

М. А. Казаков

РЕАЛИЗАЦИЯ КОНЦЕПЦИИ МНОГОУРОВНЕВОЙ СИСТЕМЫ ДИСТАНЦИОННОГО ОБУЧЕНИЯ НА БАЗЕ ИНТЕРНЕТ ШКОЛЫ ПРОГРАММИРОВАНИЯ

Рассмотрена концепция многоуровневой подготовки квалифицированных кадров в области информационных технологий на базе Интернет-школы программирования реализуемой автором на технической базе кафедры «Компьютерные технологии». Кратко рассмотрены концептуальные и технические аспекты реализации.

Введение

С развитием информационных технологий широкое распространение получили системы дистанционного обучения. В настоящее время наблюдается интенсивный рост заинтересованности в таких системах, как со стороны учебных заведений, так и со стороны учащихся. Причиной такого интереса является наличие автоматизации большей части учебного процесса и возможность обучаться на расстоянии, что снижает нагрузку на преподавателей и повышает доступность обучения [1, 2].

В настоящей статье рассматривается дистанционное обучение элементарной теории алгоритмов программирования [3, 4]. Основной проблемой при обучении программированию является необходимость тестирования практических навыков [5, 6]. Учебный материал должен содержать большое количество динамических иллюстраций «визуализаторов» [7].

В последнее десятилетие в нашей стране был разработан ряд дистанционных Интернет систем, обеспечивающих информационную, методическую и учебно-тренировочную поддержку творческим конкурсам по информатике и программированию. Исторически системы дистанционной интернет-поддержки одаренных и профильномотивированных учащихся

развивались по двум направлениям. Одно из них было связано с непосредственной поддержкой олимпиад по информатике, программированию и компьютерным технологиям. Эти системы предназначены для уже имеющих достаточно обширную базовую подготовку студентов и школьников. На соответствующих олимпиадных сайтах, как правило, представлены достаточно сложные олимпиадные задачи. Второе направление ориентировалось на учащихся, которые хотели получить начальную подготовку по базовым фундаментальным курсам компьютерных информационных технологий и в случае успешного освоения материала перейти на следующую образовательную ступень и стать пользователями олимпиадных сайтов. Сайты второго направления обычно назывались интернет-школами или интернет-курсами.

В настоящее время на технической базе кафедры «Компьютерные технологии» активно разрабатывается, апробируется и внедряется система дистанционного обучения под названием «Интернет-школа программирования» [8]. В первой части статьи рассматривается концепция многоуровневого обучения специалистов в области информационных технологий. Интернет-школа программирования объединяет в себе особенности обоих указанных направлений, поскольку в рамках одной системы объединяет как Интернет-школу, так и систему поддержки Интернет-соревнований. Во второй части дано краткое техническое описание реализации Интернет-школы, особенности архитектуры и используемых технологий.

Описание концепции

Подробное описание терминологии предметной области можно найти в статье [6]. Поэтому здесь мы не будем останавливаться на этом, а перейдем непосредственно к концепции и ее реализации в рамках Интернет школы программирования.

В основе многоуровневой системы подготовки специалистов лежит разделение процесса обучения на некоторое количество этапов. По окончании

каждого этапа ученик выходит на качественно новый уровень подготовки.

Условно каждый этап можно разделить на три стадии:

- Получение новых знаний
- Освоение полученных знаний
- Контроль уровня освоенных знаний.

В рамках Интернет-школы предлагается схема представленная на рис. 1.

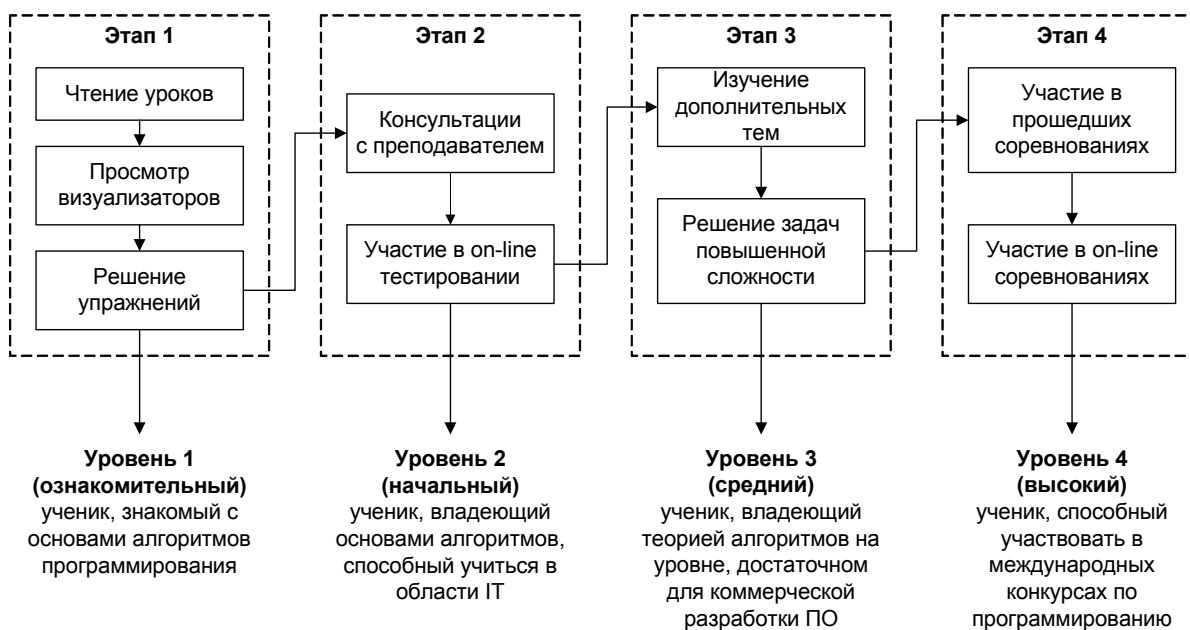


Рис. 1. Концепция 4-х уровневой подготовки теории алгоритмов

Рассмотрим отдельно каждый этап подготовки.

Первый этап соответствует ознакомительному этапу подготовки. На этом этапе ученику доступны следующие возможности:

- Знакомство с учебным материалом. Материал представлен на сайте Интернет-школы программирования в форме HTML уроков, снабженных иллюстрациями и визуализаторами.
- Решение упражнений. В текстах уроков встречаются упражнения, предлагаемые для самостоятельной проработки.

Целью данного этапа является *профильная ориентация*. Среди артефактов данного этапа можно выделить:

- Текст урока – словесное описание теоретических основ с примерами исходного кода, разборы упражнений, приведенных в предыдущих уроках.
- Визуализаторы – динамические иллюстрации, представляющие выполнение алгоритма при определенных входных данных.
- Упражнения – относительно небольшие задания для самостоятельной проработки учебного материала.

В процессе прохождения первого этапа ученик получает начальное представление об основах алгоритмов. При этом посредством самоконтроля при помощи системы автоматической проверки решений задач ученик способен оценить свои способности в области программирования.

Второй этап характеризуется более детальной отработкой практических навыков ученика. Особенностью данного этапа является наличие более жесткого расписания обучения и взаимодействие с преподавателем. На этом уровне ученик детально изучает учебные материалы с целью *научиться решать задачи*. Консультации преподавателя (тьютора) помогают ученику разобраться с учебным материалом. По окончании изучения разделов учебного курса ученик проходит on-line тестирования с целью определить уровень освоения материала. На данном этапе появляются следующие артефакты:

- Обратная связь с преподавателем – реализуется в виде форума, переписки по e-mail.
- Автоматическая проверка решений – технология, позволяющая проводить дистанционную проверку практических навыков посредством автоматической проверки решений задач. В данном случае Интернет-школа программирования обеспечивает техническую базу для проведения таких тестирований.
- Групповые тестирования – специальные тестирования, организуемые в Интернет-школе программирования с ограниченными временными рамками. Результаты этих тестирований рассматриваются на

предмет выявления абитуриентов способных к обучению в области ИТ.

На рассматриваемом этапе начинает играть роль концепция *персонализации* [9] ученика в системе. В результате успешного прохождения этого этапа ученик владеет основами алгоритмов и готов к поступлению в вуз по специальности в области ИТ.

Третий этап подразумевает знание расширенного набора алгоритмов и умение их применять. На этом этапе ученик переходит в другой раздел единой Интернет системы – архив задач. В этом разделе представлено множество задач различной степени сложности, автоматическая система проверки решений проверяет правильность решения задач. Для решения многих задач требуется воспользоваться справочной системой, включающей в себя разборы задач с подробными описаниями стандартных алгоритмов и способами их применения. Целью данного этапа является *выработка практических навыков* использования элементарной теории алгоритмов. На этом этапе появляются новые артефакты:

- Справочные материалы – набор решений и комментариев к различным типовым задачам.
- Архив задач – структурированная система задач с различных олимпиад, снабженная набором тестов.

Следует отметить, что основой обучения на этапах 3 и 4 является обучение через решение задач.

Четвертый этап (не обязательный) подразумевает *расширенные знания алгоритмов*, умение быстро и эффективно применять их. На этом этапе формируются такие навыки как:

- Быстрота реализации алгоритмов.
- Умение реализовывать алгоритмы с наименьшими затратами на тестирование.
- Умение быстро локализовать и устранить ошибку в готовом тексте программы.

С одной стороны этот этап можно рассматривать как подготовку к участию в олимпиадах по программированию, но с другой стороны можно рассматривать этот этап как высший уровень подготовки специалиста IT в области теории алгоритмов. На этом этапе появляются следующие артефакты:

- Исторические соревнования – соревнования, результаты которых хранятся на сайте Интернет школы программирования в разделе олимпиадного on-line сервера. Любой ученик может принять участие в таких виртуальных соревнованиях на правах участника, тем самым попробовать свои силы не только в задачах повышенной сложности, но и в условиях ограниченного времени.
- On-line соревнования – эти соревнования проводятся на основе автоматической проверочной системы, которая в этом случае не только проверяет решения, но и подводит итоги соревнований, так же как это происходит в случае on-line тестирований. Следует отметить, что в последние годы система проведения on-line соревнований используется для проведения официальных четвертьфинальных соревнований чемпионата мира по программированию.

Контрольными испытаниями на этом этапе является участие в соревнованиях наряду с другими участниками (бывшими учениками).

Таким образом, четырехуровневая система подготовки квалифицированных кадров в области информационных технологий позволяет поэтапно, шаг за шагом, пройти все этапы, от ознакомления с предметом элементарной теории алгоритмов до высококвалифицированного специалиста в области IT. Следует отметить, что подготовка в области технологий и других областей IT выходит за рамки данной концепции.

Технические аспекты реализации

Для описания технических подробностей реализации проведем классификацию и группировку артефактов Интернет школы программирования с целью описания реализации каждой группы артефактов в отдельности.

1. Статические материалы: уроки, упражнения, задания, справочные материалы.
2. Визуализаторы алгоритмов
3. Задачи для проверки практических навыков, разработка тестов.
4. Архив задач.
5. Система автоматической проверки решений.
6. Система проведения соревнований и тестирований.

Поскольку все справочные материалы находятся в общем доступе на *http*-сервере, то они оформлены в виде статических *HTML*-файлов.

Для формирования уроков используется паттерн *представление с преобразованием (Transform view)* [10]. При этом для увеличения скорости обработки запросов преобразование производится не динамически, а статически. Такой подход позволяет хранить исходные данные для уроков в *XML*, в то время как пользователю виден результат преобразований в виде *HTML*. Для формирования уроков используется специальный генератор уроков. Схема действия генератора представлена на рис. 2.

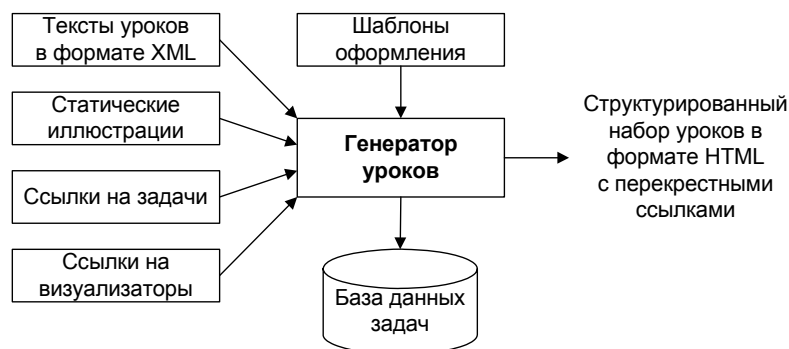


Рис. 2. Схема действия генератора уроков Интернет школы программирования

Необходимость в генераторе уроков возникла в результате опытных исследований. Основная проблема в ручном редактировании уроков была связана с тем, что авторы уроков часто вносят исправления, что приводит к огромному количеству ручной работы по поддержанию стиля форматирования. Поэтому был выбран формат *XML* как универсальный формат описания данных, в том числе и текстовых. В результате такой унификации, все правки вносятся в *XML*-документы, а далее все уроки, задействованные в правке, пересоздаются. Дополнительным удобством автоматической генерации является отсутствие в результирующих уроках лишних пробелов, поскольку генератор при создании *HTML*-документов делает их максимально компактными. При автоматической генерации уроки и задания автоматически нумеруются, а ссылки на них автоматически обновляются. Таким образом, автоматическая генерация существенно экономит время и уменьшает количество ошибок в тексте уроков.

Визуализаторы алгоритмов являются одной из особенностей Интернет школы программирования. История развития технологии создания визуализаторов, а также наиболее современная технология их создания подробно описана в работе [11]. Здесь же кратко отметим основные аспекты построения визуализатора.

Визуализатор – это программа, иллюстрирующая выполнение алгоритма при определенных входных данных. Применительно к дискретной математике и программированию, визуализаторы обычно моделируют некоторые алгоритмы, давая возможность обучающемуся при помощи интуитивно понятного интерфейса проходить алгоритм шаг за шагом от начала до конца, а при необходимости, и обратно.

Визуализатор «по-крупному» предлагается строить следующим образом:

- по программе, реализующей визуализируемый алгоритм, строится автомат логики визуализатора (контроллер – controller);
- выбираются визуализируемые переменные (модель – model);

- проектируется формирователь изображений и комментариев, который преобразует номер состояния и соответствующие значения визуализируемых переменных в «картинку» и поясняющий текст (представление – view).

Такая конструкция визуализатора соответствует одному из основных паттернов проектирования объектно-ориентированных программ, который обозначается аббревиатурой *MVC (Model-View-Controller)* [10]. Для реализации логики визуализатора используется формальное преобразование программы, реализующей визуализируемый алгоритм в конечный автомат.

Разработка учебных задач для Интернет-школы программирования состоит из двух этапов:

- Разработка условия задачи.
- Разработка тестов к задаче.
- Регистрация задачи в системе автоматической проверки.

Технология разработки тестов основывается на теории тестирования программного обеспечения. Специфика разработки тестов для автоматической проверки решений задач детально рассматривается в [5]. Здесь лишь отметим, что разработка тестов состоит из следующих этапов:

1. Написание верного решения задачи.
2. Написание условно верных решений задачи.
3. Рассмотрение возможных ошибок.
4. Написание системы тестов:
 - a. Тест из примера.
 - b. Набор тестов на проверку граничных условий.
 - c. Тесты на отсеечение ошибок.
 - d. Тесты на отсеечение условно верных решений.
5. Составление проверочной программы.

При регистрации задачи в автоматической проверочной системе, каждой задаче сопоставляется специальный уникальный идентификатор по структуре схожий со структурой *DNS* – стандарта соответствия между *IP*-адресами и

уникальными именами компьютеров в сети Интернет. Примером такого идентификатора может служить, например, следующая строка:

```
ru.ifmo.ips.Algorithms.lesson1.problem1
```

В данном случае это задача с сайта Интернет-школы программирования, глава «Алгоритмы», урок № 1, задача № 1.

Прежде чем рассматривать оставшиеся части программного комплекса Интернет-школы программирования, остановимся на общей схеме *PCMS2web*, изображенной на рис. 3 [6]. Система *PCMS2web* выполняет основную часть всех функций поддерживающих учебный процесс и проведение соревнований. Система *PCMS2web* реализована в виде *web*-приложения.

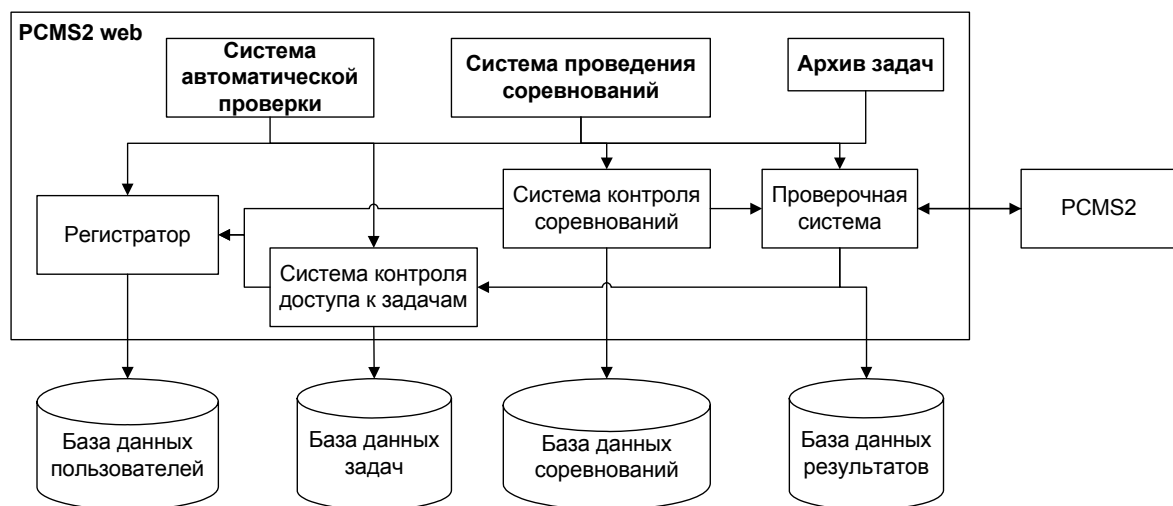


Рис. 3. Схема взаимодействия подсистем *PCMS2web*

Как видно из иллюстрации *PCMS2web* условно можно разделить на три уровня:

- **Уровень базы данных** (база данных пользователей, база данных задач, база данных соревнований).
- **Уровень бизнес логики** (регистратор, система доступа к задачам, система контроля соревнований, проверочная система, *PCMS2* – проверочное ядро).
- **Уровень представления** (система автоматической проверки, система проведения соревнований, архив задач).

Рассмотрим подробнее реализацию каждой из *компонент бизнес-логики*, которая реализована с использованием технологии сервлетов.

Регистратор реализует логику персонализации пользователя в рамках *PCMS2web*. К функциональности регистратора относится:

- Авторизация;
- Хранение в сессии *web*-сервера данных о пользователе;
- Проверка прав пользователя на доступ к отдельным страницам;
- Сохранение и отображение информации о пользователе.

Система контроля доступа к задачам отвечает за ограничение доступа к задачам и условиям задач. Условия задач находятся в структуре каталогов, не входящей в структуру *web*-сервера. Основная функция этой системы это ограничение доступа к условиям задач. Условия выдаются только в случае наличия разрешения у пользователя прав на просмотр задачи. Это позволяет во время тестирований и соревнований сохранять тайну условий до начала соревнований.

Система контроля соревнований отвечает за ведение процесса тестирований и соревнований. К функциям этой системы относятся:

- Хранение статистики подходов по конкретным задачам, а также статистики в рамках текущих для пользователя соревнований.
- Создание/удаление/переключение сессий (участия пользователя в соревновании), а также начало и окончание тестирования.
- Формирование таблицы результатов по сессии или по соревнованиям.

Проверочная система отвечает за проверку решений и запись результатов решения в базу данных. К функциям проверочной системы относятся:

- Помещение решение в очередь на проверку
- Выдача решений *PCMS2 kernel* на проверку и получение результатов

Проверка решения осуществляется на основе пакета, который содержит:

- Идентификатор задачи.
- Язык программирования.

- Текст исходного кода для одного или нескольких файлов.

Компоненты уровня представления выполняют функции взаимодействия с пользователем. Пользовательский интерфейс реализован с использованием технологии *JSP*, т.е. реализован паттерн представление по шаблону (*Template view*)[10].

Система автоматической проверки выполняет следующие функции:

- По результатам взаимодействия с пользователем формирует пакет с заданием для проверочной системы;
- Выдает пользователю статистику по проверке решений зада. Для каждого решения задачи выдается
 - Название задачи.
 - Время решения.
 - Результат.
 - Исходный код.

Архив задач выполняет функции поиска задач по соревнованиям, теме, названию.

Система проведения соревнований реализовывает пользовательский интерфейс к системе контроля соревнований и реализует все ее функции.

Система реализована, функционирует и продолжает совершенствоваться на технической базе кафедры «Компьютерные технологии». Следует отметить, что к настоящему моменту Интернет-школа программирования содержит три учебных курса, количество проверенных решений – около 30000, из них около 7000 верные решения, кроме того, система проведения соревнований использовалась для проведения 10 официальных соревнований – этапов чемпионата мира по программированию.

Заключение

В работе рассмотрена концепция многоуровневой подготовки квалифицированных кадров в области информационных технологий в применении к Интернет-школе программирования. Рассмотрены комплекс

программных и технологических решений позволяет утверждать, что в рамках Интернет школы программирования реализована четырехуровневая система подготовки в области элементарной теории алгоритмов.

Литература

1. Подготовка и проведение учебных курсов в заочно-дистанционной форме обучения. СПб.: СПбГТУ, 2000.
2. Агапонов С. В., Джалиашвили З. О., Кречман Д. Л., Никифоров И. С., Ченосова Е. С., Юрков А. В. Средства дистанционного обучения. Методика, технология, инструментарий. СПб.: БХВ-Петербург. 2003. 336с.
3. Васильев В.Н., Елизаров Р.А., Парфенов В.Г., Столяр С.Е. Организация дистанционного обучения программистов /Телематика'98. Всероссийская научно-методическая конференция. СПб.: СПбГУ ИТМО. 1998, с. 172–173.
4. Казаков М.А., Осипова Т.Г., Парфенов В.Г., Столяр С.Е. Интернет-школа программирования в СПбГИТМО /Телематика'99. Всероссийская научно-методическая конференция. СПб.: СПбГУ ИТМО. 1999, с. 165, 166.
5. Столяр С. Е., Казаков М. А. Подготовка тестов как часть технологии автоматизированного тестирования /XXX науч.-техн. конф. проф.-преподавательского состава. СПб.: СПбГИТМО(ТУ). 1999, с.105.
6. Казаков М. А. Система автоматического тестирования программных решений и проведения соревнований в режиме on-line // Вестник конференции молодых ученых СПбГУ ИТМО. Сборник научных трудов 2004. СПб.: СПбГУ ИТМО. 2004, с. 181–189.
7. Казаков М. А., Столяр С. Е. Визуализаторы алгоритмов как элемент технологии преподавания дискретной математики и программирования.

// Телематика'2000. Международная научно-методическая конференция.
СПб.: СПбГУ ИТМО. 2000, с.189–191.

8. Интернет-школа программирования <http://ips.ifmo.ru/>
9. Udi Manber, Ash Patel, and John Robison. Experience with Personalization on Yahoo! //Communications of the ACM. Vol.43. 2000. No. 8, p. 35 – 39.
10. Фаулер М. Архитектура корпоративных программных приложений. М.: Вильямс. 2004. 544 с.
11. Казаков М.А. Использование автоматного подхода для построения визуализаторов /Вестник конференции молодых ученых СПбГУИТМО. Сборник научных трудов. СПб: СПбГУИТМО, 2004, с. 166–180.