвей Казаков, Марк Сандлер – 5 место
- **1996 год, Роман Елизаров, Денис Кисловский, Марк Сандлер – 1 место, чемпионы России**

3. *Imagine Cup* – международный технологический студенческий конкурс, проводимый при поддержке компании *Microsoft*.

Imagine Cup 2008, категория «Программные проекты»

- Анатолий Никитин (СПбГУ ИТМО), Роман Белов (СПбГУ) и Дарья Элькина (СПб ЛЭТИ) – победители Всероссийского этапа, награда «The Engineering Excellence»

Achievement Award» (награда за выдающееся инженерное решение) на финальном этапе в Париже.

4. TopCoder Open — открытый чемпионат для участников старше 18 лет, фактически чемпионат мира по программированию в индивидуальном зачете.

4.1. TopCoder Open 2009

- Владислав Исенбаев – 2 место, Андрей Станкевич и Павел Маврин участвовали в финале в номинации «Алгоритмы». Дмитрий Трофимов – 4 место в номинации «Марафон».

4.2. TopCoder Open 2008

- Андрей Станкевич – 6–9 место, Павел Маврин и Роман Сатюков участвовали в финале.

4.3. TopCoder Open 2007

- Михаил Дворкин – 8 место, Павел Маврин, Роман Сатюков и Андрей Станкевич – участвовали в финале.

4.4. TopCoder Open 2006

- Андрей Станкевич – 6 место, Дмитрий Абдрашитов участвовал в финале.

5. TopCoder Collegiate Challenge – открытый чемпионат для студентов и аспирантов. С 2008-го года не проводится.

5.1. TopCoder Collegiate Challenge 2007

- Михаил Дворкин и Федор Царев участвовали в финале.

5.2. TopCoder Collegiate Challenge 2006

- Андрей Станкевич – 4 место, Михаил Дворкин и Роман Сатюков участвовали в финале.

6. Google Code Jam – открытые индивидуальные соревнования по программированию, проводимые компанией Google.

6.1. Google Code Jam 2008

- Павел Маврин, Дмитрий Трофимов, Андрей Станкевич, Владислав Исенбаев, Антон Ахи участвовали в финале.

6.2. В 2007-м году соревнования не проводились.

6.3. Google Code Jam 2006

- Андрей Станкевич – 3 место, Павел Маврин – 7 место, Роман Елизаров, Роман Сатюков, Михаил Дворкин участвовали в финале.

6.4. Google Code Jam Europe 2006

- Роман Елизаров – 3 место, Михаил Дворкин – 6 место, Андрей Станкевич – 10 место, Павел Маврин участвовал в финале.

7. International Problem Solving Contest – Международные командные интернет-соревнования по решению задач, проводимые ежегодно университетом Братиславы.

- 2009 год, индивидуальный зачет – Владислав Исенбаев – 5 место
- 2008 год, Владислав Исенбаев, Андрей Станкевич, Федор Царев – 3 место
- 2007 год, Искандер Акишев, Михаил Дворкин, Роман Сатюков – 1 место
- 2007 год, индивидуальный зачет – Андрей Станкевич – 1 место
- 2006 год, Искандер Акишев, Михаил Дворкин, Роман Сатюков – 4 место
- 2005 год, Георгий Корнеев, Андрей Станкевич – 8 место
- 2004 год, Павел Маврин, Сергей Оршанский, Дмитрий Павлов – 9 место
- 2003 год, Андрей Станкевич, Евгений Южаков – 2 место
- 2002 год, Георгий Корнеев, Андрей Станкевич – 1 место

8. Кубок Главы Республики Карелия – командные соревнования, проходящие ежегодно с 2004 года на зимних учебно-тренировочных сборах в Петрозаводске.

- 2009 год, Максим Буздалов, Владислав Исенбаев, Евгений Капун – 2 место
- 2008 год, Дмитрий Абдрашитов, Дмитрий Паращенко, Федор Царев – 4 место
- 2007 год, Искандер Акишев, Михаил Дворкин, Роман Сатюков – 1 место
- 2006 год, Искандер Акишев, Михаил Дворкин, Роман Сатюков – 1 место
- 2005 год, Павел Маврин, Сергей Оршанский, Дмитрий Павлов – 1 место
- 2004 год, Павел Маврин, Сергей Оршанский, Дмитрий Павлов – 1 место

9. Всесибирская олимпиада по информатике и программированию им. И. В. Поттосина. Проводится ежегодно в Новосибирском государственном университете.

9.1. Олимпиада 2010 года.

- Владислав Исенбаев, Евгений Капун, Сергей Мельников – 1 место

9.2. Олимпиада 2009 года.

- Максим Буздалов, Владислав Исенбаев, Евгений Капун – 1 место

9.3. Олимпиада 2008 года.

- Максим Буздалов, Владислав Исенбаев, Евгений Капун – 1 место

9.4. Олимпиада 2007 года.

- Дмитрий Абдрашитов, Дмитрий Паращенко, Федор Царев – 2 место

10. Test The Best (Республика Беларусь) – открытые индивидуальные интернет-соревнования по программированию с условиями задач на русском языке.

Зимний Кубок Test The Best 2007:

- Михаил Дворкин – 1 место;
- Павел Маврин – 3 место.

11. Всероссийская студенческая олимпиада «Информатика. Программирование. Информационные технологии». Проводится ежегодно в Воронеже.

Олимпиада 2007-го года:

- Михаил Дворкин – 1 место;

- Антон Ахи – победитель в номинации «Первокурсники».

12. *Snarknews Series* (организаторы — журнал «Мир ПК» и издательство «Открытые системы»)

12.1. SnarkNews Summer Series – 2008

- Владислав Исенбаев — 3 место

12.2. Snarknews Winter Series – 2007

- Андрей Станкевич – 3 место, Федор Царев – 9 место

13. *Открытый кубок по программированию им. Е. В. Панкратьева.* Проводится МГУ и корпорацией CBOSS (до 2007 года), Яндекс (с 2008 года).

- 2008 год, Максим Буздалов, Владислав Исенбаев, Евгений Капун – 4 место
- 2007 год, Дмитрий Абдрашитов, Дмитрий Парашенко, Федор Царев – 3 место
- 2006 год, Искандер Акишев, Михаил Дворкин, Роман Сатюков – 10 место
- 2005 год, Искандер Акишев, Михаил Дворкин, Роман Сатюков – 5 место
- 2004 год, Павел Маврин, Сергей Оршанский, Дмитрий Павлов – 1 место
- 2004 год, Искандер Акишев, Михаил Дворкин, Роман Сатюков – 3 место

14. *Турнир по программированию, проводимый ОАО «ICL-КПО ВС» (Казань)*

- 2009 год, Антон Ахи, Антон Банных, Сергей Поромов – 1 место
- 2009 год, Михаил Кевер, Сергей Мельников, Всеволод Опарин – 4 место
- 2008 год, Денис Елкин, Владислав Исенбаев, Всеволод Опарин – 1 место

15. *Всероссийская командная олимпиада школьников по программированию* проводится ежегодно с 2000-го года одновременно в двух городах — Санкт-Петербурге и Барнауле.

- 2006 год, Центр подготовки при СПбГУ ИТМО (Антон Ахи, Сергей Поромов, Антон Феськов) – 3 место
- 2005 год, Центр подготовки при СПбГУ ИТМО (Максим Гладких, Олег Давыдов, Александр Сидоров) – 3 место
- 2003 год, Центр подготовки при СПбГУ ИТМО (Роман Белов, Александра Зыкова, Кирилл Чихачев) – 3 место
- 2002 год, Центр подготовки при СПбГУ ИТМО (Дмитрий Зинченко, Роман Сатюков, Игорь Синев) – 1 место
- 2001 год, Центр подготовки при СПбГУ ИТМО – 10 место
- 2000 год, Центр подготовки при СПбГУ ИТМО – 1 место

16. Чемпионат Урала. Проводится ежегодно в Уральском государственном университете.

16.1. 2010 год.

- Сергей Мельников, Владислав Исенбаев, Евгений Капун – 1 место
- Антон Ахи, Антон Банных, Сергей Поромов – 3 место

16.2. 2009 год.

- Максим Буздалов, Владислав Исенбаев, Евгений Капун – 1 место

16.3. 2008 год.

- Максим Буздалов, Владислав Исенбаев, Евгений Капун – 3 место

17. Олимпиада компании Facebook

- Андрей Станкевич – 9 место в предварительных соревнованиях. Участник очного тура соревнований.

18. Рейтинг TopCoder. Рейтинг имеют более 6000 человек в мире (<http://topcoder.com/tc>).

В этом рейтинге представители университета ИТМО, например на 30.10.2009 г., занимали:

- Владислав Исенбаев – 3 место, рейтинг 3356;
- Андрей Станкевич – 8 место, рейтинг 3156.

19. Международная деятельность. В результате тренировок, проведенных Федором Царевым, Максимом Буздаловым, Андреем Станкевичем, команда Высшей политехнической школы Цюриха (ETH Zurich), в которой учились и работали 21 Нобелевский лауреат, в 2010 году впервые в истории выиграла полуфинальные соревнования юго-западного европейского региона (<http://swerc.eu/>) и единственная от этого региона прошла в финал чемпионата мира. Тренер команды на чемпионате мира – Федор Царев.