

УДК 004.4:378.147

В.М. Бондарев, Ю.Ю. Черепанова

Харьковский национальный университет радиоэлектроники, Харьков

СЕТЕВАЯ СРЕДА ДЛЯ ПОДГОТОВКИ И ЧТЕНИЯ ЛЕКЦИЙ

Рассматриваются вопросы проектирования и разработки сетевого приложения, позволяющего создавать презентации лекций по программированию в виде динамических HTML-документов, осуществлять проведение лекций, контроль знаний. Контроль знаний по программированию предлагается проводить путем выдачи условий задач и автоматической проверки решения с помощью сетевой службы. Предложенная система находится в эксплуатации и хорошо себя зарекомендовала.

Ключевые слова: подготовка лекций, обучение программированию, презентация, автоматический контроль знаний.

Введение

Постановка проблемы. Сегодня мы уже привыкли к тому, что преподаватель, читая лекцию, держит в руке не мел, а пульт управления проектором, а студенты смотрят не на доску, а на экран с презентацией Power Point. И это правильно и хорошо, что, впрочем, не означает, что нельзя сделать еще лучше. Любой документ начинается с черновика и становится законченным произведением (например, презентацией) после долгой и не всегда увлекательной работы над ним. Хотелось бы сократить путь от черновых набросков к готовой презентации, не потеряв при этом в качестве результата.

Напомним, что здесь речь пойдет в основном о преподавании программирования с присущими ему особенностями. В частности, постоянно приходится обращаться программному коду, его показу, разбору, и даже выполнению. Часто приходится говорить об алгоритмическом языке, показывать синтаксические схемы и правила. И конечно, максимум внимания приходится уделять решению задач, которые состоят в написании небольших фрагментов программ по заданной спецификации.

Наблюдая за наиболее удачными презентациями на тему программирования, можно заметить, что материал в них подается последовательно, небольшими порциями и таким образом, что очередная порция не уничтожает предыдущую, а добавляется к ней, постепенно вытесняя за пределы экрана то, что уже не требует внимания слушателей. Примерно так же движутся титры на киноэкране, только не непрерывно, а дискретно и, конечно же, в ином темпе.

Понятно, что Power Point при всех его достоинствах не очень подходит для подобных презентаций, и лучшим решением, с нашей точки зрения, является динамический html-документ, который можно продемонстрировать при помощи веб-браузера, благо последний всегда под рукой. Размеры окна и масштаб изображения можно менять как угодно, что бывает необходимо, когда экран компьютера нужно

разделить между презентацией и чем-то еще или просто улучшить видимость презентации. Проблема «всего лишь» в том, что создание таких документов требует основательных знаний html, css, javascript и может увлечь разве что преподавателей веб-дизайна.

Вот тут-то и возникает наше предложение, которое состоит в том, чтобы создавать лишь «черновик» презентации в форме простого текста с минимальной и необременительной разметкой, а умная программа пусть доделает все остальное. Но раз уж мы допускаем программу к непростому процессу подготовки презентаций, на нее можно возложить обязанность хранения черновиков, объединения их в курсы лекций, публикацию лекций в сети и разные прочие заботы, вроде импорта и экспорта презентаций в различные форматы.

Во время собственно презентации докладчик сможет отвечать не только на устные вопросы, но и на письменные, полученные им по сети (возможность задать вопрос, не привлекая к себе всеобщего внимания, не будет лишней для некоторых слушателей). Можно даже предоставить слушателям обратную связь в виде кнопки «Мне не понятно», чтобы преподаватель узнал об этом своевременно, а не на экзамене в конце семестра.

Анализ последних исследований и публикаций. Ближайшим аналогом представленной системы являются образовательные платформы, такие как [1] или [2]. Безусловно, – это замечательные ресурсы, но они ориентированы на массовые открытые онлайн-курсы, не имеют средств достоверного контроля знаний, а изготовление учебных материалов в них требует значительных трудозатрат. Особенностью же предлагаемого решения является минимальное «расстояние» от черновика до полноценной презентации, возможность перманентного рефакторинга презентаций плюс быстрое изготовление задач с автоматической проверкой решения. В целом можно сказать, что отличие данного решения от крупной образовательной платформы такое же, как отличие велосипеда от автомоби-

ля, автомобили нужны, но и в пользу велосипедов никто не сомневается.

Поскольку решение задач есть основа обучения программированию, имеет смысл отдельно рассмотреть аналоги в этой узкой области, т.е. в области задачников с автоматической проверкой решений. Здесь ресурсов значительно больше (поэтому рассмотрим лишь несколько), это системы автоматической проверки решений с архивами задач [3–4] или без них [5] и опционально с возможностью участия или проведения соревнований по программированию [6–7]. Есть также системы, ориентированные на конкретные языки программирования [8–9]. Всех их объединяет направленность на соревнования по программированию, которые требуют алгоритмического мышления, а не знания особенностей того или иного языка программирования. Задачи там абстрагированы от языка и унифицированы по форме – ввод из стандартного потока, решение, вывод результатов в стандартный поток. Это отличает их от задач представленной системы, которые могут варьировать от исправления ошибок в заданном фрагменте кода до определения класса с заданным интерфейсом или одного только метода. Несомненно, это ближе к тому, что требуется для разнообразных упражнений будущего программиста.

Постановка задачи. Для повышения эффективности и качества обучения программированию необходимо разработать сетевое приложение, позволяющее создавать презентации лекций в виде динамических HTML-документов, осуществлять проведение лекций, контроль знаний в виде выдачи условий задач и автоматической проверки решений.

Сейчас мы рассмотрим возможности программы более подробно, но сначала определим несколько необходимых терминов.

Будем называть упоминаемый выше черновик исходным текстом презентации или исходником.

Порции, которыми информация подается слушателям, будем по традиции называть слайдами.

Результат переработки исходника в html-документ назовем ленточной презентацией (лентой), а процесс переработки – компиляцией.

Автора и исполнителя презентации будем называть преподавателем, а слушателей – студентами.

Изложение основного материала

Основной сценарий

Преподаватель авторизуется в системе и попадает на страницу со списком своих уже созданных презентаций (рис. 1).

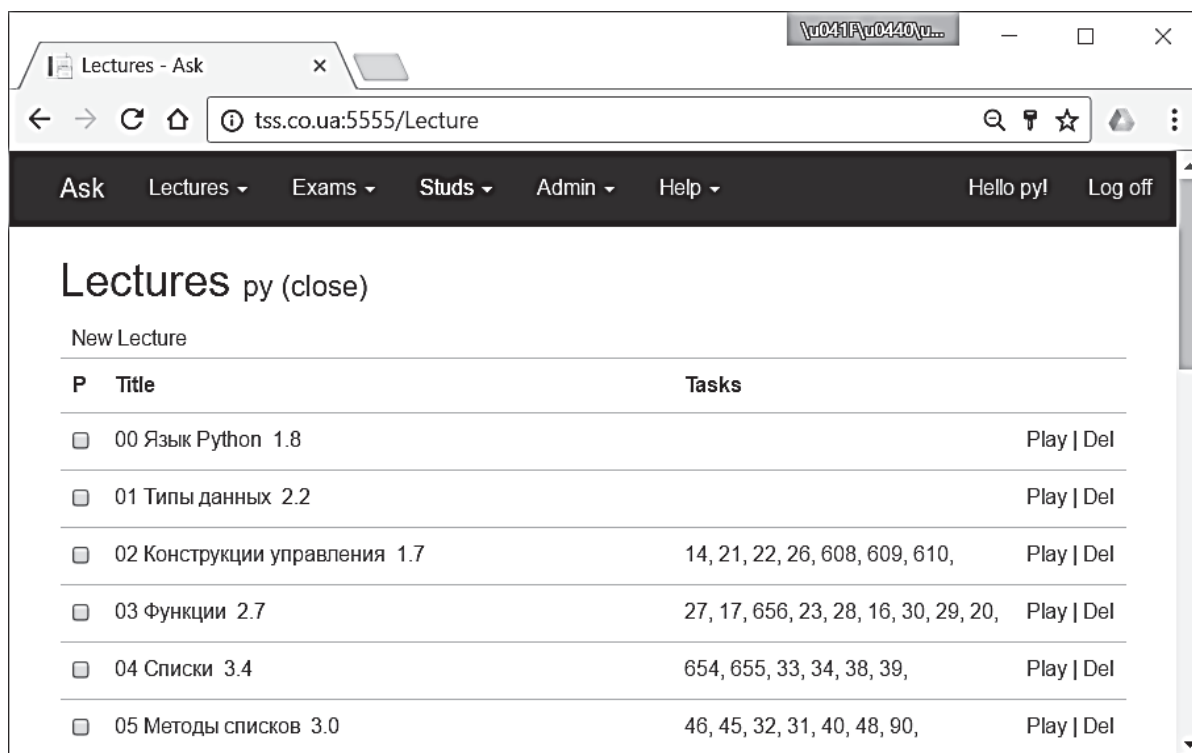


Рис. 1. Домашняя страница. Список презентаций

Там он выбирает презентацию для редактирования или создает новую. Готовые презентации преподаватель может «публиковать» делая их доступными в сети для всех или только для зарегистрированных пользователей, т.е. для студентов.

Когда приходит время читать лекцию, преподаватель выбирает презентацию и запускает режим воспроизведения. При этом открывается тот же HTML-документ, но с возможностью показа его последовательно открывая слайд за слайдом.

Во время лекции студенты могут задавать вопросы как устно, так и письменно, отправляя их со своей студенческой страницы.

Подготовка исходного текста

Работа над исходным текстом презентации происходит в окне редактирования (рис. 2). Результат правки немедленно виден в другом окне, где открывается скомпилированный HTML-документ.

Исходник презентации это простой текст, но маркеры начала слайдов подсвечены для наглядности: красным – слайды заголовков, серым – текстовые слайды, зеленым – слайды задач. Пустые строки между слайдами служат только для читабельности, компилятор презентаций их игнорирует. На рисунке можно заметить, что выполняется проверка орфографии, но это уже заслуга браузера.

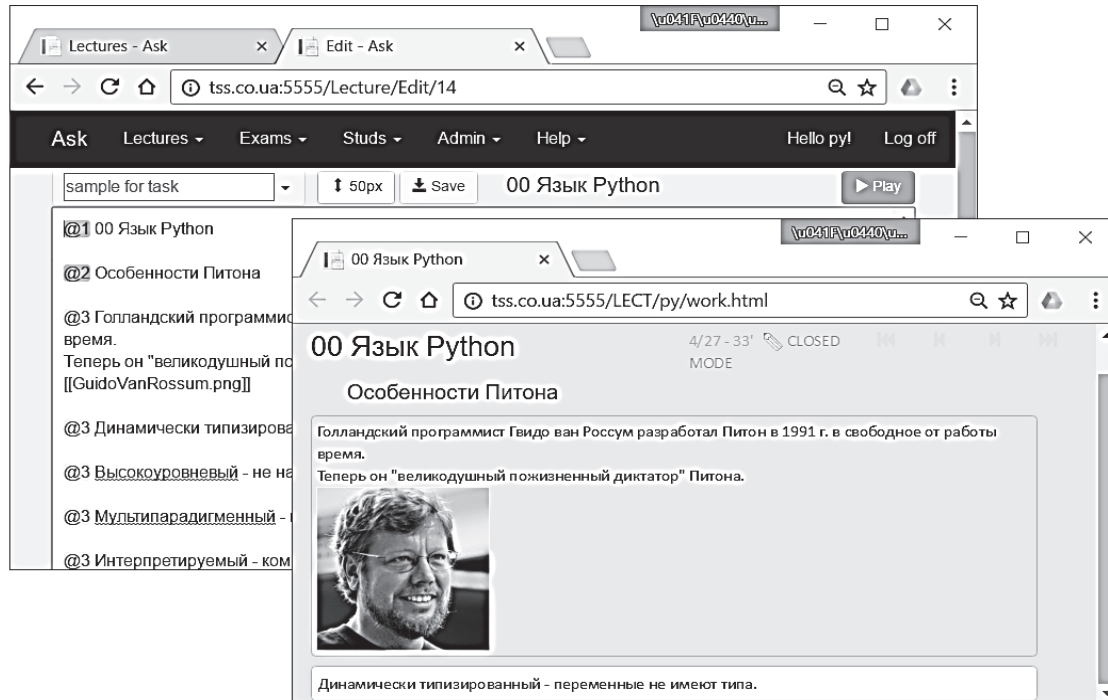


Рис. 2. Исходный текст и результат компиляции

Возможности окна редактирования мало отличаются от возможностей простого текстового редактора, но есть несколько полезных функций. Например, задачи с автоматической проверкой можно добавлять в презентацию «в один клик» при помощи выпадающего списка вверху слева. Так же легко при помощи всплывающего меню можно добавлять готовые рисунки с локального компьютера.

Состав презентации

Исходный текст презентации – это последовательность слайдов, никаких других структурных единиц в нем нет. Начало каждого слайда отмечается двухсимвольным маркером «@п», конец никак не отмечается – предыдущий слайд заканчивается там, где начинается следующий. Второй символ в составе маркера обозначает тип слайда. В настоящий момент допускаются слайды шести типов, соответственно маркерами служат комбинации символов @1, @2, @3, @4, @5, @6. Бегло представим эти типы.

Слайд типа @1 – это заголовок презентации или ее части. В исходном тексте заголовком является одна строка. На ленте – это та же строка, но на-

писанная подходящим для заголовка шрифтом. Первый из заголовков служит в то же время и названием презентации.

Слайд типа @3 – основной текст. В этом типе слайда возможны стили:

- блок программного кода (окружается двойными фигурными скобками);
- выделенный текст (окружается фигурными скобками);
- ссылка (окружается двойными квадратными скобками).

Содержание ссылки говорит само за себя, и в зависимости от него, в готовой презентации это будет рисунок, ссылка на ресурс в интернете или авторский вопрос (о них мы поговорим позднее). Кроме того, все слова, написанные латинским шрифтом, в готовой презентации автоматически будут выделены курсивом.

Безусловно, это самый нагруженный разметкой тип слайда, и разметка делается тремя типами скобок: фигурными, двойными фигурными и двойными квадратными.

Слайд типа @4 – определение. Это просто текст в рамке. Служит для показа того, что стоит обводить рамкой, например, для синтаксиса оператора цикла.

Слайд типа @5 – задача с поэтапным решением. Иногда хочется предложить слушателям написать несложную программку или хотя бы подумать о том, как это можно сделать.

Какой-то вариант решения можно подготовить заранее, чтобы потом показать его целиком или частями. Формат слайда таков:

@5 имя_задачи | условие_задачи
блок программного кода

...

блок программного кода

Блок кода – это одна или более строк, заключенных в двойные фигурные скобки, как в слайде @3.

В презентации студенты видят только название и условие задачи, а решение (блоки кода) скрыты от них. Дав слушателям какое-то время, чтобы самостоятельно обдумать возможные решения, преподаватель отрывает очередной блок, кликая мышью по названию задачи.

Слайд типа @6 – задача на программирование с автоматической проверкой решения. Эти задачи студенты должны решать после лекции, работая с

опубликованной презентацией. Публиковать или не публиковать презентацию, естественно, решает преподаватель, но возможность публикации «в один клик» в систему заложена.

В исходном тексте слайд @6 – это всего лишь номер и название задачи из автоматического задачника, о котором будет сказано немного позже. Формат слайда таков:

@6 номер_задачи | название_задачи

На рис. 1 изображена страница редактора исходного текста, в котором показано начало лекции о языке Питон. Презентация начинается с заголовков первого и второго уровня, дальше следует слайд основного текста со ссылкой на изображение.

На переднем плане рисунка виден результат компиляции этого исходника – динамический HTML-документ, который демонстрирует преподаватель во время чтения лекции.

Чтение лекции

Как уже говорилось, готовая презентация – это HTML-документ, который преподаватель демонстрирует в веб-браузере во время чтения лекции. Для перемещения по слайдам служат клавиши со стрелками Up, Down, PageUp и PageDown. Пример страницы с презентацией показан на рис. 3.

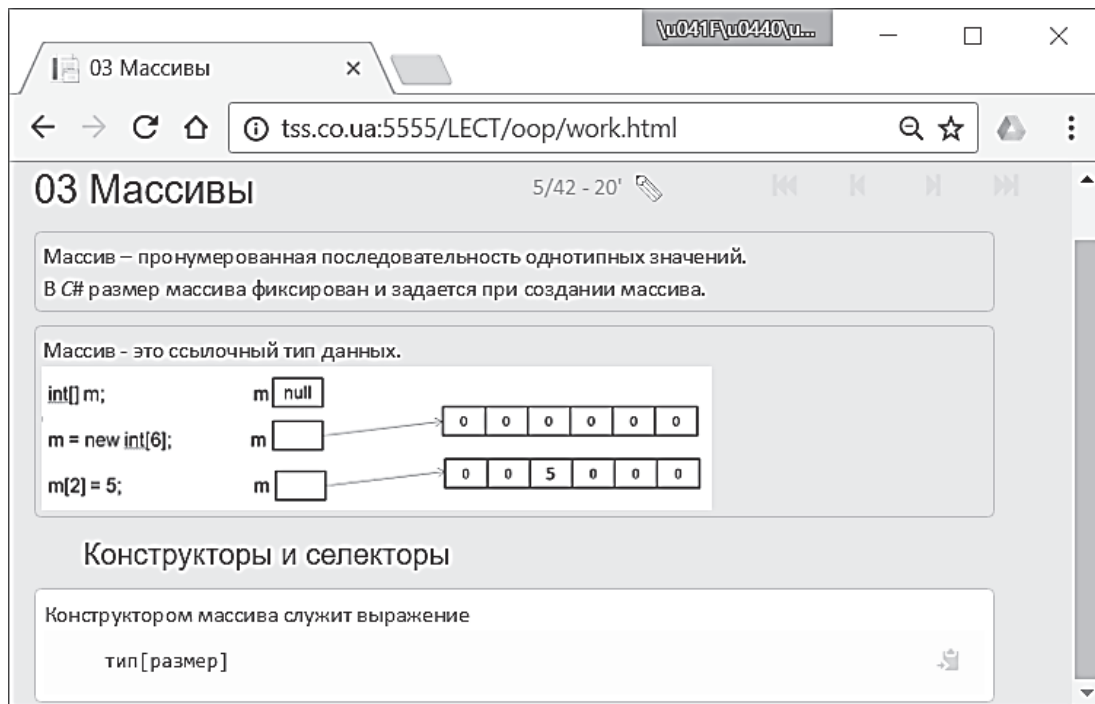


Рис. 3. Страница презентации

В верхней части документа находится полезная для лектора информация: номер очередного слайда, общее количество слайдов, число минут, оставшихся до звонка (расписание звонков задается в настройках). Там же находится иконка маркера для проведения линий поверх слайда и кнопки, дубли-

рующие клавиши навигации, не случай, если демонстрация ведется с планшета. Всё это почти невидимо, чтобы не отвлекать внимание слушателей, и обретает полную видимость только при наведении курсора мыши.

Примеры программного кода в презентации даются с подсветкой, присущей выбранному языку программирования (тоже – в настройках). На каждом блоке кода имеется кнопка, переносящая код в буфер обмена для последующей работы с ним.

На рис. 3 видно начало презентации по теме «Массивы» из курса лекций по объектно-ориентированному программированию. Преподаватель уже открыл несколько слайдов, текущий слайд выделен яркостью, чтобы фокусировать на нем внимание слушателей.

В верхней части страницы видна вспомогательная информация: текущий слайд – пятый, всего слайдов 42, до звонка 20 минут, иконка маркера. Еще правее кнопки для управления на тачскрине. Вспомогательная информация полупрозрачна, но при наведении на нее курсора становится отчетливой.

Страница студента

Настал момент посмотреть на все происходящее глазами студента. Разумеется, во время лекции его глаза устремлены (или должны быть устремлены) на экран. Но у него есть своя страница в этом приложении (рис. 4), с которой он, в основном, работает после лекции, но может и во время оной. На странице показан список курсов, по которым имеются опубликованные преподавателями материалы. В данном случае их два: asp (курс по ASP.NET Core) и opr (Основы программирования). Ссылки в первом столбце ведут к оглавлениям и оттуда – к конкретным лекциям.

Ссылки в третьем столбце ведут на страницу, с которой можно задавать вопросы, как во время, так и после лекции (рис. 5).

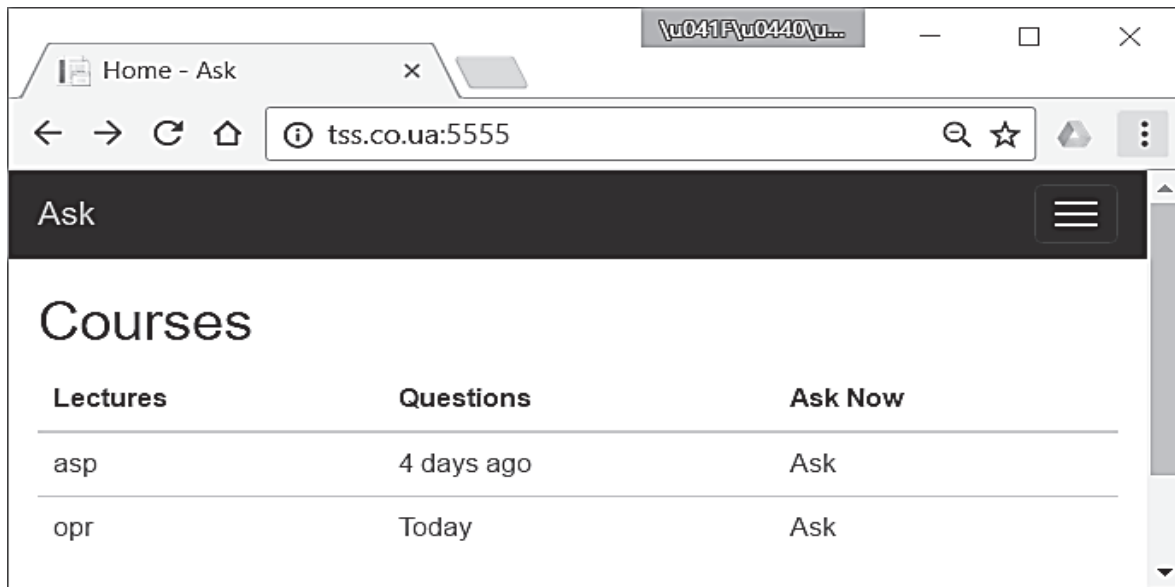


Рис. 4. Домашняя страница студента

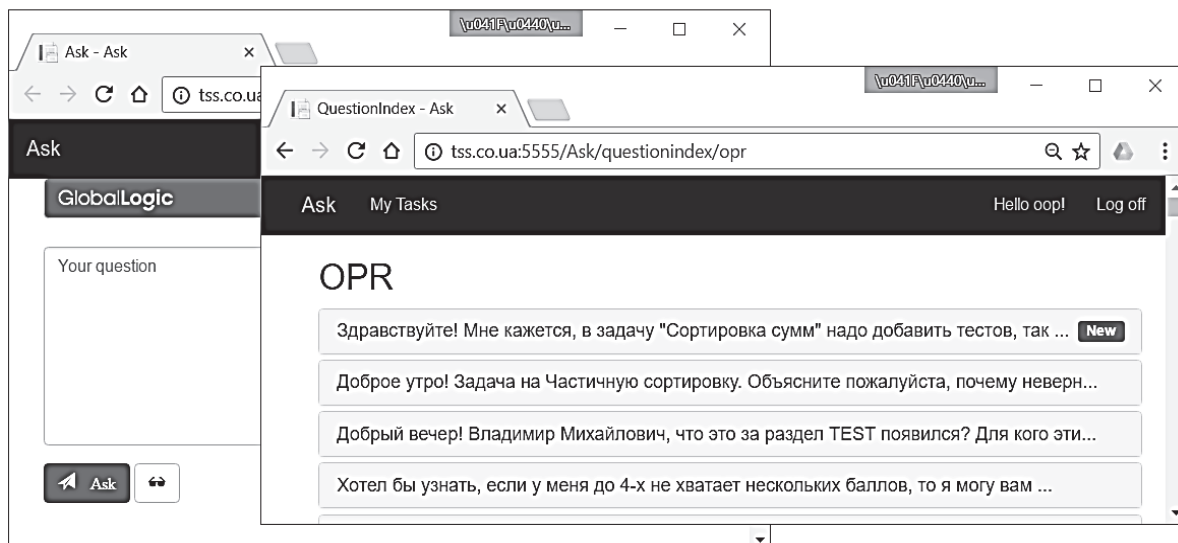


Рис. 5. Страницы вопросов и ответов

Независимо от того, когда был задан вопрос, он попадает в общий список вопросов и ждет там внимания преподавателя, который со своей стороны имеет доступ к этому списку. Если вопрос задан во время лекции, сигнал об этом в виде вопросительного знака появится на панели вверху презентации, и преподаватель сможет ответить на него в ходе лекции. На заочные вопросы преподаватель отвечает письменно по мере возможности, обычно вечером.

Ссылки во втором столбце ведут к списку вопросов с ответами преподавателя. Он нужен тем, кто задавал вопросы заочно.

Как можно понять, вопросы задаются не анонимно, и если преподаватель захочет, он сможет увидеть имя спросившего. Конечно, это лишает диалог интереса для некоторых студентов, но вводит его в профессиональное русло.

Очевидно, что возможность почти анонимно задавать вопросы (имя спросившего не видно слушателям) должно облегчить эту процедуру для тех, кто не любит попадать в центр внимания большой аудитории. Здесь же кроется еще одно средство сделать лекцию более динамичной. Лектор может не ждать милости от студентов и сам заготовить себе пару-тройку вопросов, на которые даст остроумный и даже блестящий ответ во время лекции. Один из авторов сам так поступал, правда, в конце лекции сознавался и предлагал угадать слушателям, какие

из заданных вопросов «авторские». Одним словом, механизм авторских вопросов имеется в системе и при желании его можно использовать.

На рис. 5 видно, что правее кнопки Ask находится еще одна кнопка – Unclear. Ее студент может нажать, когда материал лекции ему непонятен в целом, и он не в состоянии сформулировать конкретный вопрос. Эти сигналы также видны преподавателю и он сможет принять их во внимание.

Ответы на свои и чужие вопросы студент может увидеть на странице ответов (на рис. 5 справа). Пункты списка разворачиваются при выборе их мышью или касанием на тачскрине.

Задачи в презентации

Поддержка задач с автоматической проверкой решения осуществляется отдельной сетевой службой, детальное описание которой выходит за рамки данной статьи. Готовая презентация является клиентом этой службы и использует предоставляемый службой программный интерфейс. Именно это позволяет в исходном тексте презентации лишь сослаться на номер задачи (слайд типа @6), чтобы при просмотре презентации получить условие задачи, текстовое поле для ввода решения и возможность проверки решения на стороне упомянутой службы. Результат проверки возвращается в презентацию в виде строки текста и появляется под решением (рис. 6).

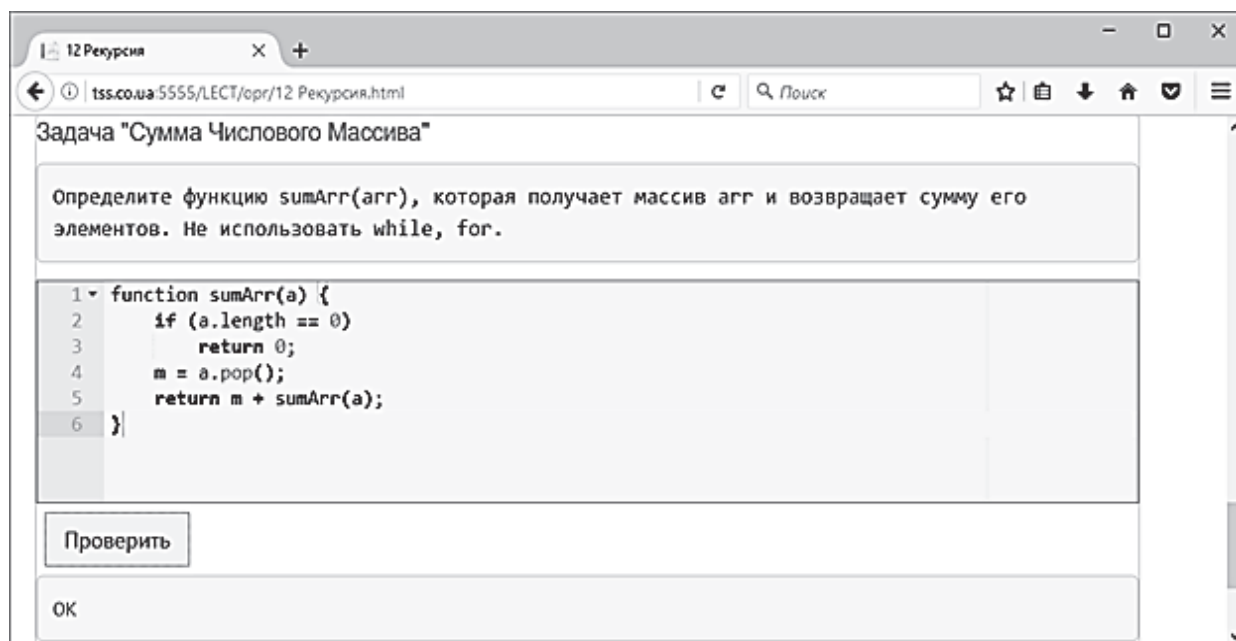


Рис. 6. Задача с автоматической проверкой

В распоряжении службы имеется задачник с задачами для нескольких языков программирования (Python, Javascript, C#, C++, Haskell). Задачи распределены по тематическим разделам и готовы к использованию любым из зарегистрированных в системе преподавателей.

Заметим, что наличие задач с автоматической проверкой решения позволяет использовать данную сетевую среду для подготовки не только лекций, но и методических указаний к практическим и лабораторным работам [10].

Заключення

Предлагается сетевое приложение, позволяющее легко создавать презентации лекций, направленных на преподавание программирования. Презентация представляет собой динамический HTML-документ загружаемый в стандартный браузер.

Помимо создания презентаций, приложение предоставляет услуги по хранению исходных текстов презентаций, объединению их в курсы лекций, поддержку проведения лекций и даже контроля знаний.

Важным элементом преподавания программирования являются задачи для самостоятельного решения. Выдача условий и автоматическая проверка решения таких задач выполняется отдельной службой, к которой имеет доступ данное приложение.

Приложение используется на кафедре программной инженерии ХНУРЭ на протяжении трех лет и за это время сэкономило много сил и нервов тем, кто им пользуется.

Список литературы

1. *Stepik (Стэпик)* – образовательная платформа и конструктор онлайн-курсов. [Электронный ресурс]. Режим доступа: <https://stepik.org/>.
2. *PROMETEUS* – бесплатные онлайн-курсы от преподавателей КНУ, КПИ и Киево-Могилянської академії. [Электронный ресурс]. Режим доступа: <https://prometheus.org.ua/>.
3. *Online Judge* – архив задач Вальядолидского университета в Испании. [Электронный ресурс]. Режим доступа: <https://uva.onlinejudge.org/>.

4. *Timus Online Judge*, архив задач с проверяющей системой [Электронный ресурс]. Режим доступа: <http://acm.timus.ru/>.

5. *Ejudge* – система для проведения различных мероприятий, в которых необходима автоматическая проверка программ. [Электронный ресурс]. Режим доступа: <http://ejudge.ru/>.

6. *TopCoder* – корпорация, проводящая соревнования по спортивному программированию. [Электронный ресурс]. Режим доступа: <http://www.topcoder.com/>.

7. *CS Academy* – образовательный проект в сфере программирования. [Электронный ресурс]. Режим доступа: <https://csacademy.com/>.

8. Абрамян М.Э. Веб-среда разработки и обучения [Текст] / М.Э. Абрамян, С.С. Михалкович // Открытые системы. СУБД. – 2012, No. 10. – С. 56-59.

9. *Programming Taskbook*. Электронный задачник по программированию. [Электронный ресурс]. Режим доступа: <http://ptaskbook.com/ru/>.

10. Бондарев В.М. Мережева підтримка навчання програмуванню [Текст] / В.М. Бондарев, Ю.Ю. Черепанова // Матеріали статей П'ятої Міжнародної науково-практичної конференції "Інформаційні технології та комп'ютерна інженерія", Івано-Франківськ, 27–29 мая 2015 р. – С. 49-50.

Поступила в редколлегию 23.06.2017

Рецензент: д-р техн. наук проф. Г.Г. Четвериков, Харьковский национальный университет радиоэлектроники, Харьков.

МЕРЕЖЕВЕ СЕРЕДОВИЩЕ ДЛЯ ПІДГОТОВКИ ТА ПРОВЕДЕННЯ ЛЕКЦІЙ

В.М. Бондарев, Ю.Ю. Черепанова

Розглядаються питання проектування та розробки мережевого додатку, що дозволяє створювати презентації лекцій з програмування у вигляді динамічних HTML-документів, здійснювати проведення лекцій, контроль знань. Контроль знань з програмування пропонується проводити шляхом видачі умов завдань та автоматичної перевірки рішення за допомогою мережевої служби. Запропонована система знаходиться в експлуатації і добре себе зарекомендувала.

Ключові слова: підготовка лекцій, навчання програмуванню, презентація, автоматичний контроль знань.

NETWORK ENVIRONMENT FOR PREPARATION AND READING LECTURES

V. Bondariev, Y. Cherepanova

The problems of the design and development of network application that allows to create presentations on programming in the form of dynamic HTML documents, to carry out lectures, to provide knowledge control is discussed in the article.

It is proposed to carry out the control of knowledge on programming with the help of the web service that gives conditions of tasks and accepts and checks task solutions. The proposed system is in operation and works well.

Keywords: preparation of lectures, learning programming, presentation, automatic knowledge control.