

# АВТОМАТИЗИРОВАННАЯ СИСТЕМА ДИСТАНЦИОННОГО ПРОВЕДЕНИЯ ОЛИМПИАД

УДК 004.42+ 004.75

---

## ПАРАМОНОВ Антон Иванович

доцент кафедры компьютерных технологий Донецкого национального университета.

**Научные интересы:** автоматизированное управление, информационные технологии.

**e-mail:** [paramonov\\_anton@mail.ru](mailto:paramonov_anton@mail.ru)

## СУРОВСКИЙ Алексей Эдуардович

магистр кафедры компьютерных технологий Донецкого национального университета.

**Научные интересы:** автоматизированное управление, информационные технологии.

### ВВЕДЕНИЕ

Тенденции развития образовательных процессов требуют постоянного совершенствования и внедрения современных технологических решений. Процесс вхождения высшей школы в мировое образовательное пространство требует модернизации, а также серьёзной переориентации компьютерно-информационной составляющей. Информационный взрыв породил множество задач и вопросов, важнейшие из которых связаны с обучением. Особый интерес представляют вопросы, связанные с автоматизацией обучения, а именно использование вычислительных ресурсов для подачи материала и обработки результатов контрольного опроса знаний учащихся. Внедрение инноваций в виде дистанционного обучения позволяет расширить возможности подготовки учебно-методического материала и организовать занятия в удаленном режиме, в удобное для учащегося время и в удобном месте. Дистанционное образование и технологическое развитие – неразделимые понятия, одно способствует развитию другого. Дистанционное образование [1] – это вид образования, при котором преподаватель и обучаемые физически находятся в различных местах и их взаимодействие полностью или частично осуществляется с помощью компьютеров и телекоммуникационных технологий и средств. Среди основных преимуществ

дистанционного образования можно выделить: доступность, гибкость, мобильность и масштабируемость. Однако, дистанционную форму образования сопровождают и сложности, которые можно разделить на такие классы, как технические (потребность в наличии аппаратно-технических средств), психологические (самодисциплина, мотивация, коммуникации) и учебные (контроль процесса обучения).

Общепризнанной формой обучающего процесса, в том числе дистанционного, являются олимпиады – мероприятия соревновательного характера. Олимпиады позволяют определить общий уровень подготовки участников, проверить их знания и умения, а также являются хорошим стимулом для развития и самоподготовки учащихся. Различные предметные олимпиады обладают своей спецификой, как заданий, так и формой проведения.

Олимпиада по программированию[2] – интеллектуальное соревнование по решению различных задач на ЭВМ, для решения которых необходимо придумать и применить какой-либо алгоритм и/или написать программу на одном из языков программирования. Олимпиады бывают личные или командные, очные и заочные. И все они могут быть реализованы в дистанционной форме.

Также следует отметить, что в последнее время соревнования по программированию не ограничивают-

ся олимпиадами. Оказалось, что соревнования по программированию можно также применять и в регулярном учебном процессе. Все более широкое применение находят тестирования – контрольные проверки знаний, умений и навыков учащихся, проводимые в форме соревнований в образовательных целях. В таких соревнованиях учащиеся решают задачи, отражающие прочитанный на уроках или лекциях материал. В такой же форме могут проводиться и тренинги учащихся. При этом учащийся, решивший задачу на практике, лучше усваивает материал и в дальнейшем, при необходимости решить подобную задачу как часть некоторой более глобальной проблемы, сделает это гораздо быстрее, чем в случае, если бы ему приходилось решать ее впервые.

Авторами был проведен подробный обзор существующих подходов к организации олимпиад и конкурсов по программированию в дистанционном режиме [3-5]. Детальное изучение правил проведения турниров позволило выделить набор характеристик, присущих различным видам олимпиад по программированию, из которых можно выделить следующие:

- временные ограничения для участников (длительность соревнования);
- взаимодействие с работами других участников (перекрестное оценивание и др.);
- порядок отправки решений (однократная/многократная проверка решений и др.);
- порядок оценивания решений;
- оценка полноты решения задачи;
- система штрафов за неверные попытки;
- временные и технические ограничения для решений.

На основе анализа полученных характеристик и особенностей рассмотренных подходов делается предположение, что для создания современной многофункциональной системы проведения олимпиад необходимо реализовать возможность учета комбинации и вариации различных параметров.

Именно учет различных подходов позволит создать систему дистанционного проведения олимпиад, на базе которой можно проводить мероприятия по различным правилам, в том числе динамически создавая новые виды соревнований. Так же это позволит создать реально расширяемую и масштабируемую систему, на базе которой можно централизованно проводить соревнования по программированию, вести рейтинг участников и формировать обширную базу задач, адаптируемую под задачи образовательного процесса.

В работе предлагается решение означенной задачи в виде автоматизированной системы дистанционного проведения олимпиад «uniJudge».

### АРХИТЕКТУРА СИСТЕМЫ

В основе решения положен компонентный подход с использованием клиент-серверной архитектуры. Разработанный программный комплекс «uniJudge» состоит из нескольких компонент (см. рис. 1): СУБД, координационный веб-сервер, клиентское веб-приложение, сервер проверки решений, АРМ «участник», АРМ «администратор/тренер».

СУБД используется для хранения данных и поддержания целостности структуры сущностей представляющих эти данные. Применяется реляционная структура. Координационный веб-сервер – это центральная часть системы, которая координирует взаимодействие остальных компонент системы. Архитектура комплекса предполагает наличие только одного координационного веб-сервера. Клиентское веб-приложение выполняет транзитную роль при взаимодействии АРМ «участник» и сервера проверки. Архитектура предполагает возможность его расположения как на одной узле с центральной частью, так и отдельно с установленным соединением по сетевому интерфейсу с основным проектом. На этот компонент полностью ложиться работа по обеспечению графического интерфейса для конечного пользователя.

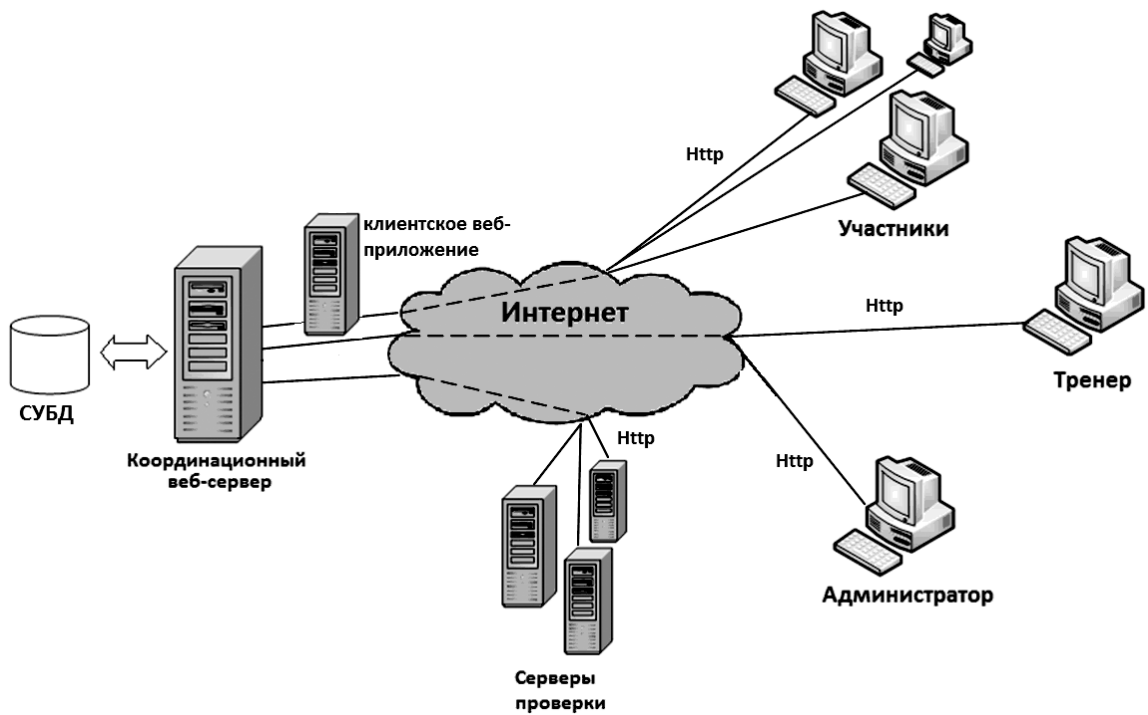


Рисунок 1 – Архитектура системы

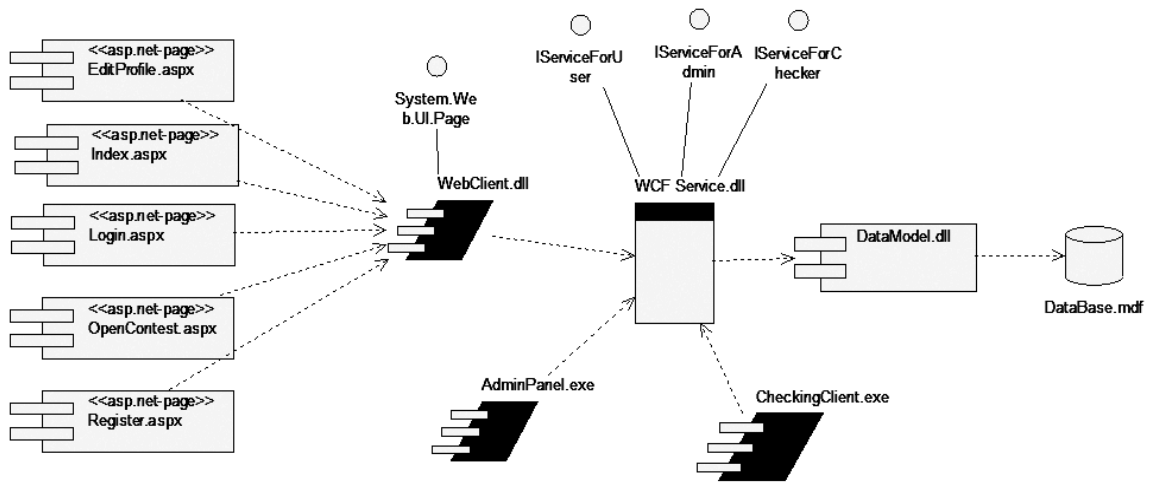


Рисунок 2 – Программная реализация компонент

В задачи компонента «Сервер проверки» входит компиляция, запуск и тестирование решений участников олимпиад. В качестве входных данных компонент принимает исходный код для проверки и наборы тестовых пар. Вердикты проверки по каждому из тестов компонент передает координационному веб-серверу для окончательного вердикта по решению участника. В качестве АРМ «Участник» выступает любой современ-

ный веб-браузер с поддержкой JavaScript. АРМ «администратор/тренер» реализован в виде приложения Windows Form, функциональность которого позволяет управлять сущностями системы: пользователями, задачами, соревнованиями и др. В АРМ администратора вводятся роли с разграничением по уровням доступа к функциональности.

Программная реализация компонент разработанной системы представлена на рисунке 2. Координационный веб-сервер, представленный на рисунке 2 библиотекой «WCF Service.dll», построен как набор веб-служб. Сервер имеет критическую зависимость только от СУБД, и может работать в режиме отсутствия любых других модулей, но этом случае в режиме ограниченной функциональности. Взаимодействие сервера с базой данных выполняется через прослойку ORM EntityFramework, изображенную на рис. 2 как модуль «DataModel.dll». Использование этого Framework позволяют легко манипулировать сущностями БД в системе.

Функциональность координационного веб-сервера реализована в виде трех 3 интерфейсов:

1. Интерфейс взаимодействия с участником соревнования (на рис. 2 – «IserviceForUser»). Через этот интерфейс взаимодействует компонент клиентской части со всей системой, реализуя возможность принимать участия в соревнованиях, возможность чтения задач, возможность предоставлять свои решения для проверки.

2. Интерфейс взаимодействия администратора с системой (на рис. 2 – «IserviceForAdmin»). Предоставляет возможность вносить изменения в систему, останавливать проверку, добавлять, изменять, редактировать всех необходимых сущностей. Эти возможности предоставляются администратору либо тренеру, но в ограниченном режиме.

3. Интерфейс взаимодействия модулей проверки (на рис. 2 – «IserviceForChecker»). Через этот интерфейс проверяющие модули берут задачи на проверку и выполняют непосредственно проверку, возвращая результат на сервер.

Клиентское веб-приложение представлено на рис. 2 модулем «WebClient.dll» и набором aspx-страниц. Данный компонент реализует клиент-серверное взаимодействие: для проверяющей системы выполняет роль клиента, а для конечного пользователя – роль веб-приложения (сервера). Взаимодействие веб-приложения и проверяющей системы выполняется по протоколу SOAP. SOAP-клиент может быть реализован на разных языках под разные платформы, в частности этот протокол поддерживают Java, .NET, PHP. Взаимодействие конечного клиента и веб-приложения проис-

ходит по протоколу HTTP (HTTPS), который выбран в силу следующих преимуществ:

- 1) Клиенты не зависят от операционной системы сервера.
- 2) Пользователь на свое усмотрение может выбрать тонкий клиент.
- 3) HTTP является самым распространенным для передачи трафика [6].
- 4) Возможность пропускать через HTTP прокси-сервера, кэшировать.

Данная архитектура позволяет полностью отделить интерфейс пользователя от проверяющей системы. Также данная архитектура позволяет одновременно использовать несколько проектов веб-приложения, что позволяет создавать множество интерфейсов для каждого из соревнований в рамках одной проверяющей системы.

Компонент сервера проверки (на рисунке 2 «CheckingClient.exe») представляет собой Windows-приложение, которое занимается проверкой присланных решений участников олимпиад. По отношению ко всей системе этот компонент работает в активном режиме, то есть он сам является инициатором проверки и сам “выбирает” с сервера решения для проверки. Проверяющий компонент работает в режиме долгого опроса, то есть если он подключается к серверу и на сервере нет задач для проверки, то сервер не спешит возвращать ответ, и оставляет клиента в “подвешенном состоянии” до тех пор, пока не появятся новые задачи, либо пока не истечет предельно допустимое время, через которое клиент должен подключиться вновь. Данное решение позволяет системе почти мгновенно передать решение на проверку, чего невозможно достичь при опросе по таймеру.

Компонент администратора/тренера (на рис. 2 «AdminPanel.exe») представляет собой Windows Form приложение, взаимодействующее с сервером по HTTP протоколу.

### ОСОБЕННОСТИ СИСТЕМЫ UNIJUDGE

Эффект, достигаемый за счет применения вычислительной техники, возрастает при увеличении масштабов обработки данных, то есть при концентрации больших объемов данных и процессов их обработки в рамках одной технической системы. Крупномасштабные системы обра-

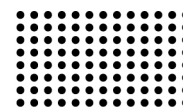
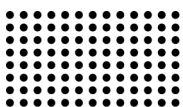
ботки данных можно создавать двумя путями: повышая мощность компьютеров или объединяя многие компьютеры в вычислительные комплексы и сети. Создание распределенных сетей из высокопроизводительных отказоустойчивых систем обработки данных сегодня рассматривается как одно из перспективных направлений развития вычислительной техники.

Основной особенностью системы «uniJudge» является ее возможность поддержки распределенной многоагентной сети проверяющих компонент. Эта возможность позволяет проводить соревнования с большими нагрузками (большим количеством участников, в условиях интенсивного использования). Каждый проверяющий сервер является клиентом для координационного сервера и выступает агентом для проверяющей распределенной сети. Все агенты ограничены (не знают о работе всей системы в целом) и децентрализованы (не имеют функций управления системой). При включении в сеть очередного проверяющего сервера он проходит авторизацию и выполняет процесс самотестирования, который заключается в расчете балансирующих коэффициентов путем выполнения ряда программ и сбора информации об использованном процессорном времени. Указанные настройки выполняются с целью организовать во время работы системы аппаратно и технически независимую оценку решений участников. Пожалуй, важнейшим аспектом распределенных систем является взаимодействие между частями системы, так как именно взаимодействие порождает основные трудности, как при разработке, так и при верификации распределенных систем. При использовании распределенной проверки в системе «uniJudge» определяется один главный проверяющий сервер, который будет считаться эталонным, и использоваться для проверки на граничных решениях. Граничными считаются те решения, для выполнения которых необходимых ресурсов требуется менее, чем на заданный порог (5%) от максимально допустимых. Чтобы обеспечить одинаковые условия проверки был реализован функционал, который граничные решения участников проверяет только на главном проверяющем сервере. Для обеспечения равномерной загрузки проверяющих серверов, в том числе главного, используется специальная система контроля, распределяющая поступающие решения между свободными серверами.

Следует отметить такую особенность системы как работа в режиме «виртуальные соревнования», что позволяет проводить одно соревнование для разных участников с раздельным стартом и общим итогом. Система позволяет применять индивидуальное время начала соревнования для каждого участника, при этом вся хронология отображения турнирной таблицы сохраняется единой общей, как если бы участники стартовали одновременно.

Задача процедуры тестирования программного решения состоит в определении, решает ли тестируемая программа требуемую задачу. В последнее время широкое применение находят методы, основанные на анализе исходного кода программ. Но для любого достаточно сложного языка программирования (эквивалентного машине Тьюринга), существуют программы, для которых эффективно невозможно определить, не только решают ли они заданную задачу, но даже, закончит ли программа выполнение за конечное время. Так как, точное определение корректности программы невозможно, то широкое применение находят различные эмпирические методы тестирования. При этом в зависимости от структуры задачи используются различные методы тестирования и оценки результатов. Универсальность разработанной системы подчеркивается возможностью проводить соревнования по программированию по различным правилам, среди которых можно выделить наиболее популярные [3-5]: ACM ICPC (International Collegiate Programming Contest), ACM+, IOI (International Olympiad in Informatics) и другие. Однако система не ограничивается приведенным выше списком и позволяет создавать новые виды соревнований на основе различных алгоритмов проверки.

Для реализации системы были использованы современные платформы и оболочки, в числе которых .Net Framework [7], Entity Framework [8], Windows Communication Foundation (WCF) [9]. Что, несомненно, также является особенностью разработанной системы. Применение возможностей WCF Framework позволяет сделать разрабатываемую систему более гибкой. Это достигается благодаря широким возможностям связки модулей системы, при этом сами модули абстрагированы от транспортного канала передачи данных, эту роль на себя берет WCF Framework. Сообщения могут отправляться по любому из нескольких встроенных транспортных протоколов в различных кодировках. Кроме того, WCF позволяет



отправлять сообщения по протоколу TCP, через именованные каналы. Сообщения можно кодировать в виде текста или использовать оптимизированный двоичный формат. Двоичные данные можно эффективно отправлять с использованием стандарта MTOM. Если ни один из предоставляемых транспортов и кодировок не подходит к текущим требованиям, вы можете создать собственный пользовательский транспорт или кодировку. WCF-сервер автоматически публикует свой интерфейс взаимодействия с помощью стандарта wsdl, что позволяет клиенту получить всю необходимую информацию о сервисе (такую как методы сервиса и передаваемые типы данных, в том числе сложные). Интерфейс взаимодействия является самодокументированным.

Взаимодействие всех модулей в системе происходит по HTTP протоколу [6]. Это позволяет пропускать трафик через HTTP прокси-сервера, не обращая внимания на ограничения брандмауэра. По такому же принципу построены и серверы проверки.

## ВЫВОДЫ

Предложенная в работе система позволяет автоматизировать процесс проведения соревнований и олимпиад по программированию. Система дает возможность организовать соревнования по различным правилам и в различных форматах, настраивать для автоматической работы подсистему расчета рейтингов участников в нужном представлении. Всё это делают её универсальной и легко адаптируемой под нужды конкретной образовательной программы. Возможность

поддержки работы сразу нескольких проверяющих серверов, которые являются самыми ресурсоемкими компонентами комплекса, позволяет легко масштабировать и расширять систему. Асинхронная обработка данных и независимость компонент системы делают её полноценно многофункциональной, что позволяет проводить параллельно множество различных соревнований, в том числе и на разных веб-узлах. При этом со стороны участников все выглядит как несколько самостоятельных проверяющих систем. Поддержка «виртуальных соревнований» предоставляет участникам возможность отдельного старта в любое удобное время и в любом месте, что гармонично вписывается в идеологию дистанционного образования.

Система прошла опытную эксплуатацию при проведении региональных олимпиад по информатике и программированию в Донецком национальном университете. Даже при больших нагрузках (более сотни одновременных участников) система стабильно работала и оперативно обрабатывала запросы пользователей.

Разработанный комплекс «uniJudge» является автоматизированным многофункциональным инструментом для дистанционного проведения мероприятий соревновательного характера. Система может быть применена как преподавателями, так и непосредственно учащимися для самоподготовки и самоконтроля. Технологии разработки системы позволяют использовать её как автономный продукт, так и как часть глобальной образовательной программы.

## ЛИТЕРАТУРА:

1. Овсянников В.И. Введение в дистанционное образование /Овсянников В.И., Густырь А.В. – М.: РИЦ «Альфа» МГОПУ, 2001. – 286 с.
2. Андреева Е.В. Московские олимпиады по информатике /Е.В. Андреева, В.М. Гуровиц, В.А. Матюхин. – М.: МЦНМО, 2006. – 256 с.
3. Официальный сайт IOI. Правила проведения соревнований International Olympiad in Informatics (Документ WWW) , URL: <http://www.ioinformatics.org/rules/>
4. Официальный сайт Кубка И.Н. Векуа. Правила проведения соревнований (Документ WWW), URL: [http://acm.spbgu.ru/~snark/vekua/index.cgi?data=2010/vekua\\_ind&year=2010&class=vekua2010](http://acm.spbgu.ru/~snark/vekua/index.cgi?data=2010/vekua_ind&year=2010&class=vekua2010)
5. Официальный сайт ACM ICPC. Правила проведения соревнований (Документ WWW), URL: <http://icpc.baylor.edu/worldfinals/rules/>
6. Дембинская Н. Объем HTTP-трафика впервые превысил P2P //Компьюлента (Документ WWW), URL: <http://net.compuenta.ru/322974>
7. Тони Нортрап Основы разработки приложений на платформе Microsoft .NET Framework /Тони Нортрап, Шон Вилдермьюс, Билл Райан. – СПб.: Питер, 2007. – 864 с.
8. Daniele Bochicchio Entity Framework 4 in Action /Daniele Bochicchio, Stefano Mostarda, Marco de Sanctis. – Island: Manning Publications, 2011. – 578 p.
9. Пабло Сибраро Windows Communication Foundation и .NET 4 для профессионалов /Пабло Сибраро, Курт Клайс, Фабио Коссолино, Йохан Грабнер. – К.: Диалектика, 2011. – 464 с.