УДК 004.4, 37.018
DOI 10.25205/1818-7900-2019-17-2-61-73

Системы автоматизированной оценки заданий по программированию: разработка, использование и перспективы

Д. В. Иртегов [1], Т. В. Нестеренко [1, 2], Т. Г. Чурина [1, 2]

[1] *Новосибирский государственный университет*
Новосибирск, Россия
[2] *Институт систем информатики им. А.П. Ершова СО РАН*
Новосибирск, Россия

Аннотация

Приводится обзор существующих систем проверки заданий по программированию, кратко описано современное состояние архитектуры системы NSUts, созданной в Новосибирском государственном университете, и практика ее использования в учебном процессе и в олимпиадах по программированию различных уровней. Приведен анализ опыта использования таких систем и обозначены перспективы их развития.

Ключевые слова

тестирование знаний, обучение программированию, олимпиады по программированию, системы тестирования, NSUts

Для цитирования

Иртегов Д. В., Нестеренко Т. В., Чурина Т. Г. Системы автоматизированной оценки заданий по программированию: разработка, использование и перспективы // Вестник НГУ. Серия: Информационные технологии. 2019. Т. 17, № 2. С. 61–73. DOI 10.25205/1818-7900-2019-17-2-61-73

Systems for Automated Evaluation of Programming Tasks: Development, Use and Perspectives

D. V. Irtegov [1], T. V. Nesterenko [1, 2], T. G. Churina [1, 2]

[1] *Novosibirsk State University*
Novosibirsk, Russian Federation
[2] *A. P. Ershov Institute of Informatics Systems SB RAS*
Novosibirsk, Russian Federation

Abstract

The paper is devoted to the issues related to the development and use of automated assessment systems for programming tasks. The work on automated testing of programming tasks had been held in Novosibirsk State University since 1998. It was created several generations of systems of this type. The first program of about 1,000 lines was written in Perl. Currently, the automated system, named NSUts, is used in the programming contests and in the educational process. As a result of work in it, as well as acquaintance with other similar systems, both Russian and foreign, it was possible to formulate requirements for software development of this kind, to analyze the difference between the systems for testing competitions and training support systems.

The article also provides an overview of the existing systems for checking tasks on programming, an analysis of the advantages and disadvantages of each of these systems, and a detailed comparison of the NSUts system with the Yandex Contest system. The current state of the NSUts system architecture is described. Much attention is paid to the practice of using the system in the educational process, statistics and graphs of students' progress of students studying with the help of this system are given. The article describes the analysis of the experience of using the NSUts system and outlines the prospects for its development.

© Д. В. Иртегов, Т. В. Нестеренко, Т. Г. Чурина, 2019

Keywords
knowledge testing, programming training, programming contests, testing systems, NSUts
For citation
Irtegov D. V., Nesterenko T. V., Churina T. G. Systems for Automated Evaluation of Programming Tasks: Development, Use and Perspectives. *Vestnik NSU. Series: Information Technologies*, 2019, vol. 17, no. 2, p. 61–73. (in Russ.) DOI 10.25205/1818-7900-2019-17-2-61-73

Введение

Идея создания автоматизированной тестовой системы для проверки заданий по программированию не нова. Попытки создания таких систем предпринимались еще в 80-е гг. XX в. [1]. Можно выделить две основные проблемы, которые затруднили обретение самостоятельности этим направлениям исследований.

Во-первых, многие из разработчиков систем такого рода используют слова «автоматизированное тестирование» в описании задачи, поэтому соответствующие исследования относят либо к автоматизированному контролю качества программного обеспечения, либо к педагогическому тестированию [2–4]. Задача автоматизированного контроля навыков программирования хотя и находится в родстве с обоими названными направлениями, но требует иных подходов и иных инструментов.

Во-вторых, многие разработчики таких систем ставят задачу слишком узко. Так, разработчики системы ejudge [1] описывают свой программный продукт как «систему для проведения различных мероприятий, в которых необходима автоматическая проверка программ», а краткое описание системы на английском языке звучит как “contest management system”.

В англоязычной литературе также нет единого термина для обозначения программных продуктов такого типа. Так, организаторы международных командных соревнований по программированию ACM ICPC (ACM International Collegiate Programming Contest) [2] называют систему, используемую для проведения соревнований, “Programming Contest Control System” (система управления программными соревнованиями) [3]. В настоящее время в качестве такой системы используется система Kattis [4], авторы которой называют ее “Automated assessment system” (автоматизированная система оценки) [5]. Еще один термин, часто встречающийся в англоязычных названиях систем такого рода, – “Online judge”, вне контекста этот термин допускает множество неверных толкований.

С 1998 г. в НГУ ведутся работы по автоматизированной проверке заданий по программированию [6]. Было создано несколько поколений систем такого типа. Первая программа объемом около 1 000 строк была написана на языке Perl. В настоящее время в олимпиадном программировании и в учебном процессе используется система NSUts. В результате работы в ней, а также знакомства с другими подобными системами – как российскими, так и зарубежными, удалось сформулировать требования к программным разработкам такого рода [7], провести анализ отличия олимпиадных систем от систем поддержки обучения [8].

Обзор существующих систем проверки знаний

Одной из первых российских систем автоматизированной проверки была система T/Run для MS-DOS, которая состояла из двух программ: run и t. Первая из них осуществляла запуск решений с ограничением по времени работы, а вторая проверяла решения на заданных тестах. Это простое решение во многих случаях является и достаточным. Будучи дополнена

[1] Система ejudge. URL: https://ejudge.ru
[2] Сайт международной олимпиады по программированию ACM-ICPC. URL: http://icpc.baylor.edu
[3] Система CCS. URL: https://clics.ecs.baylor.edu/index.php/Contest_Control_System
[4] Система Kattis. URL: https://open.kattis.com/.

набором командных файлов MS-DOS, T/Run успешно обеспечивала тестирование решений даже на Всероссийских олимпиадах школьников по информатике. Сейчас T/Run представляет ограниченный интерес из-за малой применимости к решениям, написанным под операционную систему Windows.

После системы T/Run на Всероссийских олимпиадах использовалась система Cyber Judge, созданная М. А. Бабенко. Это сложное и расширяемое решение основано на платформе .NET. Дописывание к ней небольшого модуля позволило, например, демонстрировать фотографии участников, вызываемых на личное тестирование, во время проведения XIII Всероссийской олимпиады. Но использование этой системы затруднительно, поскольку ее можно применять только в присутствии автора. Это связано с тем, что, во-первых, настройка не ориентирована на пользователя и предполагает редактирование файлов конфигурации. Во-вторых, по системе нет документации. В-третьих, часто настройка системы предполагает дописывание к ней небольших модулей для конкретной олимпиады, что при отсутствии документации является большим препятствием даже для профессиональных программистов.

В следующей системе olympiads.ru [5] содержится, кроме программы запуска решений с контролем времени работы и использования памяти, еще база данных задач. Эта система уже имеет простой графический интерфейс пользователя, с помощью которого осуществляется редактирование базы данных и запуск решений на проверку. Появление в системе самотестирования базы данных и интерфейса отражает потребность участников в упрощении настройки систем тестирования. Между тем к появлению базы данных ведет и другое требование – автоматизация деятельности жюри по подсчету баллов и подведению итогов.

Примером системы самотестирования является tchoose, которая использовалась на Всероссийской олимпиаде по информатике для отображения результатов работы правильных и неправильных решений на подготовленных тестах. Отображение результатов тестирования в графическом виде кажется перспективной возможностью, удобной не только при подготовке олимпиад, но и во время тренировок.

Объединение двух систем Automated Programming Problem Evaluation System (APPES) на Java и Programming Contest Management System (PCMS) на Delphi, созданных в Санкт-Петербургском государственном институте точной механики и оптики, и их последующее развитие превратилось в проект PCMS2 [6]. Гибкость этой системы достигается низкой связностью модулей и качественным выделением высокоуровневых концепций. В плане возможностей PCMS2 значительно превосходила на тот момент все современные требования [9].

Однако такая гибкость имеет высокую цену. Даже использование настроенного сервера PCMS2 требует предварительного обучения и понимания синтезированных концепций, не относящихся к предметной области. Кроме того, «настроенным» PCMS2 никогда не бывает, поскольку большинство действий в нем осуществляется написанием файлов конфигурации, а пользовательский интерфейс имеет преимущественно информационные функции. Файлы конфигурации в формате XML составляют значительную часть системы как по важности, так и по объему, связывая ее модули вместе. Одни XML-файлы описывают формат других XML-файлов, внутри текстовых строк применяется особо разработанный синтаксис подстановок, некоторые XML-файлы представляют собой программы на специально созданном языке сценариев. Отдельные части системы еще не до конца разработаны, поэтому часть настроек выполняется путем написания и подключения специализированных модулей, что также могут выполнить только авторы системы Г. А. Корнеев и А. С. Станкевич. Отметим, что сервер PCMS2 поддерживает только ОС Windows. Связано это с различиями в способах запуска решений и замера потребления ресурсов под разными операционными системами.

В Московском государственном университете была создана система ejudge, которая поддерживает только операционную систему Linux. Она написана на языке Си и предоставляет

[5] Система olympiads.ru. URL: http://www.olympiads.ru/
[6] Система PCMS2. URL: http://neerc.ifmo.ru/trains/information/software.html

определенный набор функций, удовлетворяющих современным требованиям. В системе ejudge имеются всё те же конфигурационные файлы, описанные в документации. Многие действия требуется выполнять из командной строки запуском соответствующих утилит. Базовые возможности администрирования доступны через веб-интерфейс.

В системе ejudge можно отметить три недостатка, которые следуют из того, что тестирование проводится на той же машине, что и серверная часть системы. Первый из них – это возможная конкуренция за ресурсы между тестируемым решением и самой системой. В результате система может выдавать ложный вердикт «превышен лимит времени» для решений, которые на выделенной машине вполне уложились бы в этот лимит. Для обхода этой проблемы ejudge использует многократную прогонку решений с вердиктом «превышен лимит времени». Но очевидно, что это только снижает вероятность возникновения проблемы, а не решает ее. Ко второму недостатку можно отнести ограниченные возможности горизонтального масштабирования. Это означает, что подключить дополнительные машины, кроме той, на которой проходит тестирование, физически невозможно. Единственное, что можно сделать, это запускать несколько процессов проверки, по числу процессорных ядер на сервере. Но это может привести к конкуренции решений за оперативную память, т. е. к росту числа ложных вердиктов «превышен лимит времени». При проверке решений на олимпиадах могут случаться значительные задержки между отправкой решения и получением вердикта. Третий недостаток состоит в том, что тестирование возможно только под той же ОС, под которой работает сама система, т. е. под Linux. Это ограничивает применение системы, особенно для школьных соревнований, где правила требуют ОС Windows и компиляторы, доступные только для Windows или для учебных курсов, где ОС и язык программирования заданы программой курса.

Система автоматической проверки с архивом задач в испанском городе Вальядолиде [7] была создана гораздо раньше аналогичных российских разработок и получила большую популярность среди студентов. Для ее работы используется операционная система Linux, поддерживаются языки программирования: GNU C, GNU C++, Free Pascal. Архив постоянно пополняется новыми задачами и тестами. Для каждой задачи приводится статистика: отношение числа правильных решений задачи к общему числу попыток сдать данную задачу. Недостатком системы является тот факт, что вся коммуникация с участниками ведется через электронную почту. Также для многих неудобным является использование системы Linux. Система развернута на серверах университета Вальядолид. Доступ к этим серверам бесплатен, но скачать программное обеспечение и развернуть его на своем оборудовании невозможно. Также все изменения в системе: добавление языков и версий компиляторов, изменение правил подсчета рейтинга – возможны только с разрешения и при активном участии ее разработчиков.

Главным же недостатком данной системы и ряда рассматриваемых далее систем является то, что они используют модель Software as Service (SaS). По такой модели программное обеспечение не устанавливается на компьютеры пользователей, а развертывается на серверах поставщика услуг. Фактически, пользуясь услугой, клиент получает и программу, и вычислительные ресурсы для ее работы. Часто программное обеспечение вообще не доступно конечному пользователю. Данная модель используется как для коммерческих, так и для бесплатных или условно-бесплатных услуг. Например, такие популярные сервисы, как Gmail или Yandex mail, работают именно по этой модели. Для многих сценариев использования эта модель привлекательна, поэтому она и получает широкое распространение.

Но эта модель имеет и ряд недостатков, часть из которых существенна для учебного процесса и «долгоиграющих» мероприятий, таких как проводимая на протяжении десятилетий ежегодная олимпиада. Наиболее очевидный недостаток – это полная зависимость от поставщика услуги. Если поставщик резко изменит условия или вообще откажется от данного сер-

[7] Архив задач с автоматической системой проверки в испанском городе Вальядолиде. URL: http://acm.uva.es

виса, пользователь, в данном случае организатор олимпиады, ничего сделать не сможет. Он может даже потерять доступ к данным старых мероприятий. Также данная модель затрудняет доработку системы под нужды конкретного пользователя или конкретного мероприятия. Если организаторы олимпиады захотят поменять правила, они вынуждены обращаться к поставщику SaS, который может и не выполнить этот запрос.

Известна система автоматической проверки timus [8] Уральского государственного университета с архивом задач. Система используется для проверки олимпиад, при этом участникам предоставляется набор языков программирования: C, C++ и Pascal (Delphi). Эта система предоставляется по модели SaS и поэтому обладает недостатками, характерными для этой модели и изложенными ранее.

В последние годы широкую известность в России получила система Yandex Contest компании Яндекс [9]. Она также предоставляется на условиях SaS. С 2013 г. она используется для проведения ряда соревнований, а также в учебном процессе, например в курсах Школы анализа данных Яндекс. Тестирование проводится под ОС Linux, и ей присущи недостатки, изложенные при описании системы ejudge. Кроме того, тестирование проводится на виртуальных машинах, а не на выделенных физических машинах. Это удешевляет развертывание системы, но создает опасность ложных вердиктов «превышено время тестирования» из-за конкуренции процессорных ядер за доступ к ОЗУ и накладных расходов виртуализации. Система Yandex Contest имеет ограниченные возможности увеличения емкости, что наблюдалось на региональных этапах Всероссийской олимпиады школьников по информатике 2017–2018 г. в тех регионах, где соревнования проводились на Yandex Contest. Время ожидания вердикта системы было довольно продолжительным. Возможно, это связано с архитектурными ограничениями на количество или размещение тестирующих машин.

Система NSUts

С помощью автоматизированной системы NSUts [10] проводятся соревнования по правилам ACM-ICPC и по правилам Всероссийской олимпиады школьников. Система поддерживает тестирование программ на языках C, C++, Java, Pascal, Pyton, Kotlin, C# для Win32. В рамках одного тура можно проводить проверку программ на нескольких диалектах одного языка, например Visual C или Borland C, так как компилятор языка программирования явно указывается участником при отправке задания. Для отправки задач участники используют веб-интерфейс системы, в котором можно отслеживать состояние тестирования, просматривать рейтинг, а также «общаться» с жюри. Для хранения задач, очереди тестирования и рейтинга используется реляционная СУБД. Заметим, что сохраняются все попытки отправки решений, все исходные тексты и точные значения времени их отправки. Это важно для проведения апелляций и для выявления попыток мошенничества. Время исполнения задачи в многозадачной среде определяется сложными сценариями конкуренции за разные ресурсы, такие как оперативная память, кэш процессора, время переключения контекста процессора и т. д. Поэтому проверка решений проводится на выделенных компьютерах, что сводит влияние такой конкуренции к минимуму.

К автоматизированным системам тестирования знаний предъявляются следующие требования:

- обеспечение глубокого и адекватного тестирования знаний и навыков;
- эффективная защита от мошенничества [10];
- простота в эксплуатации для специалистов средней квалификации.

В системе NSUts [6] можно выделить три основные составляющие:

- сервер олимпиад;

[8] Архив задач с автоматической системой проверки в Екатеринбурге. URL: http://acm.timus.ru
[9] Система Яндекс Контест. URL: https://contest.yandex.ru/
[10] Система NSUts. URL: https://olympic.nsu.ru/nsuts-new/login.cgi

- клиентское программное обеспечение;
- тестирующий клиент.

Главной частью системы является сервер. Тестирующих клиентов в системе может быть несколько. К клиентскому программному обеспечению относятся браузер и среда разработки для создания и отладки решений. Оно должно быть установлено на компьютере пользователя, и не предоставляется системой, но принимает активное участие в работе.

Олимпиада – это основной объект, над которым производятся действия сервера. Она может включать несколько туров. Каждый тур – это «олимпиада» с точки зрения участников, которая состоит из набора задач, представляющихся им для решения. Сервер тестирования осуществляет проведение олимпиады (тура). Он предоставляет участникам и членам жюри веб-интерфейс для взаимодействия с системой тестирования, автоматизирует управление олимпиадой (туром).

Под управлением олимпиадой понимается:

- осуществление хранения в базе данных условий задач, тестов и решений;
- проверка решений участников и визуализация ее результатов;
- создание и отображение рейтинга мест, которые заняли участники соревнования;
- генерация отчетов о проведении олимпиады (тура);
- предоставление средств для перетестирования решений участников;
- поддержка системы регистрации участников, обеспечение обратной связи с жюри и решение других организационных задач;
- администрирование олимпиад и туров.

Тестирующий клиент забирает решения участников, полученные сервером, из очереди решений, и выполняет их проверку на наборе тестов. В процессе обработки решения он получает исходный код, информацию о требуемом компиляторе и тестовые данные. Результат проверки решения передается обратно серверу, который производит его обработку, включающую отправку его участнику и создание рейтинга. Взаимодействие с системой участников олимпиад и членов жюри осуществляется посредством веб-интерфейса, который исключает обращение к тестирующему клиенту напрямую.

Тестирующие клиенты взаимодействуют с сервером через протокол HTTP, поэтому они могут быть размещены как на той же физической машине, что и сервер, так и на выделенных машинах, физических или виртуальных. В НГУ для тестирования олимпиад используются тестирующие клиенты на выделенных физических машинах.

Проведем сравнение системы NSUts, например, с системой Yandex Contest.

1. Система NSUts развернута на инфраструктуре разработчиков, и ее исходный код доступен НГУ. Это позволяет планировать мероприятия, в том числе статусные или просто ответственные, без согласований с внешними для НГУ организациями. При проведении ответственного мероприятия на системе Yandex Contest необходимо это согласовывать с компанией Yandex, так как в ней на это время могут быть запланированы профилактические или другие работы, закрывающие или ограничивающие доступ. Изменять правила соревнований, добавлять или исключать языки программирования невозможно без согласования с внешними организациями.

На Открытой Всесибирской олимпиаде им. И. В. Поттосина традиционно проводится первый тур по нестандартным правилам. «Стандартные» правила ACM хоть и медленно, но также изменяются. Языки и версии компиляторов требуется добавлять каждый год. Нужно иметь возможность дорабатывать и адаптировать систему в любом желательном для НГУ направлении, например добавить MathLab для приема задач у физиков.

2. В системе NSUts тестирование происходит на выделенных компьютерах под операционной системой Windows. Это решение обеспечивает адекватное судейство, связанное, во-первых, с точным подсчетом процессорного времени, поскольку правила большинства мероприятий предусматривают ограничение времени на исполнение программы. Во-вторых, обеспечивается возможность проводить школьные олимпиады и использовать компиляторы, доступные только под Windows (Visual Studio, Delphy).

Yandex Contest использует виртуальные машины с операционной системой Linux. У используемых компиляторов есть аналоги под систему Linux (Mono для C#, Free Pascal), но все-таки сами компиляторы и среды исполнения не тождественным тем, что указаны в правилах соревнований, а это дает основания для апелляций и иногда приводит к реальным проблемам.

3. В системе NSUts хранится история олимпиад, проводившихся в НГУ с 2010 г.

История знает немало случаев, когда публичные сервисы внезапно закрывались или изменяли политику до неприемлемой, самый известный пример – Sourceforge, который после смены собственника начал подкладывать программные модули, показывающие рекламу (adware), в размещенные на нем программные проекты. Для многолетней деятельности этот риск вполне реалистичен и неприемлем.

4. Yandex Contest заявляют одновременную поддержку 1 000 пользователей, сервер NSUts тоже на такую нагрузку регулярно тестируется.

Опыт использования системы NSUts

Как было отмечено, система NSUts создана для обеспечения проверки решений участников олимпиад по программированию, поэтому главной частью ее использования является проведение олимпиад по программированию всех уровней, в том числе открытых интернет-олимпиад. Она используется при проведении Открытой Всесибирской олимпиады по программированию им. И. В. Поттосина [11], районных и региональных школьных олимпиад по программированию в Новосибирской области [11]. В первых этапах Всероссийской олимпиады школьников по информатике, проводимых в Новосибирской области с помощью системы NSUts, одновременно участвовало несколько сотен школьников, было проверено свыше тысячи решений. Некоторые студенческие олимпиады собирают около 1 000 участников, при этом проверяется в онлайн-режиме более 10 000 решений в течение одного тура.

Приведем некоторые цифры по Открытой Всесибирской олимпиаде по программированию им. И. В. Поттосина. Олимпиада является командной, каждая команда состоит из трех участников. Имея один компьютер, команда стремится сдать как можно больше задач. Олимпиада проводится с 2000 г. Первые годы эта олимпиада проводилась в три тура. Первые два тура были отборочными с помощью Интернета, третий – очный. В последние годы олимпиада проводится в два тура: первый, отборочный, интернет-тур проходит заочно по традиционным правилам международного студенческого чемпионата ACM-ICPC. Второй тур – очный, состоит из двух номинаций. Первая номинация проводится по правилам, напоминающим Marathon Match на TopCoder [12]. В нем предлагается одна игровая или исследовательская задача на 5 часов работы. Как правило, для него жюри разрабатывает оригинальный проект, а также интерфейс и модули, которые позволяют встраивать решения участников в интерфейс, визуализировать и проигрывать их на сайте олимпиады. Для проведения этой номинации выполняется специальная настройка автоматизированной системы тестирования NSUts. Вторая номинация проводится по правилам международного чемпионата ICPC-ACM, а результат определяется по сумме двух номинаций.

30 сентября 2018 г. прошел интернет-тур XIX Олимпиады [13]. В нем участвовало 411 команд более чем из 100 университетов и 100 школ 21 страны. Результаты тура приведены на сайте олимпиады.

По правилам ACM-ICPC задача считается непринятой до тех пор, пока не пройдет все тесты жюри. Поэтому многие команды в течение тура на одну задачу отправляют на проверку несколько решений. Всего во время тура было отправлено около 4 000 решений.

[11] Сайт олимпиад Новосибирского государственного университета. URL: https://olympic.nsu.ru/

[12] URL: https://www.topcoder.com/community/data-science/how-to-compete

[13] XIX Открытая Всесибирская олимпиада по программированию им. И. В. Поттосина. URL: https://olympic.nsu.ru/widesiberia/2018/news

Очный тур прошел с 17 по 19 ноября 2018 г. В нем приняло участие 40 команд из Абакана, Барнаула, Екатеринбург, Ижевска, Иркутска, Кирова, Красноярска, Москвы, Новосибирска, Омска, Перми, Рубцовска, Санкт-Петербурга, Томска, Тюмени, Якутска.

I номинация была проведена 17 ноября 2018 г. Команды получили игровую задачу. В данной задаче требовалось написать программу «Умный Пилот» для управления космической ракетой. Цель полета – взлететь с поверхности Земли и выйти на эллиптическую орбиту с перигейным расстоянием не меньше заданного. Перигей – точка орбиты, ближайшая к центру Земли, а перигейное расстояние – расстояние от перигея до центра Земли. В условии задачи описано строение ракеты, ее управление и физика задачи, представлен протокол взаимодействия, описан статус ракеты и программа-интерактор, которая после статуса выдает текущее состояние ракеты. В архиве с игрой представлены все материалы, необходимые для работы с данной задачей, а также приведены примеры решений. Помимо описания задачи и протокола работы для нее представлены требования к программе, система оценки, генератор тестов.

Для тестирования первой номинации использовались средства NSUts. Участники сдавали свои решения через нее, результаты промежуточной проверки тоже отображались в системе. Однако проверяющая программа была реализована специально для данной задачи отдельно от системы. Вторая номинация была проведена 18 ноября 2018 г. Результаты обеих номинаций и общие результаты тура представлены на сайте олимпиады. Статистика за последние одиннадцать лет – в табл. 1.

Таблица 1

Статистика Открытой Всесибирской олимпиады им. И. В. Поттосина
(2008–2018 гг.)

Table 1

Pottosin Open Widesiberia Olympiad Statistics
(2008–2018)

Год	Количество команд интернет-тура	Количество участников (округленно)	Количество команд – участниц очного тура	Количество вузов + школ	Количество стран
2008	274	820	46	67 + 14	7
2009	272	820	48	98 + 30	7
2010	259	760	49	96 + 23	7
2011	301	870	49	95 + 29	7
2012	253	850	48	78 + 21	6
2013	228	680	51	82 + 12	8
2014	341	1000	48	98 + 74	7
2015	372	Более 1 000	50	96 + 39	12
2016	393	Более 1 000	47	96 + 24	12
2017	379	Более 1 000	46	101 + 27	31
2018	411	Более 1 000	42	106 + 11	22

В течение последних семи лет система NSUts активно используется преподавателями НГУ для промежуточного контроля заданий студентов по дисциплине «Программирование».

Отметим ряд свойств системы NSUts, которые дали возможность использовать ее в учебном процессе.

1. В NSUts реализована двухуровневая система организации данных. На первом уровне выделены в качестве основных объектов олимпиады, а на втором, внутри олимпиад, – туры. Каждый тур содержит некоторое количество задач. Применительно к учебному процессу под

олимпиадами можно понимать годовой или семестровый практический курс, а каждый тур рассматривать как недельное задание, состоящее из задач, объединенных темой, которая разбирается на семинарских занятиях. Регистрация на такую «олимпиаду» ограничена студентами одной группы, таким образом обеспечивается некоторая конфиденциальность информации.

2. Студенты могут сдавать задачи в систему в любое время суток, что поддерживает их самостоятельную работу.

3. Набор задач каждого тура вместе с тестами подготавливается заранее и может использоваться в течение многих лет. Решения студентов проверяются на этих тестах, что значительно экономит время преподавателя при проверке заданий. На занятиях остается больше времени для обсуждения методов кодирования алгоритмов или правил оформления текстов программ. Интерфейс системы NSUts предоставляет преподавателю средства, позволяющие проверять задания студентов удаленно и во внеурочное время. Заметим, что автоматизированная проверка программ не может исключить личного общения преподавателя со студентом. Преподаватель, например, может не зачесть решение студента, которое прошло все тесты и получило вердикт «принято», если посчитает, что студент не до конца понимает суть решаемой задачи, не может ее объяснить или решил задачу несамостоятельно.

4. Еженедельную работу студентов можно стимулировать системой баллов, которые непосредственно влияют на общую оценку. При этом важно учитывать время сдачи задания, оно хранится в системе NSUts для каждой посылки. Объявленные заранее правила получения оценок порождают стремление студентов сдать все задачи в кратчайшие сроки, что, несомненно, идет на пользу обучению.

5. Использование автоматизированной проверки задач способствует формализации отношений между преподавателем и студентом, практически исключает необъективность и предвзятость.

Рассмотрим статистику использования NSUts на первом курсе ФИТ НГУ за шесть лет (табл. 2). В первом семестре требовалось выполнять по одному заданию в неделю, всего 15 заданий. В каждое задание входило от трех до восьми задач. Всего за семестр нужно было сдать 60 задач различного уровня сложности. При успешной сдаче решения задачи в систему в течение первой недели со дня выдачи задания за нее начисляется 10 баллов, а далее баллы понижаются. При сдаче программы в первый день после соответствующего семинара за нее начисляется еще один бонусный балл.

Таблица 2

Результаты работы студентов за семестр

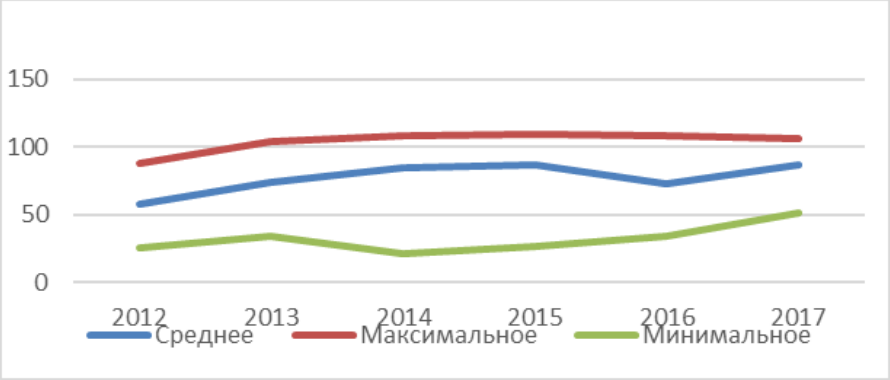
Table 2

Students Work Statistics for Semester

	Количество баллов, набранное студентами за семестр						Количество задач, решенных студентами за семестр, из 60 возможных					
	2012	2013	2014	2015	2016	2017	2012	2013	2014	2015	2016	2017
Среднее	350	442	507	519	440	520	39	49	54	53	48	53
Максимальное	537	621	648	653	647	637	52	59	60	60	60	60
Минимальное	157	205	128	159	202	304	24	28	31	35	24	34

Студенту выставляется отличная оценка, если он набирает не менее 75 % от базового количества баллов, что составляет не менее 450 баллов. Из приведенной в табл. 2 статистики видно, что среднее количество баллов, которое набирали студенты, очень близко к этой циф-

ре, а в половине случаев даже ее превосходит. Максимальное же количество баллов значительно превосходит 600. Это говорит о том, что в каждой группе были студенты, которые стремились все сдать в первый день и получить за каждую задачу по 11 баллов. Процентное соотношение набранных баллов по годам показано на рисунке.



Процент от возможных 600 баллов, набранный студентами за семестр
The Percentage of Possible 600 Points Scored by Students per Semester

Поскольку система NSUts изначально создавалась не для использования ее в учебных занятиях, в ней не была предусмотрена соответствующая функциональность. Отметим некоторые ее особенности, которые хотелось бы модифицировать и развить.

1. Большой проект, состоящий из нескольких модулей, в данной системе проверить нельзя, так как решение участника для проверки записывается в одном файле.

2. Для усвоения некоторых структур данных, таких как списки, стеки, очереди, деревья, необходимо научиться эти структуры обрабатывать. Для этого часто требуется уметь писать функции, которые для построенных структур должны выдавать результат преобразования. В системе NSUts, как и в любой другой системе проверки решений, этого сделать нельзя, поскольку входные и выходные данные для решения должны быть записаны в файлы. Следовательно, студенту дополнительно нужно строить саму структуру, что является отдельной задачей.

3. Никакая система не может заменить преподавателя, который должен проверять код студента. Иначе может оказаться, что система при проверке решения выдала вердикт «принято», но задача решена не та, которая была поставлена.

4. В системе рейтинг участников строится в туре. Построение общего рейтинга по всем турам осуществимо только внешними средствами. Автоматизация процесса построения такого рейтинга была бы желательна.

5. В олимпиадах по программированию доступ к набору тестов имеет только жюри, в учебном процессе в этой роли выступает преподаватель. Было бы полезно давать возможность студентам видеть тесты, это часто помогает отладить программу.

Заключение

Многолетний опыт использования нескольких поколений систем автоматической оценки заданий по программированию [6] позволил разработать мощное и гибкое решение, применимое для широкого спектра мероприятий: проведения олимпиад, практикумов по программированию, тренировок олимпиадных команд и др. Но в процессе работы определились направления развития системы. Одним из них является адаптация системы к требованиям учебного процесса. Хотелось бы, например, реализовать в системе поддержку для различных стратегий обучения. Также имеется большая потребность создать и поддерживать банк задач

без привязки к конкретной олимпиаде или туру. Это связано с тем, что накопилось большое количество задач разных уровней сложности и тематики, все они хранятся в своих турах, что усложняет их поиск.

В настоящее время ведутся работы по переходу сервера на клиент-серверную архитектуру AJAX [12]. Большая часть системы реализована по технологии динамического HTML, когда сервер полностью генерирует код страницы и передает ее клиентскому браузеру на каждый запрос. При характерной для NSUts нагрузке, когда разные пользователи в большинстве случаев видят разные данные, это приводит к тому, что сервер делает много лишней работы, и невозможно использовать кэширование. AJAX – подход к разработке современных веб-сайтов, используемый ведущими интернет-сервисами Gmail, Facebook, Vkontakte и др., состоящий в том, что сервер генерирует только данные в компактном формате, например в формате JSON, а генерация и отрисовка всего пользовательского интерфейса производится на клиентском браузере при помощи JavaScript. Это позволяет решить проблемы с «лишней работой» для сервера и кэшированием, задействовать процессоры клиентских компьютеров (когда сервер один, а клиентских компьютеров сотни, преимущества кажутся очевидными), а также ввести ряд полезных функций, уже реализованных в других системах: автообновление очереди и рейтинга, оповещение об изменениях рейтинга.

Существующая система уже начинает испытывать нехватку ресурсов, если в ней оказывается больше 1 000 пользователей. Созданы прототипы новой архитектуры, которые обеспечивают приемлемую производительность вплоть до 2–3 тысяч пользователей. В результате модернизации системы ожидается повышение ее производительности под высокими нагрузками, кратное повышение емкости, а также снижение потребности сервера в ОЗУ в десятки или сотни раз, по сравнению с текущими 16 Gb.

Система NSUts постоянно развивается, предполагаются дополнительные направления ее доработки, такие как выкладывание исходного кода системы в открытый доступ, улучшение дизайна системы и ее эргономики.

Список литературы / References

1. **Reek K. A.** The TRY System – or – How to Avoid Testing Student Programs. *Proceedings SIGCSE Bulletin*, February 1989, vol. 21, no. 1, p. 112–116.
2. **Jackson D., Usher M.** Grading Student Programs Using ASSYST. In: Proceedings SIGCSE '97, 1997, p. 335–339.
3. **Jones E. L.** The SPRAE Framework for Teaching Software Testing in the Undergraduate Curriculum. In: Proceedings ADMI 2000, June 1–4. Hampton, VA USA, 2000.
4. **Edwards S.** Using Software Testing to Move Students from Trial-and-Error to Reflection-in-Action. AMC Press, 2004, p. 26–30.
5. **Enstrom E., Kreitz G., Niemela F., Soderman P., Kann V.** Five years with kattis – Using an automated assessment system in teaching. In: Proceedings of the 2011 Frontiers in Education Conference (FIE '11). IEEE Computer Society. Washington, DC, USA, p. T3J-1–T3J-6. DOI 10.1109/FIE.2011.6142931
6. **Иртегов Д. В., Нестеренко Т. В., Чурина Т. Г.** Разработка систем автоматизированной оценки заданий по программированию // Системная информатика. 2017. № 11. С. 91–116. DOI 10.31144/si.2307-6410.2017.n11

   Irtegov D. V., Nesterenko T. V., Churina T. G. Development of Automated Evaluation Systems for Programming Tasks. *System Informatics Journal*, 2017, no. 11, p. 91–116. (in Russ.) DOI 10.31144/ si.2307-6410.2017.n11

7. **Боженкова Е. Н., Иртегов Д. В., Киров А. В., Нестеренко Т. В., Чурина Т. Г.** Автоматизированная система тестирования NSUts: требования и разработка прототипа // Вестник НГУ. Серия: Информационные технологии. 2010. Т. 8, № 4. С. 46–53.

Bozhenkova E. N., Irtegov D. V., Kirov A. V., Nesterenko T. V., Churina T. G. NSUts Automated Testing System: Prototype Requirements and Development. *Vestnik NSU. Series: Information Technologies*, 2010, vol. 8, no. 4, p. 46–53. (in Russ.)

8. **Чурина Т. Г., Иртегов Д. В.** Требования к автоматической системе тестирования знаний // Тр. VI Междунар. конф. «Интеллектуальные технологии в образовании, экономике и управлении». Воронеж, 2009, С. 309–317.

Churina T. G., Irtegov D. V. Requirements for an automated knowledge testing system. In: Proceedings of the VI International Conference “Intellectual Technologies in Education, Economics and Management”. Voronezh, 2009, p. 309–317. (in Russ.)

9. **Корнеев Г. А., Елизаров Р. А.** Автоматическое тестирование решений на соревнованиях по программированию // Телекоммуникации и информатизация образования. 2003. № 1. С. 61–73.

Korneev G. A., Elizarov R. A. Automatic testing of solutions in programming competitions. *Telecommunications and education informatization*, 2003, no. 1, p. 61–73. (in Russ.).

10. **Боженкова Е. Н., Воронков А. Д., Иртегов Д. В., Конышева Е. Н., Черненок С. А., Чурина Т. Г.** Модель разграничения прав доступа в системе автоматизированной проверки корректности программных приложений // Вестник НГУ. Серия: Информационные технологии. 2011. Т. 9, № 4. С. 79–85.

Bozhenkova E. N., Voronkov A. D., Irtegov D. V., Konysheva E. N., Chernenok S. A., Churina T. G. Model of access rights delimitation in the system of automated verification of software applications correctness. *Vestnik NSU. Series: Information Technologies*, 2011, vol. 9, no. 4, p. 79–85. (in Russ.)

11. **Чурина Т. Г., Боженкова Е. Н., Нестеренко Т. В.** Задачи Открытой Всесибирской олимпиады по программированию имени И. В. Поттосина: от теории к практике // Вестник НГУ. Серия: Информационные технологии. 2007. Т. 5, № 1. С. 40–46.

Churina T. G., Bozhenkova E. N., Nesterenko T. V. Problems of the Pottosin Open Widesiberia Olympiald: from theory to practice. *Vestnik NSU. Series: Information Technologies*, 2007, vol. 5, no. 1, p. 40–46. (in Russ.)

12. **Боженкова Е. Н., Иртегов Д. В., Колбин Я. С.** Оптимизация производительности веб-интерфейса приложения NSUts средствами динамического HTML // Вестник НГУ. Серия: Информационные технологии. 2015. Т. 13, № 2. С. 13–21.

Bozhenkova E. N., Irtegov D. V., Kolbin Ya. S. Optimizing the performance of the NSUts web interface using dynamic HTML. *Vestnik NSU. Series: Information Technologies*, 2015, vol. 13, no. 2, p. 13–21. (in Russ.)

Материал поступил в редколлегию
Received
15.01.2019

Сведения об авторах / Information about the Authors

Иртегов Дмитрий Валентинович, доцент, заведующий лабораторией, факультет информационных технологий, Новосибирский государственный университет (ул. Пирогова, 1, Новосибирск, 630090, Россия)

Dmitry V. Irtegov, Associate Professor, Novosibirsk State University (1 Pirogov Str., Novosibirsk, 630090, Russian Federation)

fat@nsu.ru

Нестеренко Татьяна Викторовна, научный сотрудник, Институт систем информатики им. А. П. Ершова СО РАН (пр. Академика Лаврентьева, 6, Новосибирск, 630090, Россия), доцент, факультет информационных технологий, Новосибирский государственный университет (ул. Пирогова, 1, Новосибирск, 630090, Россия)

Tatyana V. Nesterenko, Researcher, A. P. Ershov Institute of Informatics Systems SB RAS (6 Academician Lavrentiev Ave., Novosibirsk, 630090, Russian Federation), Associate Professor, Novosibirsk State University (1 Pirogov Str., Novosibirsk, 630090, Russian Federation)

nest@iis.nsk.su

Чурина Татьяна Геннадьевна, кандидат физико-математических наук, доцент, старший научный сотрудник, Институт систем информатики им. А. П. Ершова СО РАН (пр. Академика Лаврентьева, 6, Новосибирск, 630090, Россия), доцент, факультет информационных технологий, Новосибирский государственный университет (ул. Пирогова, 1, Новосибирск, 630090, Россия)

Tatyana G. Churina, Senior Researcher, A. P. Ershov Institute of Informatics Systems SB RAS (6 Academician Lavrentiev Ave., Novosibirsk, 630090, Russian Federation), Associate Professor, Novosibirsk State University (1 Pirogov Str., Novosibirsk, 630090, Russian Federation)

tanch@iis.nsk.su