

ЧАСТЬ III. ИННОВАЦИОННЫЕ ТЕХНОЛОГИИ В СОВРЕМЕННОМ ОБРАЗОВАТЕЛЬНОМ ПРОЦЕССЕ

Е.В. Болгарина
(РГППУ, Екатеринбург)

НЕОБХОДИМОСТЬ КОРРЕКТИРОВКИ МЕТОДИКИ ОБУЧЕНИЯ ПРОГРАММИРОВАНИЮ

В 2013 г. российская высшая школа отмечала 50-летний юбилей первого выпуска профессиональных программистов. Казалось бы, за полвека дисциплина сформировалась и устоялась, отобрано содержание курсов, определены методологические подходы, имеется методическая база. Тем не менее, вопросы методики обучения программированию как никогда актуальны. В чем причины?

Первая причина диктуется изменившимися требованиями рынка труда и появлением новых профессий. Программирование всегда считалось технической дисциплиной. Исторически сложилось так, что подготовка первых программистов началась на математических факультетах университетов, и долгое время на программирование смотрели, как на специфический раздел математики. Это объясняется тем, что первые программы решали узкоспециализированные задачи научно-технического характера. Целевая направленность и уровень начальной подготовки студентов определяли отбор содержания и методику обучения программированию.

Но вычислительные машины и, чуть позднее, компьютеры очень быстро заняли, а правильнее было бы сказать, захватили лидирующее место во многих отраслях, далеких от научных и технических расчетов.

Анализ исследований, проводимых ведущими рекрутинговыми компаниями, такими как HeadHunter, IT-работа, подразделение Роскомтруд и др., показывает, что в настоящее время одними из заметных трендов на рынке труда стали так называемые «двойные профессии» – сегодня по-прежнему высоко востребованы программисты, но с дополнительными, более узкими компетенциями. Например, сотрудник должен сочетать функции

программиста и 3D-дизайнера, программиста и менеджера продукта и т. д. Аналитик известной российской компании интернет-рекрутмента HeadHunter считает, что «эта тенденция не нова; такие волшебники востребованы всегда и везде, а особенно в регионах и в небольших стартапах» [1]. В такой быстро меняющейся сфере, какой являются информационные технологии, профессионалы не могут замыкаться на одной специализации. Как отмечает эксперт, «в любой сфере востребованы люди, которые могут эффективно решать актуальные рабочие задачи. А для этого, как правило, требуется разбираться и в смежных разделах» [1]. По мнению директора по распространению технологий «Яндекса» Г. Бакунова, граница между дизайнерами и программистами постепенно стирается. Это связано, с одной стороны, с возрастающими требованиями к интерактивности веб-приложений, а с другой – с упрощением программирования. В то же время, с увеличением количества высококонкурентных веб-продуктов, становится все сложнее выделиться на общем фоне. Сегодня недостаточно создать красивый продукт – приложение должно быть наделено уникальным функционалом, а для этого необходимо сочетать знания и навыки дизайнера и программиста [2]. В связи с подобными требованиями появляются направления подготовки, сочетающие в себе дизайнерскую и математико-программистскую подготовку, например, направление подготовки бакалавров «Информационные технологии в медиаиндустрии». Подразумевается, что одной из сфер профессиональной деятельности выпускника будет разработка мультимедийных приложений. Абитуриент, поступающий на данное направление, зачастую не имеет необходимой базовой подготовки в области математики и программирования, а главное, он не мотивирован на их упорное изучение. Разумеется, что в такой ситуации привычные методики обучения программированию не срабатывают.

Вторая причина актуальности корректировки содержания и методики обучения программированию кроется в предъявлении новых требований к выпускнику в связи с введением двухуровневой системы высшего образования и ее компетентностной ориентацией. Сегодня не просто сократился срок подготовки бакалавра, но изменилась ее целевая направленность, появилась ориентированность на формирование практических компетенций. Выпускники должны иметь практические навыки разработки программных продуктов с широкими функциональными возможностями, удовлетворяющими многообразные потребности пользователей.

Сфера образования должна готовить специалистов, которые могли бы приступить к практической деятельности сразу после получения диплома. Но при этом обучение должно быть поставлено таким образом, чтобы IT-специалисты в сфере медиаиндустрии обладали фундаментальными знаниями, которые позволяли бы постоянно повышать свою квалификацию, так как ситуация в сфере информационных технологий постоянно развивается.

Для подготовки бакалавров по направлению «Информационные технологии в медиаиндустрии», обладающих необходимыми компетенциями в области программирования, следует отойти от привычных шаблонов обучения программированию, внести новое и в содержание, и в методику, подобрать более подходящие инструментальные среды.

Исторически сложилось так, что обучение программированию повторяет эволюционный путь развития самого программирования: изучение теоретических основ информатики, дискретная математика и алгебра логики, создание консольных приложений, написанных в структурном ключе, изучение объектно-ориентированного подхода. Чаще всего для обучения используются классические языки типа *Паскаль* или *Cи*, а до современных технологий разработки, например, с применением Agile-методов, студент может так и не добраться. Выход из сложившейся ситуации может быть найден за счет использования иных, новых языков программирования и современных сред разработки.

Третья причина, требующая модификации содержания и методики обучения, продиктована появлением новых инструментальных сред для разработки программных продуктов. Вопрос выбора языка программирования для начального обучения не такой простой. Э. Дейкстра в своей книге «Дисциплина программирования» отмечал, что «наиболее незаметным свойством любого инструмента является его влияние на формирование привычек людей... Когда этот инструмент – язык программирования, его влияние, независимо от нашего желания, сказывается на нашем способе мышления» [3, с. 9]. Важно уже на первых этапах обучения привить определенную культуру программирования, показать современные подходы к решению задачи. Объектно-ориентированная парадигма должна определять стиль мышления бакалавра ИТ, стать основной методологией, определяющей идеологические подходы к анализу и проектированию программных средств.

Масштабность и сжатые сроки выполнения современных проектов на первый план ставят вопросы *технологичности* разработки. Объектно-ориентированное и компонентное программирование являются стандартом для разработки конкурентоспособного программного обеспечения. Любая программа – обработка данных, и почти всегда в реальной ситуации это данные сложной структуры. Особенно сказанное относится к программированию мультимедийных приложений, включающих в себя интерактивную работу с графикой, звуком и видео.

Введение понятий «объект», «класс» в самом начале обучения программированию поможет студентам избежать «стрессовой ситуации ломки мышления», которая практически всегда возникает при переходе от структурного стиля написания программ к объектно-ориентированному. Современные программные системы проектируются и разрабатываются как системы, построенные на классах. Необходимо, чтобы с первых проектов,

создаваемых студентами, формировался стиль, присущий профессиональному программированию, и та степень ответственности, которая отличает профессионального программиста.

Автор данной статьи предлагает использовать для обучения программированию язык *C#* и интегрированную среду разработки *Visual Studio 2010 (2012)*. Практический опыт показал, что эти язык и среда являются отличным решением не только для профессиональной разработки приложений, но и для целей обучения: язык строг и логичен, востребован на рынке профессиональной разработки программного обеспечения (ПО), перспективен и активно развивается; среда многофункциональна и, что немаловажно, бесплатна (версия *Express*), может быть скачана с сайта разработчика – компании «Майкрософт».

Microsoft Visual Studio — линейка продуктов компании «Майкрософт», включающих интегрированную среду разработки ПО и ряд других инструментальных средств. Данные продукты позволяют разрабатывать как консольные приложения, так и приложения с графическим интерфейсом, в том числе с поддержкой технологии *Windows Forms*, а также веб-сайты, веб-приложения, веб-службы как в родном, так и в управляемом кодах для всех платформ, поддерживаемых *Microsoft Windows, Windows Mobile, Windows CE, NET Framework, Xbox, Windows Phone, NET Compact Framework* и *Microsoft Silverlight* [3].

Visual Studio включает в себя редактор исходного кода с поддержкой технологии *IntelliSense* и возможностью простейшего рефакторинга кода. Встроенный отладчик может работать как отладчик уровня исходного кода, так и отладчик машинного уровня. Остальные встраиваемые инструменты включают в себя редактор форм для упрощения создания графического интерфейса приложения, веб-редактор, дизайнер классов и дизайнер схемы базы данных. *Visual Studio* позволяет создавать и подключать сторонние дополнения (плагины) для расширения функциональности практически на каждом уровне, включая добавление поддержки систем контроля версий исходного кода, например, *Subversion* и *Visual SourceSafe*). Имеются возможности для добавления новых наборов инструментов, например, для редактирования и визуального проектирования кода на предметно-ориентированных языках программирования или инструментов для прочих аспектов процесса разработки программного обеспечения, например, клиент *Team Explorer* для работы с *Team Foundation Server*.

Несомненным достоинством данной среды является возможность установки бесплатных *Express*-решений с официального сайта разработчика, компании «Майкрософт», на домашние компьютеры обучаемых. *Express* версии не предъявляют чрезмерных требований к техническим компьютеру и устойчиво работают под операционными системами *Windows XP* и *Windows 7*. Весьма привлекательно и решение компании «Майкрософт»

по предоставлению академических лицензий для учебных заведений. Язык С# является языком со строгой типизацией, полностью базирующийся на объектно-ориентированном подходе к созданию программ. Эти особенности языка позволяют преподавателю строить обучение, изначально опираясь на объектную парадигму, вводя в терминологию дисциплины такие понятия, как классы, объекты, методы уже с первых занятий. Сегодня ни один серьезный проект на С# не обходится без обмена данными с внешними источниками – базами данных, Интернет и прочими хранилищами данных. Использование технологии ADO.NET для взаимодействия с источником данных SQL-Server, а также введение в язык инструмента, получившего название LINQ (Language INtegrated Query) позволит студентам освоить технологию создания полноценных современных программных продуктов. Многочисленные достоинства языка С# позволяют расценить его как перспективную замену языков Паскаль, BASIC и С++ при обучении программированию как студентов, так и школьников старших классов. Язык С# как средство обучения программированию обладает рядом несомненных достоинств.

В Российском государственном профессионально-педагогическом университете изучение дисциплин, связанных с программированием, построено на базе Visual Studio 2010. Результаты обучения, отзывы выпускников, успешно адаптировавшихся в компаниях, ведущих разработку программных продуктов, позволяют сделать вывод о перспективности использования языка С# и среды разработки Microsoft Visual Studio в учебном процессе вузов.

Сфера профессиональной деятельности бакалавров направления подготовки «Информационные технологии в медиаиндустрии» чрезвычайно многообразна, сложна и динамична. Без применения программного кода невозможны интерактивные мультимедийные приложения, обработка потокового видео и аудио, интерактивное телевидение, а также виджеты и гаджеты, которыми насыщен сегодня мир IT. Поэтому вопросы корректировки содержания обучения программированию и методики формирования *профильно специализированной* компетентности бакалавров остаются остро актуальными.

Библиографический список

1. Дефицит IT-специалистов // Центр современных технологий [сайт] URL: <http://www.cst.vbg.ru/node/143> (дата обращения 28.09.2013).
2. Рынок труда IT-специалистов: обзор 2011 года и тенденции 2012-го [сайт] // HR планета. 2012. URL: <http://planetahr.ru/publication/4841> (дата обращения 28.09.2013).
3. Дейкстра Э. Дисциплина программирования. М: Мир, 1978.