

УДК 33

DOI: 10.34670/AR.2020.92.10.039

Экономические аспекты реализации сценарием обучения программированию в телекоммуникационной среде

Ратанова Ольга Валентиновна

Старший преподаватель,
Московский финансово-промышленный университет «Синергия»,
125190, Российская Федерация, Москва, просп. Ленинградский, 80;
e-mail: ratanova@mail.ru

Аннотация

В статье представлены различные сценарии обучения программированию в телекоммуникационной среде. Автор отмечает, что программирование занимает важное место в системе подготовки обучающихся. Система обучения программированию основана на интеграции различных принципов, разных парадигм программирования. Обучение школьников и студентов различным принципам программирования позволяет сформировать умение моделировать и проектировать предметную область разными стилями, основанными на разных парадигмах программирования. При изучении различных принципов программирования студенты познают особенности использования различных методов и принципов структуризации и обработки информации. Сделан вывод о том, что при обучении программированию в системе, поддерживающей автоматизированное обучение, должна быть предусмотрена возможность погружения теоретических материалов по изучаемым дисциплинам с примерами различных тестов, а также заданий на написание программного кода. Кроме того, должно быть предусмотрено ведение журналов прохождения тестов и написания программ. Очень полезна была бы система обобщения информации о пользователе, т.е. сбор информации о прохождении одним студентом тестов по различным дисциплинам. Очевидно, что было бы выгодно использовать такую систему в уже имеющихся онлайн-курсах по программированию.

Для цитирования в научных исследованиях

Ратанова О.В. Экономические аспекты реализации сценарием обучения программированию в телекоммуникационной среде // Экономика: вчера, сегодня, завтра. 2019. Том 9. № 10А. С. 330-337. DOI: 10.34670/AR.2020.92.10.039

Ключевые слова

Программирование, обучение, сценарии, телекоммуникационная среда, учебный курс, Интернет, пользователь.

Введение

В настоящее время в университетах и других образовательных организациях создано большое количество автоматизированных учебных курсов по различным областям знаний, ориентированных на работу в среде Intranet/Internet. Большинство из них строятся по одной схеме и имеют сходные возможности. В общем случае такие курсы выполняют предоставление обучаемому материала в виде гипертекстового справочника; регистрацию пользователей в системе; сохранение информации о пользователе в базе данных на сервере; идентификацию зарегистрированного пользователя в системе; прохождение теста; выдачу результатов прохождения тестов и запись результатов на сервер.

Специфика обучения программированию такова, что большое значение для освоения курса имеет практическое написание работающего кода на языке программирования, то есть получение работающей программы. В программировании недостаточно изучить теоретические основы языка программирования. Чтобы приобрести навык программирования, необходимо писать реальные действующие программы, прохождения тестов для получения данного навыка недостаточно. Современные тесты предоставляют обучаемым выбор одного правильного ответа из нескольких по теоретическим вопросам курса, также в тестах есть анализ небольшого кода, установление соответствия и порядка действий, возможно даже написание нескольких строк кода. Тем не менее, этой практики недостаточно для освоения языка программирования.

Основная часть

В настоящее время разработаны программы верификации программного кода, что позволяет автоматически проверять написанный обучающимся программный код без привлечения для проверки кода живого преподавателя. Используя такие системы в курсах по обучению программированию, можно повысить их эффективность и качество, исключая при этом участие преподавателя. Это делает дистанционное обучение программированию более доступным и эффективным.

Системы регистрации, идентификации и тестирования таких курсов могут быть написаны на различных языках веб-программирования и в различных базах данных для хранения информации, от простых текстовых файлов до такой мощной системы управления базами данных (СУБД), как MS SQL. В каждом из таких курсов, как правило, разработана своя система регистрации и друг с другом они никак не связаны, поэтому обучаемому надо регистрироваться каждый раз по-новому.

Таким образом, при создании автоматизированных курсов в виде информационных ресурсов, эксплуатируемых в среде Интернет, возникает ряд проблем, разрешение которых труднодостижимо в рамках технологии создания традиционных HTML-документов.

Во-первых, для обеспечения малой предсказуемости вопросов в контрольных разделах курсов и защищенности эталонных ответов от прочтения пользователем автору учебного курса приходится заниматься достаточно сложным программированием как серверной, так и клиентской части системы тестирования курсов.

Во-вторых, статистика успешности обучения студента ведется автономно для каждого курса, что приводит не только к лишним затратам труда разработчика, но и усложняет задачу интеграции данных по нескольким курсам с целью получения суммарного рейтинга успеваемости.

В-третьих, эксплуатация большого количества курсов на каком-либо сервере сопряжена со значительными затратами времени на администрирование в силу того, что многие вопросы управления регистрационными данными об обучаемых, данными о ходе обучения в различных системах решаются по-разному либо вообще никак не решаются [Цветкова, Великович, 2012].

Сегодня существуют системы, позволяющие авторам курсов погружать свои тесты в уже готовые системы. Однако эти системы удобны по большей части для однократного проведения тестов, а не для системного их использования. Кроме того, такие системы хоть и решают вопросы предсказуемости и защищенности эталонных ответов, но при этом создают проблемы интеграции результатов и, как правило, имеют свою систему регистрации пользователей. Выдать ссылку на тест для группы учащихся и обработать их ответы при очной форме обучения, вероятно, будет удобно для преподавателя. Однако, если мы говорим об автоматизированном обучении и автоматизированных курсах, то использование сторонних тестирующих систем оказывается неудобным, несмотря на их возможности.

Возможности систем тестирования в бесплатном варианте обычно сильно урезаны. Есть ограничения по количеству вопросов, например, Easy Test Maker ограничивает не только количество тестов, но и возможность экспорта их в «бумажные» форматы. Системы Plickers и Kahoot! предполагают использование тестов в аудитории на очном занятии. А система Quizlet позволяет загружать картинки только в платном варианте. Система Proprofs в платном варианте позволяет управлять группами учащихся. Таким образом, несмотря на разнообразие существующих систем для создания тестирования, перед создателем автоматизированных курсов стоит серьезная проблема: либо интеграция с существующими системами, либо написание собственной оригинальной системы тестирования [Денисова, Дженжер, 2012].

Если речь идет о написании собственной системы тестирования для автоматизированных курсов, то это решит и вторую проблему, а именно составление статистики успешного прохождения курсов. Во многих университетах в настоящее время существуют собственные системы для ведения посещаемости студентами занятий, успеваемости студентов и т.п. Но такие системы разрабатываются обычно для работы со студентами очного и вечернего обучения, то есть они не поддерживают онлайн-курсы с системой тестирования. Таким образом, при разработке сценариев обучения в среде Интернет нужна система, которая совмещала бы системы учета обучающихся, системы тестирования и интерактивную подачу обучающего материала, присущую онлайн-курсам.

Если рассматривать автоматическую систему верификации и валидации программного кода, то основные программные средства работают с коммерческими программными продуктами большой сложности. Такие средства в большинстве случаев платные, и их использование сопряжено с большими сложностями. Коммерческие средства верификации кода работают на всем жизненном цикле программного продукта и проверяют не только правильность кода, но и безопасность, сопровождаемость, эффективность, переносимость и надежность. При обучении же программированию программные коды, которые пишут студенты на начальном этапе обучения, достаточно простые. Для проверки правильности студенческого кода нужно определить соответствие данных и результат. Для таких проверок неэффективно использование мощных и дорогостоящих средств верификации кода.

Однако существуют и некоммерческие системы верификации программного кода. В открытых источниках можно найти системы формальной верификации. Кроме того, в некоторых университетах при организации олимпиад используют свои системы автоматической верификации программного кода, написанного участниками [Яникова, 2013]. Часто это

применяется именно при заочном этапе олимпиады, когда участников и их программ очень много. Конечно, при использовании автоматической проверки кода требование к этому коду должны быть строго формализованы. Однако для обучения программированию жестко заданные рамки программирования и четко сформулированное задание являются дополнительным плюсом, так как начинающие программисты часто подходят к выполнению заданий слишком творчески, что уводит их в сторону от заданной цели.

Таким образом, основными подсистемами автоматизированного учебного курса при обучении программированию должны быть:

- подсистема авторизации доступа, обеспечивающая регистрацию пользователей различных категорий и контроль доступа к функциям и данным в соответствии с определенными правами;
- подсистема поддержки создания курсов, позволяющая формировать множество кадров и сценарии в диалоговом режиме;
- интерпретатор сценариев с встроенными средствами автогенерации;
- подсистема поддержки создания тестов с поддержкой тестов разного типа;
- подсистема тестирования;
- подсистема формального описания задач для программирования;
- подсистема верификации и валидации программного кода;
- подсистема работы с данными о ходе обучения;

В системе должны объединяться технологии представления учебного материала в базах данных и интернет-технологии доступа к функциям и данным.

Таким образом, при обучении программированию в системе, поддерживающей автоматизированное обучение, должна быть предусмотрена возможность погружения теоретических материалов по изучаемым дисциплинам с примерами различных тестов, а также заданий на написание программного кода. Должна иметься система регистрации пользователей, предусмотрено ведение журналов прохождения тестов и написания программ. Очень полезна была бы система обобщения информации о пользователе, т.е. сбор информации о прохождении одним студентом тестов по различным дисциплинам. Очевидно, что было бы выгодно использовать такую систему в уже имеющихся онлайн-курсах по программированию.

Во что же выльется такая интеграция автоматизированных учебных курсов для создателя контента? Во-первых, необходимо погрузить данные тестов – вопрос, варианты ответов, а также другие типы тестов. При этом если тесты в текстовых файлах, то их погружение может происходить через интерфейс с помощью стандартных средств текстового редактора (через Clipboard). Это не будет трудоемким, если тесты содержат не более 30 вопросов. Если же вопросов больше, необходимым становится создание средства автоматического переноса тестов в базу данных. В случае, если данные тестов хранятся в какой-либо базе данных, придется предусматривать создание специальных программ-скриптов для переброса теста из одной базы данных в другую [Райтман, 2011]. Во-вторых, нужно организовать доступ к погруженным в систему тестам. Так как вызов тестирования представляет собой обычно ссылку в каком-либо месте курса, например в гипертекстовом справочнике по дисциплине, то для вызова тестов следует сделать эту ссылку. Если тесты были в каждом разделе курса, а не по всему курсу в целом, то они погружаются отдельно, как самостоятельные независимые тесты. В-третьих, нужно организовать погружение задания на программирование [Лаврищева, 2018]. Причем задания должны быть максимально формализованы. Возможно, для этого понадобится специализированный интерфейс.

Для загрузки готовых программ от студентов в системе верификации необходимо

предусмотреть интерфейс. Процесс регистрации нового пользователя, его идентификации в системе, организации тестов, обработки и выдачи результатов тестов и проверки программного кода берет на себя система. Таким образом, предполагаемая схема системы обучения программированию со всеми подсистемами, входящими в нее, будет выглядеть следующим образом.

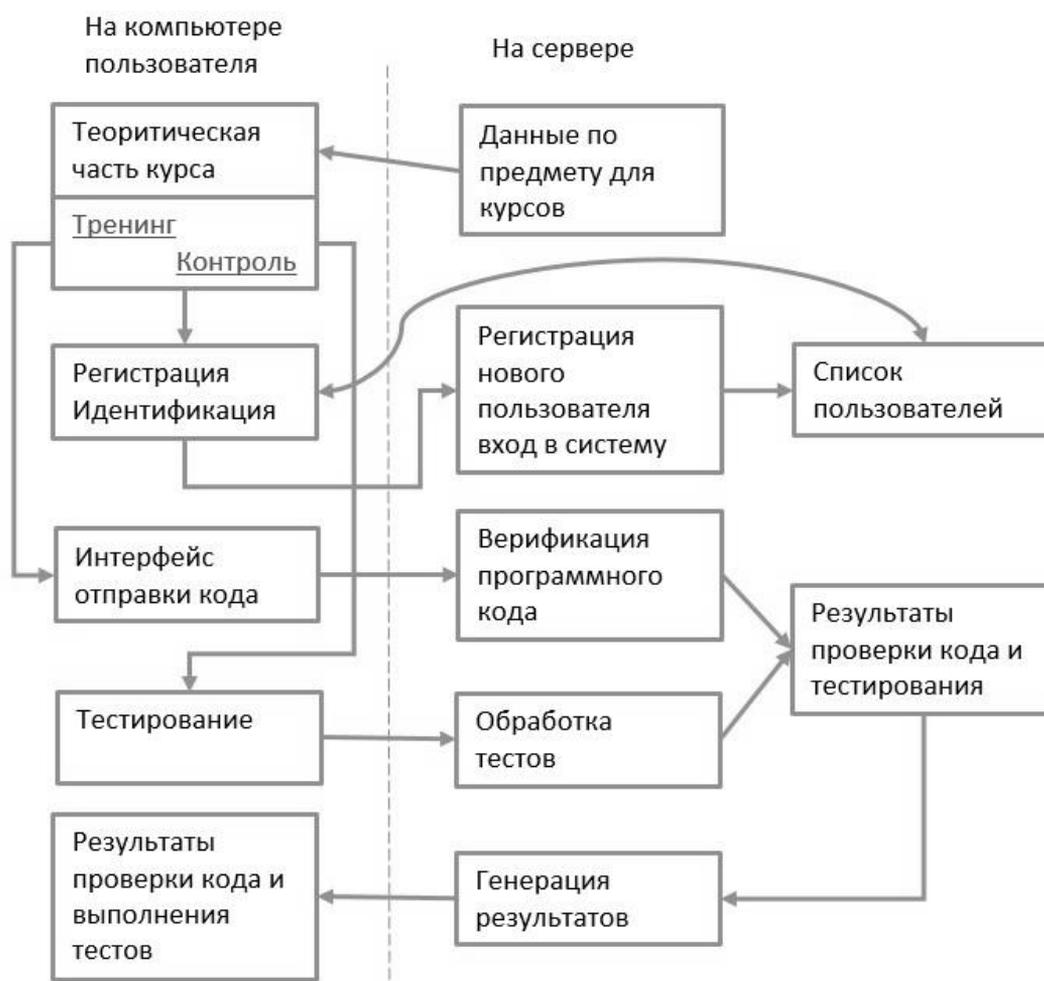


Рисунок 1 - Схема системы обучения программированию

В случае использования автоматизированных курсов в местах, где нет доступа к сети, целесообразно использование варианта курсов, которые студент мог бы получить для работы на своем компьютере, то есть изучать теоретический материал и тренироваться в прохождении тестов без доступа к сети. В этом случае вариант курсов для индивидуального использования не требует системы регистрации и идентификации. В таком автоматизированном курсе остается справочник по теоретическому материалу, система верификации программного кода и тестовая система. Результаты прохождения теста только предъявляются студенту для самоконтроля. В этом случае система тестирования выступает как средство тренинга студента. База данных тестов для простоты эксплуатации на домашнем компьютере, конфигурация которого может быть самой различной, может находиться в текстовых файлах.

Заключение

Программирование занимает важное место в системе подготовки обучающихся. Система обучения программированию основана на интеграции различных принципов, разных парадигм программирования. Обучение студентов различным принципам программирования позволяет сформировать умение моделировать и проектировать предметную область разными стилями, основанными на разных парадигмах программирования. При обучении программированию нужно использовать систему задач, показывающих различие подходов при их решении. При изучении различных принципов программирования студенты познают особенности использования различных методов и принципов структуризации и обработки информации.

В результате проведенного исследования можно сделать следующие выводы. При обучении программированию в системе, поддерживающей автоматизированное обучение, должна быть предусмотрена возможность погружения теоретических материалов по изучаемым дисциплинам с примерами различных тестов, а также заданий на написание программного кода. Должна иметься система регистрации пользователей, предусмотрено ведение журналов прохождения тестов и написания программ. Очень полезна была бы система обобщения информации о пользователе, т.е. сбор информации о прохождении одним студентом тестов по различным дисциплинам. Очевидно, что было бы выгодно использовать такую систему в уже имеющихся онлайн-курсах по программированию.

Библиография

1. Бузунова О.А. Теоретические основы моделирования деятельности компании в неравновесных условиях // *Modern Economy Success*. 2018. №2. С. 45 – 48.
2. Волкова Н.М., Надточий Ю.Б. Новые направления исследований в неэкономике // *Экономические системы*. 2019. Том 12. № 1. С. 23 – 32.
3. Герасимова Ю.В. Современные логистические подходы в организации межфирменного взаимодействия компаний // *Russian Economic Bulletin*. 2018. Том 1. №2. С. 10 – 18.
4. Горбачева А.А., Кормишин А.Е. Актуальные тенденции цифровизации и их влияние на преобразование энергетического сектора экономики России // *Modern Economy Success*. 2019. № 1. С. 46 – 51.
5. Денисова Л.В., Дженжер В.О. Среда Scratch в практике учителя начальной школы // *Начальная школа*. 2012. № 5. С. 31-35.
6. Лаврищева Е.М. Программная инженерия и технологии программирования сложных систем. М.: Юрайт, 2018. 432 с.
7. Лаврищева Е.М., Петренко А.К. Информатика: становление программного обеспечения и технологий программных систем // *Труды Института системного программирования РАН*. 2018. № 30(5). С. 7-30.
8. Мамонова Т.Е. Информационные технологии. Лабораторный практикум. М.: Юрайт, 2019. 178 с.
9. Пейперт С. Переворот в сознании: Дети, компьютеры и плодотворные идеи. М.: Педагогика, 1989. 224 с.
10. Райтман М.А. Action Script 3.0 для Adobe Flash Professional CS5. М.: Эксмо-Пресс, 2011. 220 с.
11. Цветкова М.С., Великович Л.С. Информатика и ИКТ. М.: Академия, 2012. 89 с.
12. Яникова Н.В. Возможности среды Scratch для развития ключевых компетенций учащихся и профессионального роста педагогов // *Информатика и образование*. 2013. № 8 (247). С. 34-36.
13. Dann W., Copper S., Pausch R. Learning to program with Alice. Upper Saddle River. NJ: Prentice Hall, 2006.
14. Liu M., Williams D., Pedersen S. Alien rescue: A problem-based hypermedia learning environment for middle school science // *Journal of Educational Technology Systems*. 2002. No. 30 (3). P. 255-270.

Economy aspects of scenarios of teaching programming in telecommunication environment

Ol'ga V. Ratanova

Senior Lecturer,
Moscow University for Industry and Finance "Synergy",
125190, 80 Leningradskii av., Moscow, Russian Federation;
e-mail: ratanova@mail.ru

Abstract

The article presents various scenarios of teaching programming in the telecommunication environment. The author notes that programming occupies an important place in the system of training students. The programming training system is based on the integration of various principles, different programming paradigms. Training schoolchildren and students in various principles of programming allows forming the ability to model and designing a subject area in different styles based on different programming paradigms. When studying various principles of programming, students learn the features of using various methods and principles of structuring and processing information. When teaching programming, they need to use a system of tasks that show the difference in approaches when solving them. It is concluded that when teaching programming in a system that supports automated learning, it should be possible to immerse theoretical materials in the studied disciplines with examples of various tests, as well as tasks for writing program code. In addition, journals of passing tests and writing programs should be provided. A system for summarizing user information would be very useful, i.e. collecting information on passing tests one by one student in various disciplines. Obviously, it would be beneficial to use such a system in existing online programming courses.

For citation

Ratanova O.V. (2019) Ekonomicheskie aspekty scenariy obucheniya programmirovaniyu v telekommunikatsionnoi srede [Economy aspects of scenarios of teaching programming in telecommunication environment]. *Ekonomika: vchera, segodnya, zavtra* [Economics: Yesterday, Today and Tomorrow], 9 (10A), pp. 330-337. DOI: 10.34670/AR.2020.92.10.039

Keywords

Programming, teaching, scenario, telecommunication environment, training course, Internet, user.

References

1. Buzunova O.A. Teoreticheskiye osnovy modelirovaniya deyatel'nosti kompanii v neravnovesnykh usloviyakh // Modern Economy Success. 2018. №2. S. 45 – 48.
2. Dann W., Copper S., Pausch R. (2006) *Learning to program with Alice. Upper Saddle River*. NJ: Prentice Hall.
3. Denisova L.V., Dzhenzher V.O. (2012) Sreda Scratch v praktike uchitelya nachal'noi shkoly [Scratch environment in the practice of elementary school teacher]. *Nachal'naya shkola* [Elementary School], 5, pp. 31-35.
4. Gerasimova YU.V. Sovremennyye logisticheskiye podkhody v organizatsii mezhfirmennogo vzaimodeystviya kompaniy // Russian Economic Bulletin. 2018. Tom 1. №2. S. 10 – 18.
5. Gorbacheva A.A., Kormishin A.Ye. Aktual'nyye tendentsii tsifrovizatsii i ikh vliyaniye na preobrazovaniye energeticheskogo sektora ekonomiki Rossii // Modern Economy Success. 2019. № 1. S. 46 – 51.

6. Lavrishcheva E.M. (2018) *Programmnyaya inzheneriya i tekhnologii programmirovaniya slozhnykh sistem* [Software engineering and programming technologies for complex systems]. Moscow: Yurait Publ.
7. Lavrishcheva E.M., Petrenko A.K. (2018) Informatika: stanovlenie programmnoogo obespecheniya i tekhnologii programmnykh sistem [Informatics: the formation of software and technology software systems]. *Trudy Instituta sistemnogo programmirovaniya RAN* [Proceedings of the Institute for System Programming of the Russian Academy of Sciences], 30(5), pp. 7-30.
8. Liu M., Williams D., Pedersen S. (2002) Alien rescue: A problem-based hypermedia learning environment for middle school science. *Journal of Educational Technology Systems*, 30 (3), pp. 255-270.
9. Mamonova T.E. (2019) *Informatsionnye tekhnologii. Laboratornyi praktikum* [Information Technology. Laboratory workshop]. Moscow: Yurait Publ.
10. Peipert S. (1989) *Perevorot v soznanii: Deti, komp'yutery i plodotvornye idei* [Coup in consciousness: Children, computers and fruitful ideas]. Moscow: Pedagogika Publ.
11. Raitman M.A. (2011) *Action Script 3.0 dlya Adobe Flash Professional CS5* [Action Script 3.0 for Adobe Flash Professional CS5]. Moscow: Eksmo-Press Publ.
12. Tsvetkova M.S., Velikovich L.S. (2012) *Informatika i IKT* [Informatics and ICT]. M.: Akademiya Publ.
13. Volkova N.M., Nadtochiy YU.B. *Novyye napravleniya issledovaniy v neoekonomike // Ekonomicheskiye sistemy*. 2019. Tom 12. № 1. S. 23 – 32.
14. Yanikova N.V. (2013) *Vozmozhnosti sredy Scratch dlya razvitiya klyuchevykh kompetentsii uchashchikhsya i professional'nogo rosta pedagogov* [Possibilities of Scratch environment for the development of key competencies of students and professional growth of teachers]. *Informatika i obrazovanie* [Computer Science and Education], 8 (247), pp. 34-36.