Д. С. Гладских,
Нижегородский государственный технический университет им. Р. Е. Алексеева,

А. А. Штанюк,
Нижегородский государственный университет им. Н. И. Лобачевского

О ПРОБЛЕМАХ ФОРМИРОВАНИЯ КОМПЕТЕНЦИЙ В ОБЛАСТИ ПРОГРАММИРОВАНИЯ У БАКАЛАВРОВ IT-НАПРАВЛЕНИЯ

Аннотация
В статье рассматриваются актуальные проблемы обучения программированию в технических вузах в плане формирования необходимых компетенций для успешного позиционирования бакалавров IT-направления на рынке труда. Анализируется влияние интеграции различных дисциплин на трудоустройство выпускников бакалавриата в IT-сфере рынка труда.
Ключевые слова: программирование, математическая логика, обучение, бакалавр, рынок труда, IT-компания, компетенции.

В настоящее время на рынке труда РФ потребность в специалистах сферы информационных технологий очень высока. За последние десять лет количество вакансий в сфере IT выросло в 18 раз [2]. Среди IT-специалистов наиболее востребованы те, кто имеет отношение к разработке программного обеспечения (например, в 2013 году среди всех размещенных вакансий в группе IT доля разработчиков программного обеспечения составила 34 % [3]). В то же время число выпускников российских вузов существенно превышает необходимый норматив специалистов с высшим образованием, что, казалось бы, должно приводить к быстрому заполнению появляющихся вакансий в IT-сфере на рынке труда. Однако в реальности этого не происходит. В большинстве случаев серьезные фирмы проводят подготовку, доучивание или просто обучение своих сотрудников необходимым технологиям в области программирования на специализированных курсах. При этом часто получается, что такое обучение выпускников бакалавриата строится практически с нуля.

То есть, **несмотря на то что бакалавры изучали программирование в течение нескольких лет обучения в вузе, уровень их подготовки оказывается недостаточным для того, чтобы занять желаемую вакансию на рынке труда.** Такая ситуация связана с целым рядом причин, среди которых можно выделить следующие:

- отсутствие в вузах специалистов-преподавателей, обеспечивающих высокий уровень образовательного процесса;
- отсутствие необходимого аппаратного и программного обеспечения для проведения занятий по дисциплинам профессионального цикла;
- отсутствие интереса и мотивации со стороны учащихся;
- недостатки организации учебного плана, отсутствие необходимых взаимосвязей между дисциплинами, в том числе общепрофессионального и профессионального циклов.

Проблема, вызванная нехваткой квалифицированного профессорско-преподавательского состава, к сожалению, имеет давние корни. Совершенно очевидно, что многие ведущие специалисты предпочитают работать в сфере IT, а не в сфере

Контактная информация

Штанюк Антон Александрович, канд. тех. наук, доцент, доцент кафедры «Программная инженерия» Нижегородского государственного университета им. Н. И. Лобачевского; *адрес:* 603950, г. Нижний Новгород, пр-т Гагарина, д. 23; *телефон:* (831-2) 62-33-56; *e-mail:* shtan@land.ru

D. S. Gladskikh,
Nizhny Novgorod State Technical University named after R. E. Alekseev,

A. A. Shtanyuk,
Lobachevsky State University of Nizhny Novgorod

ON THE PROBLEMS OF FORMING COMPETENCIES IN PROGRAMMING IN THE BACHELORS OF IT SPHERE

Abstract
The article describes the actual problems of teaching programming in technical universities in terms of forming the necessary competencies for the successful positioning of bachelors of IT sphere in the labor market. The analysis of the effect of integration of the different disciplines on the employment of graduates of undergraduate in IT sphere of the labor market is given.
Keywords: programming, mathematical logic, teaching, bachelor, labor market, IT companies, competencies.

образования по причине недостаточно высокой оплаты труда. Еще одним фактором, влияющим на ситуацию, является нестабильность системы образования, непрерывная череда реформ, снижающая уверенность в завтрашнем дне. Преподаватели, остающиеся на своих местах, к сожалению, часто не могут поддерживать актуальность своих знаний из-за высокой загруженности и занятости, поскольку им приходится работать сразу на нескольких работах. Решение данной проблемы видится в росте заработной платы в сфере образования, что могло бы привлечь специалистов IT-сферы для сотрудничества с вузами, а также в подготовке учебными заведениями собственных научно-педагогических кадров, заинтересованных в дальнейшей работе со студентами.

Для решения ***проблемы, связанной с нехваткой аппаратных и программных средств*** для профессиональной подготовки молодых специалистов, часто не хватает волевого решения руководства вуза, поскольку для приобретения соответствующих средств и инструментов необходимы финансовые затраты, которые в определенных случаях не удается обосновать. Но в этой области намечаются положительные сдвиги, поскольку материально-техническому оснащению вузов начинают уделять все большее внимание.

Удивительно, но в сфере информационных технологий ***низкая мотивация со стороны студентов*** не менее распространена, чем в других сферах. Часто можно слышать о том, что молодежь испытывает естественный интерес к информационным технологиям, а высокая потребность в специалистах на рынке труда стимулирует процесс обучения. Однако в [1] отмечается, что до 87 % всех студентов и IT-специалистов выбирают эту специальность не по причине интереса к предмету, а по другим соображениям, среди которых лидируют пример родных и воля родителей. Для исправления ситуации необходимо повышать популяризацию IT-сферы, проводить работу со школьниками и стимулировать их интерес. Сейчас эту задачу пытаются решить широким распространением детской робототехники и организацией соответствующих курсов.

Попробуем рассмотреть ***возможное решение проблемы, связанной с разработкой учебных планов.***

Суть этого решения заключается **в формировании необходимых связей между тремя наборами дисциплин:**

- *первый набор* — «Математическая логика», «Теория алгоритмов», «Дискретная математика». Этот набор, который мы далее будем называть *«Математическая логика и теория алгоритмов»*, является своего рода теоретическим фундаментом для последующего изучения программирования;
- *второй набор* — дисциплины, которые можно обозначить как *«Алгоритмы и структуры данных»*. Здесь обучение концентрируется на конкретных алгоритмах, рассматриваются наиболее распространенные коллекции данных, например, списки, деревья и хэш-таблицы;
- *третий набор* — это набор профессиональных дисциплин под общим названием *«Программирование»*, который может включать в себя изучение языков, технологий и конкретных прикладных областей.

Далее в статье мы покажем важность последовательного изучения сначала «Математической логики и теории алгоритмов», затем «Алгоритмов и структур данных» и потом самого «Программирования».

На наш взгляд, **часть проблем формирования необходимых компетенций будущего бакалавра в плане информационных технологий связана с разобщенностью преподавания указанных наборов дисциплин.** Так, «Программирование» обычно в большей части касается изучения того или иного языка (а также среды программирования) на примере несложных интуитивно понятных алгоритмов. Если это остается основной составляющей в ходе всего процесса обучения, то как следствие возникает проблема с востребованностью специалистов. В настоящее время на рынке труда в основном востребованы специалисты для программирования сложных систем с использованием современных информационных технологий. В результате при рассмотрении требований вакансий бакалавры не видят себя в качестве реальных кандидатов на перспективные рабочие места либо сознательно завышают уровень своих знаний с целью получить работу.

Дисциплина «Структуры данных и алгоритмизация», на наш взгляд, призвана заполнить разрыв между «Теорией алгоритмов» и «Программированием». Дело в том, что в рамках теории алгоритмов изучаются довольно абстрактные математические проблемы, применимые к теоретической информатике. Рассмотрению подвергаются такие понятия, как «сложность алгоритма», «вычислимость», «теоретический компьютер». При переходе к программированию, к решению прикладных задач у студентов часто возникает своеобразный разрыв между абстракциями и практическими реализациями, что негативно отражается на всем комплексе знаний и неизбежно приводит к пониманию программирования как «ремесла», делая невозможным компетентностный подход к профессии.

Рассматривая **общие проблемы изучения компьютерных дисциплин в современном вузе**, следует прежде всего отметить, что в ряде учебных заведений дисциплины всех трех наборов читаются без внутренней связи. Будущие бакалавры познают программирование, не изучив предварительно структуры данных и основные приемы алгоритмизации. В итоге ими часто применяются решения «в лоб» и построчное кодирование без предварительного составления алгоритма (модели) программы. Кроме естественной неэффективности полученной таким образом программы это часто приводит к ошибкам (особенно в задачах с несколькими единицами трансляции). Итак, подчеркнем, что ***одной из основных проблем в изучении программирования является непоследовательность:*** подход к изучению языков программирования до освоения логики и теории алгоритмов, когда учащиеся не имеют общих представлений о том, что есть алгоритм и почему компьютерная программа является алгоритмом ре-

шения какой-либо поставленной задачи (неумение «мыслить, как компьютер»).

Как показывает опыт, важной проблемой часто становится ***неумение студентов воспринимать то, как компьютер обрабатывает информацию***. Для многих это приводит к тому, что вычислительная машина воспринимается в виде «черного ящика», структура которого неизвестна и неизвестен принцип действия. В частности, распространено применение в операциях и выражениях тех данных, которые не определены до их использования. В этом плане перед изучением языков программирования мы бы порекомендовали обучение студентов работе с машиной Тьюринга, как наиболее наглядным пособием по составлению алгоритма для вычислительной задачи с помощью простых обозначений.

В процессе программирования осуществляется последовательное преобразование различных описаний решаемой задачи: от неформального описания до текста, написанного на определенном языке программирования. В ходе преобразования возникают ошибки, вызванные ***неправильной интерпретацией отдельных конструкций алгоритма, незнанием специфики языка программирования***. На первом этапе дается самое общее описание задачи, а основными ошибками со стороны студентов являются: неверная интерпретация поставленной задачи, непредусмотренное сокращение или расширение перечня действий для решения задачи. Следует добавить, что такие ситуации могут возникать как по вине студента, так и по вине преподавателя, который недостаточно четко и однозначно сформулировал задачу.

Часто встречаются ***две крайности преподавательского подхода к обучению***. Одна из них связана с тем, что после разъяснения теории происходит непосредственный переход к самостоятельному решению задач студентами и это оказывается очень сложным. Большинству студентов для решения необходимы аналогии, без которых многие часто лишаются мотивации («слишком сложно», «все равно не смогу»). Другая крайность заключается в том, что после разъяснения теории производится методичное прорешивание задач разных типов преподавателем самостоятельно или интерактивно со студентами. После этого этап самостоятельного решения аналогичных задач сводится исключительно к поиску нужной аналогии и сводит на нет овладение способами самостоятельного поиска новых решений. В этом случае также может произойти снижение мотивации к обучению программированию — встречая исключительно простые задачи, студент расслабляется, а затем, по ходу обучения дойдя до сложной задачи, теряется. В целом можно сказать, что ***большинство студентов имеют трудности с самостоятельным поиском решений задач и часто пользуются аналогиями***. При этом встречается неумение работать с аналогиями: определение общей аналогии решения, разделение алгоритмов на части и синтез частей разных алгоритмов для решения конкретной задачи. Можно сказать, что необходим баланс между самостоятельным поиском новых решений и работой с аналогиями: овладение способами составления алгоритмов для решения новых задач путем анализа, разбиения, синтеза уже изученных алгоритмов.

Для профессионального занятия программированием очень важно умение выявлять сложности в решении задач и пытаться самостоятельно справляться с ними. В течение всей карьеры специалист совершенствует подобные навыки, тратя на очередную поставленную задачу все меньше и меньше времени.

С нашей точки зрения, **оптимальным подходом при организации занятий** является следующий. После теоретической части преподавателем приводится несколько простейших программ, демонстрирующих работу конкретных теоретических аспектов (операторы, выражения и стандартные функции). Далее студентам предлагается пример многосоставной задачи, для решения которой необходимо (сначала интерактивно, а потом самостоятельно) разбить ее на подзадачи, каждая из которых будет так или иначе связана с уже разобранными задачами (аналогиями). В дальнейшем в ходе изучения дисциплины, по мере возрастания сложности и объема, разбиение становится более сложным. В этом случае ряд подзадач разбиваются на еще более простые подзадачи, для решения каждой из которых и затем объединения алгоритмов в единый алгоритм для решения данной задачи потребуется рассмотрение ранее изученных аналогий, разбиение известных алгоритмов также на части, синтез разных частей и подчастей одного алгоритма или разных.

Это возвращает нас к изначальному вопросу — **тесной связи различных наборов курсов**, описанных выше. Набор «Математическая логика и теория алгоритмов» объясняет студентам глобальность понятия «алгоритм», дает понятие о том, что вся действительность так или иначе работает по неким алгоритмам, и о том, что написание алгоритма — важнейшая часть написания программы (тем самым важнейшая составляющая решения задачи). Логика помогает избежать в дальнейшем проблем с интерпретацией данных, обобщением, сравнением, противопоставлением данных. Практические задания (не приводимые преподавателем в качестве примера, а предлагаемые студентам для решения сначала интерактивно, а затем самостоятельно) в идеале должны начинаться сразу со «сложных» (составных) ситуаций, требующих четкого составления алгоритма, разбиения на подзадачи. При этом на начальных этапах обучения это разбиение должно быть достаточно очевидным, а компоновка из аналогий и их элементов — достаточно несложной. По ходу дисциплины усложняются задачи, разбиение на подзадачи становится менее очевидным и при решении каждой подзадачи (и тем самым всей задачи) студент пользуется уже не данными преподавателем образцами, а собственными, ранее написанными алгоритмами и кодами. Фактически *эта технология позволяет учащимся приблизиться к той форме организации работы, которая имеет место в компаниях по производству программных продуктов*.

Опыт учебных занятий по программированию позволил нам систематизировать **типичные ошибки в процессе обучения будущих бакалавров**. Так, на начальном этапе обучения студенты зачастую закрывают глаза на coding style, стиль написания программы (отступы, названия переменных) — в их

понимании это мелочи. Дело в том, что на начальном этапе обучения программированию ряд студентов программируют по принципу «главное, чтобы работало» и, уже хорошо освоив написание программ, небрежно оформляют код, что негативно сказывается на профессиональной компетентности и дальнейшем трудоустройстве.

Один из подходов к обучению строится на написании достаточно коротких, несложных программ, в которых соблюдаются условия задания, но при этом программа является чисто утилитарной для учебного процесса (не несет какого бы то ни было осмысленного функционала). Плюсом такого подхода являются разумные временные затраты, незагруженный код, меньшая вероятность ошибок (особенно на ранних этапах обучения). Минусом — отсутствие внимания к функционалу программы.

Другой подход — написание объемных программ с обширным функционалом, таким, что программу при должной корректировке и усовершенствовании можно применять в реальной жизни (разумеется, при этом увеличивается объем кода). Плюсом данной стратегии являются понимание необходимости функционала программы, восприятие программы как алгоритма, для написания которого также используются алгоритмы. Фактически это подход к программированию как к методу решения целого ряда задач. Минусом же становятся увеличение временных затрат студента при проработке алгоритма и написании кода, повышение вероятности ошибок, ошибок в основном в «функциональных» подпрограммах и подзадачах.

При **обучении программированию с предварительным курсом «Математическая логика и теория алгоритмов»** легче объяснить широту применения алгоритмов в жизни. Так, можно представить практически любой процесс действительности как алгоритм. Тем самым удобно пояснить учащимся «правильно» и «неправильно» работающую программу.

Как известно, высшее образование закладывает фундамент для более широкого взгляда на предметную область, чем иное образование. Это особенно важно при обучении программированию, поскольку сложность и многогранность встающих перед специалистами задач требуют самых широких знаний во многих областях. Ремесло программирования, позволяющее в совершенстве освоить язык и ряд инструментов, отступает перед изменчивостью и развитием современного рынка труда, где от специалистов требуется все больше и больше умений и, прежде всего, умение использовать совершенные инструменты совершенным образом. Математическая логика и теория алгоритмов помогают сформировать наиболее фундаментальный взгляд на информационные процессы любой сложности, на сущность алгоритмического подхода к решению задач, особенно в таких передовых областях, как искусственный интеллект, нечеткая логика и эвристика.

В заключение хотелось бы отметить, что многие IT-компании с некоторых пор активно влияют на процесс обучения своих будущих сотрудников. Раньше это касалось в первую очередь студентов-старшекурсников и будущих магистров, через создание института интернатуры и стажировки. Теперь все чаще говорят о подготовке заинтересованных и талантливых учащихся еще со школьной скамьи. В вузах открываются специальности, курируемые компаниями, а трудоустройство происходит еще во время обучения. Такие студенты, закончив вуз, становятся параллельно и полноценными специалистами, обладающими знаниями и опытом для работы в компании.

Литературные и интернет-источники

1. *Гапотченко Д.* Чему научит семья и школа // Computerworld Россия. 2015. № 1–2.

2. *Гончарова О.* Государство, планируя увеличить число IT-специалистов, осознанно закрывает глаза на их профессионализм // Ведомости. 24.11.2014. http://www.vedomosti.ru/career/news/36360621/professionaly-idut-lesom

3. Обзор заработных плат и тенденций российского рынка труда в сфере информационных технологий. Прогнозы на 2014 год // Luxoft Personnel. http://www.luxoft-personnel.ru/press/research/itanalysisrussia/

НОВОСТИ

TNS изучила популярность электронных денег в России

TNS Россия изучила, насколько платежные сервисы популярны у отечественных пользователей. Как выяснилось, многие россияне в буквальном смысле носят электронный кошелек с собой: 40 % пользователей электронных денег платят ими как через компьютер, так и через мобильное устройство.

Исследование проводилось в апреле—мае этого года. В нем участвовали экономически активные пользователи в возрасте от 20 до 44 лет из городов с населением от 700 тыс. человек. Про электронные деньги знают абсолютно все участники опроса, 73 % респондентов пользуются ими минимум раз в год, а почти 50 % — раз в месяц и чаще. Возраст около половины плательщиков — от 25 до 34 лет.

Самыми популярными категориями платежей из электронных кошельков являются интернет-покупки, сотовая связь, денежные переводы, коммунальные услуги и цифровой контент. С компьютеров чаще всего россияне оплачивают онлайн-покупки, а со смартфонов — мобильную связь.

Молодежь от 20 до 24 лет при этом активнее всех использует электронные деньги для покупки билетов на концерты и оплаты онлайн-контента. Пользователи же в возрасте от 35 до 44 лет чаще других платят ими за мобильную связь и коммунальные услуги.

Как показало исследование, с помощью «Яндекс. Денег» не реже чем раз в год платят 44 % интернет-пользователей в возрасте от 20 до 44 лет, проживающих в крупных городах. В свою очередь, через WebMoney то же самое делают 43 % россиян, через Qiwi — 36 %, через PayPal — 35 %.

(По материалам CNews)