

И. Д. Камскова, А. А. Штанюк,

Национальный исследовательский Нижегородский государственный университет им. Н. И. Лобачевского

ПРЕПОДАВАНИЕ ОСНОВ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ В СПО: ПРОБЛЕМЫ, РЕШЕНИЯ, ОПЫТ

Аннотация

В статье представлен опыт преподавания дисциплины «Основы алгоритмизации и программирования» учащимся СПО по специальности 230401 «Информационные системы (по отраслям)»: обоснован выбор первого изучаемого языка программирования и предложены рекомендации по организации образовательного процесса.

Ключевые слова: основы алгоритмизации и программирования, языки программирования, самостоятельная работа студентов, электронные образовательные ресурсы, социальные сети, системы контроля версий.

Преподавание основ алгоритмизации и программирования в наше время является распространенным явлением. Уже в восьмом классе школы на уроках информатики изучаются основы алгоритмизации, базовые понятия информатики, а в старших классах — основы программирования. Для учащихся технических специальностей среднего профессионального образования особенно важно изучение данных областей компьютерных наук, поскольку именно они закладывают базовые знания, применяемые для разработки программного обеспечения.

В данной статье обобщен опыт преподавания дисциплины «Основы алгоритмизации и программирования» учащимся СПО по специальности 230401 «Информационные системы (по отраслям)» в Национальном исследовательском Нижегородском государственном университете им. Н. И. Лобачевского, а также предложены рекомендации по выбору первого языка программирования для обучения и методы преподавания данной дисциплины.

Вопрос о выборе первого языка при обучении программированию является предметом дискуссий преподавателей и представителей ИТ-индустрии

на протяжении многих лет. В этой дискуссии преподаватели зачастую выступают консерваторами и ратуют за сохранение традиционного подхода, а представители промышленного программирования — сторонниками прогресса, непрерывно меняя традиционные подходы на новые, современные.

Нужно отметить, что сам по себе язык не самоцель, а важное средство практической деятельности будущих специалистов, и от правильного его выбора будут зависеть качество образования, интерес к специальности и легкость освоения новых языков. Анализируя существующие языки высокого уровня, которые, с одной стороны, являются не слишком сложными с точки зрения изучения, но, с другой стороны, имеют логически понятные описания операций, в качестве первого изучаемого языка мы бы рассмотрели: Pascal (одну из версий), C++, Visual Basic и C#.

Рассмотрим особенности каждого из языков с точки зрения их использования в качестве первого изучаемого языка программирования в предмете «Основы алгоритмизации и программирования» для учащихся СПО по специальности «Информационные системы (по отраслям)».

Контактная информация

Камскова Инна Дмитриевна, канд. экон. наук, доцент, доцент кафедры математических и естественнонаучных дисциплин Национального исследовательского Нижегородского государственного университета им. Н. И. Лобачевского; *адрес:* 603950, г. Нижний Новгород, пр-т Гагарина, д. 23; *телефон:* (831) 245-54-11; *e-mail:* kamskovaid@mail.ru

Штанюк Антон Александрович, канд. тех. наук, доцент, доцент кафедры программной инженерии Национального исследовательского Нижегородского государственного университета им. Н. И. Лобачевского; *адрес:* 603950, г. Нижний Новгород, пр-т Гагарина, д. 23; *телефон:* (831) 245-54-11; *e-mail:* ashtanyuk@gmail.com

I. D. Kamskova, A. A. Shtanyuk,

National Research Lobachevsky State University of Nizhni Novgorod

TEACHING THE BASICS OF ALGORITHMS AND PROGRAMMING IN COLLEGES: PROBLEMS, SOLUTIONS, EXPERIENCE

Abstract

The article presents the experience of teaching "The basics of algorithms and programming" course for college students on specialty "Information systems". The choice of the first programming language is justified. Recommendations for the organization of the education process are presented.

Keywords: basics of algorithms and programming, programming languages, independent work of students, electronic educational resources, social networks, version control systems.

Итак, Basic стал одним из первых языков программирования, используемых для обучения (60-е годы XX века). Затем появился Pascal (70-е годы), который очень быстро приобрел популярность как у программистов в качестве среды разработки, так и у преподавателей информатики в качестве среды, используемой для обучения. «Это было связано с тем, что он вырос из <...> транслятора, предназначенного только для обучения, в полноценный язык программирования, на котором многие стали писать профессиональные программные продукты. Basic не смог в полной мере конкурировать с <...> Pascal только по одной причине — это был не структурированный язык» [4]. Поэтому в то время Pascal стал ведущим языком для обучения основам программирования.

Вскоре наступила эра объектно-ориентированных языков (80-е годы), появились Object Pascal и C++, затем они были дополнены возможностью использовать визуальные компоненты при разработке программ: для Object Pascal это Delphi, для C++ — Visual C, для Basic — Visual Basic и VBA (Visual Basic for Application).

Визуальные компоненты значительно упростили процесс создания программ, но проблемы обучения основам алгоритмизации и программирования остались. В определенном смысле использование визуальных компонент с первых занятий по программированию даже наносит вред, поскольку у учащихся формируется неверное представление об архитектуре программ. Визуальная логика, логика графического интерфейса заслоняют логику алгоритмическую. Вот почему большинство преподавателей при изучении *основ* программирования не используют эти возможности, а применяют консольный интерфейс и процедурный подход в качестве базовых на начальных этапах изучения процесса разработки программ.

Исходя из нашего опыта преподавания предмета «Основы алгоритмизации и программирования» для учащихся СПО, мы присоединяемся к мнению, что использование визуально-ориентированного подхода на начальных стадиях изучения программирования может быть очень опасным для восприятия профессиональной деятельности учащимися. Ведь визуальное программирование — это в первую очередь технология, облегчающая жизнь *разработчикам*, и к необходимости ее использования важно прийти в результате планомерного развития, усвоения знаний и опыта в разработке собственных программ. Кроме того, большую осторожность нужно проявлять в деле использования объектно-ориентированного подхода, который также можно отнести к продвинутым технологиям программирования, и его неправильное использование начинающими может принести вред формированию профессионального мышления.

Особую ожесточенность приобрели дискуссии между сторонниками и противниками использования C++ в качестве первого языка для изучения основ программирования. Данный язык является не только одним из самых распространенных «промышленных» языков, но и своеобразной «латынью» для большинства используемых сегодня в программировании. Для начинающих, однако, его нельзя рекомендовать, поскольку язык требует достаточно высокого порога вхождения и ручного управления

памятью, что при исследовании алгоритмических проблем создает ненужные трудности.

Язык VBA (Visual Basic for Applications — Visual Basic для приложений) — это язык программирования от Microsoft, встроенный во множество программ — приложения Microsoft Office, Microsoft Project, Visio, AutoCAD, специализированные приложения для управления производством, учета финансов и многого другого. Разработчики VBA анонсируют его как будущий стандарт в индустрии создания приложений, что весьма спорно, на наш взгляд.

Преимущество использования данного языка в том, что при работе с ним пользователю не нужно дополнительно устанавливать на свой компьютер новую инструментальную среду разработки — в Microsoft Office, который установлен практически на всех компьютерах российских пользователей, VBA интегрирован.

Но в качестве первого языка программирования мы бы его не стали рекомендовать, поскольку язык носит ярко выраженный прикладной характер, связанный на решение офисных задач (управление таблицами, графиками, документами), а также основан на устаревших и мало где применяющихся синтаксических конструкциях Basic.

От Basic перейдем к Pascal. В нашей стране в большинстве школ и учебных заведений СПО первым языком программирования до сих пор остается Pascal, несмотря на то что многим он кажется старомодным, невостребованным. Большое количество учебной литературы и примеров в Интернете позволяют подобрать к решению любой алгоритмической задачи. Серьезными недостатками Pascal могут выступать синтаксические несоответствия между популярными современными диалектами (версиями), такими как Free Pascal, Object Pascal, Pascal ABC.Net и пр. Но, как показывает наш опыт преподавания, современные студенты легко справляются с данной проблемой, самостоятельно изучая и используя особенности незнакомых им версий Pascal.

Наиболее приемлемым является использование среды Pascal ABC.Net, которая является свободно распространяемым программным обеспечением. Таким образом, студенты могут без особых проблем установить данный программный продукт на своих домашних компьютерах. Кроме этого существует возможность онлайн-работы в среде Pascal ABC.Net, что тоже немаловажно при работе, например, с планшета или другого мобильного устройства.

В последнее время много разговоров ведется по поводу использования в качестве первого изучаемого языка программирования C#. Этот относительно молодой язык вобрал в себя опыт использования C++ и Java на протяжении многих лет. Синтаксически он очень похож на C++, но лишен многих его трудностей в лице указателей, ручного управления памятью и множества близких по духу операций, вроде точки, стрелки и двоеточия (используется только точка). Главное же преимущество заключается в том, что программы на C# легче создаются и быстрее отлаживаются в связи с более простым синтаксисом и резким уменьшением потенциальных источников ошибок. Необходимо помнить о том, что C# является объектно-ориентированным языком и требует ис-

пользования классов даже в самых простых задачах (как и в случае с языком Java).

Искусство работы преподавателя должно заключаться в правильном использовании классов на ранних этапах обучения, постепенном использовании составляющих объектного подхода (инкапсуляция, агрегация, наследование, перегрузка операций, полиморфизм). Сложность используемых средств и методов программирования должна соответствовать сложности решаемых задач, поэтому не оправдано использование объектно-ориентированного программирования там, где оно по своей сути не нужно.

Язык C# интересен тем, что с ростом сложности решаемых задач он будет предоставлять все новые и новые средства для их решения. К услугам разработчика богатый набор классов .NET, необходимых для различных прикладных областей, развитая система поддержки исключений, делающая исполнение кода более безопасным. В языке имеются средства поддержки функционального подхода (лямбда-выражения, продолжения-continuations, вывод типов), реализация простых шаблонов проектирования. Достаточно хорошо развиты модульный принцип организации программ, работа с динамическими библиотеками, что делает язык удобным не только для учащихся всех уровней, но и для специалистов. А наличие проекта Mono делает программы кроссплатформенными, что позволяет изучать программирование на операционных системах, отличных от семейства Windows.

Из рассмотренных языков (C++, C#, Pascal и VBA), на наш взгляд, оптимальным является использование C# по причине своего возраста (около 10 лет), поддержки современных концепций и подходов (объектно-ориентированный, функциональный), активного развития и распространенности в мировом сообществе. Важно помнить и об интеграции с другими программными продуктами компании Microsoft.

Учащиеся, получающие среднее профессиональное образование, значительно отличаются от студентов вузов. Предмет «Основы алгоритмизации и программирования» изучается на втором курсе учреждения СПО. Возраст учащихся 16–17 лет — это ранний юношеский возраст, возраст прощания с детством, обретения новых социальных позиций, поиск своего места в жизни. В этом возрасте ребята стремятся к самовыражению и самораскрытию, к общению со сверстниками, к самостоятельности. Все эти особенности необходимо учитывать при общении с учащимися. Желательно, чтобы преподаватель был не просто преподавателем, доносящим свои знания до студентов, а наставником-другом, который интересуется личной жизнью ученика (в рамках разумного, разумеется), может дать совет в некоторых жизненных вопросах и пр. Достаточно интересным, на наш взгляд, является использование в качестве практических заданий не простых математических или алгоритмических задач, а приближенных к реальности практических ситуаций.

При изучении дисциплины «Основы алгоритмизации и программирования» большое внимание должно быть уделено самостоятельной работе обучающихся. Работая в аудиториях, учащиеся успевают разобрать лишь небольшое количество алгоритмов, в то время как их достаточно много. В ходе выполнения домашних самостоятельных за-

даний студенты закрепляют практические навыки написания программ, самостоятельно разбирают новые алгоритмы.

Исходя из этого, хотелось бы подробнее остановиться на организации самостоятельной работы учащихся. В Нижегородском государственном университете используется система дистанционного образования (СДО) Moodle, которая помогает организовывать взаимодействие преподавателя и студентов как во время проведения аудиторных практических занятий, так и при самостоятельной работе обучающихся. При работе с Moodle используются такие интерактивные элементы курса, как электронные лекции, задания с обратной связью, форум, рабочие тетради, чат, wiki [2].

Наиболее удобными и используемыми элементами СДО Moodle в дисциплине «Основы алгоритмизации и программирования» являются интерактивные элементы — рабочие тетради и задания с обратной связью. Эти элементы предназначены для организации индивидуального диалога преподавателя со студентом.

Чаще всего в рамках рассматриваемой дисциплины нами используются задания с обратной связью, которые предусматривают формирование задания со стороны преподавателя и выполнение этого задания студентом. В качестве заданий с обратной связью нами предлагается набор задач, для которых необходимо составить алгоритм решения (в виде блок-схем или программного кода).

Целесообразно предлагать учащимся задачи разного уровня. Часть задач (общего уровня) обязательна для выполнения всеми учащимися. Задачи общего уровня составляют приблизительно 60 % от всего количества. Для более глубокого усвоения материала, более детальной проработки различных алгоритмов, решения сложных задач и, соответственно, получения более высокой оценки нами предлагаются задачи высокого уровня. Они выполняются только теми учащимися, которые хотят глубже изучить предмет и получить высокую оценку.

Задание с обратной связью предполагает:

- получение ответа от студента;
- оценку данного ответа преподавателем;
- комментарии преподавателя.

В Moodle существует четыре вида ответов: вне сайта, в виде текста, в виде файла и в виде нескольких файлов. При создании задания преподаватель определяет, каким образом студент будет представлять результаты выполнения задания. Мы рекомендуем формировать ответ в виде файла или нескольких файлов. При формировании ответа в виде файла студент загружает только один текстовый файл, содержащий коды программ. При ответе в виде нескольких файлов студент, соответственно, может загрузить несколько файлов, как правило, это непосредственно файлы проектов Pascal.

Можно также использовать такой вариант ответа, как ответ вне сайта, который предусматривает получение преподавателем результатов выполнения задания при помощи e-mail или на консультации при личной встрече. Вариант ответа в виде текста в данной дисциплине нами не приветствуется.

Поскольку существует возможность формирования ответа и пересылки его в СДО Moodle в виде

файла/нескольких файлов, то учащийся может пересылать практические задания учебной дисциплины в разных видах — текстовый документ формата MS Word с текстами кодов программ, непосредственно проекты или исполняемые файлы. Таким образом, задание с обратной связью — это очень удобный элемент для изучения дисциплины «Основы алгоритмизации и программирования».

Преподаватель оценивает ответы учащихся по той шкале, которую он сам сформировал при создании задания. Каждая оценка сопровождается комментарием, в котором преподаватель указывает ошибки студента или предлагает другой способ решения задачи. Оценки и комментарии будут видны студенту.

Еще одна особенность задания с обратной связью — возможность для учащегося многократно присылать ответ на решение одной задачи. Такая возможность появляется, если преподаватель при создании задания установит параметр «Несколько попыток». Это имеет смысл при выполнении сложных заданий, которые нуждаются в периодических подсказках или поправках преподавателя.

Таким образом, использование возможностей СДО Moodle необходимо для изучения и закрепления пройденного материала в рамках учебной дисциплины в ходе выполнения практических заданий.

Однако СДО Moodle имеет и недостатки. Большинство современных молодых людей активно используют в своей повседневной жизни и в учебе мобильные устройства: смартфоны и планшеты. «Традиционные СДО, с одной стороны, удобны для полноценного обеспечения учебного процесса, а с другой стороны, студенты, особенно первых курсов, встречая непривычный и строгий интерфейс, воспринимают курс без заинтересованности и энтузиазма. В СДО стиль общения преподавателя и обучающегося, интерфейс среды строгий и неинтересный для современного молодого, избалованного многообразием красивых веб-ресурсов пользователя. Кроме этого преподавателю необходимо осваивать особенности управления контентом системы» [3].

В настоящий момент молодое поколение «живет» в социальных сетях, в частности в отечественной социальной сети «ВКонтакте», которая является самым интересным и популярным ресурсом для молодежи. Отдельные возможности социальной сети «ВКонтакте» можно использовать для организации самостоятельной работы учащихся СПО.

Преимущество здесь в том, что интерфейс социальной сети хорошо знаком учащимся, а также хорошо ими освоен. Молодые люди чувствуют себя на страничках социальной сети более комфортно, чем на страничках систем дистанционного образования.

Как правило, каждая группа студентов создает в сети «ВКонтакте» свою собственную виртуальную группу (сообщество), в которую допускаются только члены реальной студенческой группы. Имея такое сообщество в социальной сети, студенты и преподаватели получают дополнительные преимущества. На страничке группы можно размещать практические задания, методические пособия и пр. Участники группы совместно могут решать задания, выкладывать их решения и обсуждать проблемы, связанные с заданиями. В этом может принимать участие и пре-

подаватель. Преподавателю очень удобно выкладывать результаты проверки заданий и контрольных работ.

В качестве дополняющего Moodle сервиса при изучении программирования можно рассмотреть сетевые площадки для размещения программных проектов GitHub и Bitbucket. Для эффективного использования их студентам необходимо изучить основы работы с таким важным инструментом разработчика, как система контроля версий Git [5]. Изучение Git не только поможет использовать современные сетевые сервисы, но и познакомит с инструментарием, который применяется в промышленной разработке программного обеспечения. В таком случае ссылки на репозитории (сетевые хранилища проектов) могут существенно упростить взаимодействие между преподавателем и студентами при необходимости отчитаться за выполненную работу, получить оценку за написанный код и комментарий. Особо следует отметить такие возможности сетевых сервисов, как совместная работа над проектами, перекрестное рецензирование и инспекция программного кода. Студенты могут получать задания в виде рецензирования кода своих одногруппников.

Существуют и другие программные решения, позволяющие упростить ряд задач учебного процесса, например работу с экзаменационными листами [1].

Итак, подведем итоги. В качестве первого изучаемого языка программирования мы бы рекомендовали язык C# и свободно распространяемую среду Visual Studio Community Edition. Для качественного усвоения предмета, т. е. для получения хороших навыков программирования, мы рекомендуем большое внимание уделять самостоятельной работе учащегося. Для организации самостоятельной работы с целью повышения качества обучения целесообразно использовать не только традиционные системы дистанционного образования, но и новые информационно-коммуникационные технологии: социальные сети вообще и отечественную социальную сеть «ВКонтакте» в частности. Необходимо изучать и анализировать все возможности социальных сетей, которые постоянно совершенствуются и дополняются новыми функциями, и применять их в учебном процессе. Хорошим дополнением к традиционному ПО могут стать системы контроля версий и сетевые площадки для размещения проектов.

Список использованных источников

1. Бондаренко В. В., Козич В. Г., Баженов Р. И. Создание проекта разработки системы учета экзаменационных листов в ganttproject // Постулат. 2016. № 7 (9).
2. Камскова И. Д. Организация самостоятельной работы студентов с использованием электронных ресурсов СДО MOODLE // Ученые записки ИСГЗ. 2014. Вып. 1.
3. Камскова И. Д. Социальные сети в образовании: за и против // Ученые записки ИСГЗ. 2016. Вып. 1.
4. Овчаров С. А. Особенности преподавания программирования бакалаврам бизнес-информатики // Известия Волгоградского государственного технического университета. 2013. Т. 14. № 16.
5. Штанюк А. А. Системы управления версиями при изучении программирования // Международное научное издание «Современные фундаментальные и прикладные исследования». 2015. № 3.