

АВТОМАТИЗАЦИЯ ПРОВЕРКИ ПРАКТИЧЕСКИХ ЗАДАНИЙ В РАМКАХ ОБУЧЕНИЯ НА КУРСЕ «ВЕБ-ПРОГРАММИРОВАНИЕ PHP»

Готская И. Б.^{1,2}, доктор педагогических наук, ✉ iringot@mail.ru
Муромцев Д. И.², кандидат технических наук, доцент, ✉ mouromtsev@itmo.ru
Государев И. Б.^{1,2}, кандидат педагогических наук, доцент, ✉ goss@corp.ifmo.ru
Сивинский С. А.², аспирант, ✉ stanislav_sivin@mail.ru

¹ Российский государственный педагогический университет им. А.И. Герцена,
набережная реки Мойки, д. 48, 191186, Санкт-Петербург, Россия

² Санкт-Петербургский национальный исследовательский университет ИТМО,
Кронверкский пр., 49, лит. А, 197101, Санкт-Петербург, Россия

Аннотация

В работе рассмотрены основные аспекты построения системы автоматизированного тестирования с целью проверки практических и самостоятельных работ слушателей в рамках изучения дисциплины «Веб-программирование PHP» на базе образовательного центра Компьютерного проектирования и дизайна, Университет ИТМО. Основная идея заключается в применении модульного тестирования (юнит тестирование) для проверки практических и самостоятельных работ. В статье рассмотрена архитектура построения взаимодействия между системой дистанционного обучения Moodle, системой тестирования IServer и системой контейнеризации Docker. Применение системы контейнеризации docker позволяет достигнуть безопасного исполнения кода слушателей без риска взлома системы тестирования и нарушения учебного процесса.

Ключевые слова: *PHPUnit, модульное тестирование, юнит-тестирование, веб-программирование, PHP, автоматизация тестирования.*

Цитирование: Готская И. Б., Муромцев Д. И., Государев И. Б., Сивинский С. А. Автоматизация проверки практических заданий в рамках обучения на курсе «Веб-программирование PHP» // Компьютерные инструменты в образовании. 2020. № 4. С. 69–78. doi: 10.32603/2071-2340-2020-4-69-78

1. ВВЕДЕНИЕ

Переход к электронному обучению с применением систем дистанционного обучения (СДО) и систем организации вебинаров обусловил необходимость совершенствования средств программной поддержки деятельности преподавателя, нагрузка на которого заметно возросла. Реализуя образовательный процесс в полностью цифровой форме, преподаватель должен выполнять целый ряд задач, включая: организацию и проведение вебинаров, индивидуальные консультации по теоретическому материалу и практическим работам, проверку практических и самостоятельных работ, а также разработку

и поддержку в актуальном состоянии учебных материалов изучаемого курса (последнее является ключевым моментом для курсов дополнительного профессионального обучения).

Как показывает опыт авторов данной статьи, существенная доля рабочего времени преподавателя распределяется на обеспечение взаимодействия с обучающимися, направленного на своевременную обратную связь, проверку результатов выполнения ими заданий (в данном случае — по программированию на языке PHP) и корректировку их индивидуальных маршрутов.

В рамках изучения дисциплин, связанных с программированием, возможна организация автоматизации проверки практических и самостоятельных работ за счет возможностей СДО и внешних учебных тренажеров.

На сегодняшний день существуют множество решений для организации автоматической проверки заданий по программированию: Яндекс.Контест (contest.yandex.ru), Школа Программиста (acmp.ru), TopCoder (topcoder.com) и другие. Самым главным недостатком большинства систем является сложность организации взаимосвязи между СДО и учебным тренажером, а также порой невозможность создания собственных задач по программированию.

Отдельно следует отметить наличие модулей (плагинов) системы Moodle, реализующих запуск программного кода, являющегося ответом обучающегося на задачу по программированию на том или ином языке. Обычно используется модуль CodeRunner. У таких модулей имеется очевидное преимущество высокой интеграции с основной LMS.

Тем не менее с точки зрения современной архитектуры веб-приложений высокоинтегрированные решения являются источником проблем, связанных с масштабированием и отладкой. В самом деле, сильная связность компонентов системы влечет затруднения в устранении дефектов и наращивании мощности необходимого функционала. При проверке программного кода обучающихся потенциальные уязвимости могут создавать угрозу для всей системы. Поэтому востребованы решения со слабой связностью, тяготеющие к микросервисной архитектуре.

На сегодняшний день при разработке веб-приложений в формате отдельных сервисов лидирующее место занимает контейнеризационный подход. Удобно и целесообразно использовать систему Docker для управления исполнением кода приложений в отдельных контейнерах. Из этого исходили авторы описываемой в данной статье системы.

2. АРХИТЕКТУРА ТЕСТОВОЙ СИСТЕМЫ ISERVER

В рамках решения данной проблемы авторы предлагают систему организации автоматической проверки заданий в курсе «Веб-программирование PHP» [1] на базе образовательного центра Компьютерного проектирования и дизайна, Университет ИТМО (design.ifmo.ru), автор курса и ведущий преподаватель Сивинский Станислав Андреевич.

В соответствии с рисунком 1 выполнение практического задания включает в себя следующие основные этапы:

1. Авторизация в СДО.
2. Выбор и открытие учебного курса.
3. Переход и авторизация слушателя в учебном тренажере «IServer».
4. Ввод исходного кода в качестве ответа на практическое задание.
5. Запуск Docker-контейнера с системой проверки задания.

6. В случае успеха возвращение положительного результата теста в СДО, отметка о выполнении задания и сохранение оценки в журнале.
7. В случае провала прохождения теста вывод ошибок на дисплей слушателя и переход к этапу ввода исходного кода.

Следует отметить, что при повторной попытке отправки исходного кода на проверку этап запуска контейнера не выполняется, так как запущенный в первый раз контейнер работает до тех пор, пока пользователь не пройдет тест, или будет бездействовать в течение часа.

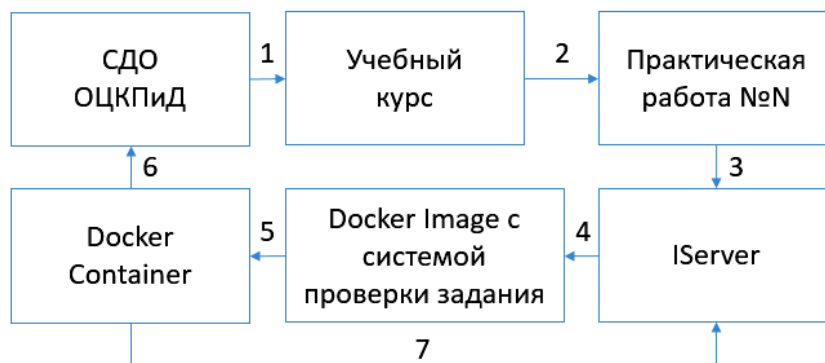


Рис. 1. Основные компоненты системы тестирования

Еще раз подчеркнем, что применение системы контейнеризации Docker обусловлено необходимостью организации безопасности исполнения потенциально опасного кода слушателей без влияния на работоспособность самой системы тестирования и возможностью логического либо физического отделения друг от друга компонентов среды обучения. Подход к построению системы представлен на рисунке 1.

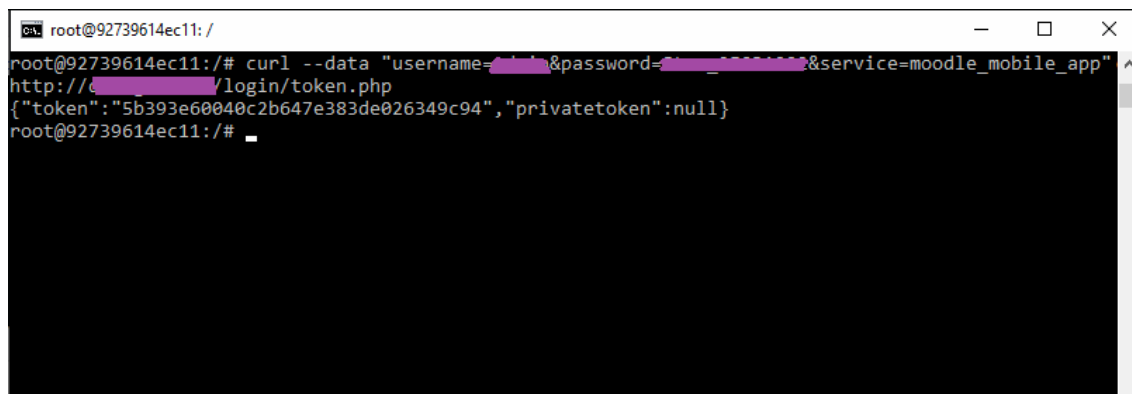
3. ОРГАНИЗАЦИЯ ВЗАИМОДЕЙСТВИЯ НА ОСНОВЕ ВЕБ-СЕРВИСА LMS MOODLE

СДО образовательного центра кафедры проектирования и дизайна (ОЦКПид) работает на основе LMS Moodle (moodle.org). Система Moodle, начиная с версии 2.0, предлагает протокол REST для работы с внешними приложениями, и с версии 2.2 поддерживается JSON формат получения данных [2].

Взаимодействие внешней системы тестирования I Server и LMS Moodle базируется на использовании веб-службы «Moodle mobile web service». Список доступных сервисных функций покрывает большинство возникающих задач при работе с системой дистанционного обучения.

Важно отметить, что для использования сервисных функций веб-службы «Moodle mobile web service» [3] в качестве обязательного get-параметра для корректного исполнения запроса от I Server выступает параметр wstoken. Данный параметр содержит значение ключа (токена) для автоматической авторизации пользователя, от которого пришел запрос.

Токен можно создавать как вручную в разделе «Администрирование — Плагины — Веб-службы — Управление ключами», так и автоматически за счет получения токена через осуществление запроса в соответствии с рисунком 2.



```
root@92739614ec11: /
root@92739614ec11:/# curl --data "username=[REDACTED]&password=[REDACTED]&service=moodle_mobile_app"
http://[REDACTED]/login/token.php
{"token":"5b393e60040c2b647e383de026349c94","privatetoken":null}
root@92739614ec11:/#
```

Рис. 2. Получение токена от системы

4. МОДУЛЬНОЕ ТЕСТИРОВАНИЕ НА ОСНОВЕ PHPUNIT

После успешной авторизации слушателю необходимо ввести в редактор кода свой ответ и подтвердить отправку нажатием на управляющий элемент «Отправить ответ» в соответствии с рисунком 3.

Практическая работа №1. Введение в PHP

Задание 1.1 Создание переменной

Создайте переменную tmp и присвойте ей текстовое значение "Hello, World!". С помощью команды echo выведите ее.



```
1 <?php
2     $tmp = "Hello, world";
3     echo $tmp;
4 ?>
```

Отправить ответ

Рис. 3. Страница выполнения задания

Далее происходит запуск Docker-контейнера, содержащего систему проверки ответа, которая основана на применении фреймворка юнит-тестирования PHPUnit (<https://phpunit.de/>).

К достоинствам фреймворка PHPUnit можно отнести следующее: наличие инструментов для создания модульных тестов и организацию их в иерархические наборы, наличие интерфейса командной строки для взаимодействия с фреймворком, поддержка проведения тестов для участков кода, работающего с базой данных, тестирование исключений и генераторы отчетов (xml, html и текстовый формат).

Применение фреймворка PHPUnit напрямую влияет на методику формулировки практических заданий и самостоятельных работ.

Рассмотрим пример формулировки задания из темы №2 «Условные операторы языка

программирования PHP».

Задание звучит следующим образом: «Создайте программу управления температурой умного дома. Вам дана переменная t , содержащая в различные моменты времени значения температуры. Если температура в комнате ниже 18°C , тогда включаем обогреватель (Вывод: Обогреватель включен). Если температура в комнате выше 25°C , тогда включаем кондиционер (Вывод: Кондиционер включен). Если температура в комнате не ниже 18°C и не выше 25°C , тогда отключаем все приборы (Вывод: Приборы отключены). Используйте конструкцию `if..else` для проверки значений переменной t и принятия решений. Для вывода текстового значения принятого решения используйте команду `echo`».

В данном случае формулировка задания предполагает наличие заранее созданной переменной t для задания числового значения температуры во время тестирования программы и получения результата в виде вывода на странице. Наряду с формулировкой задания слушателю доступен пример входных и выходных данных, в соответствии с рис. 4.

Создайте программу управления температурой умного дома. Вам дана переменная t , содержащая в различные моменты времени значения температуры.

Если температура в комнате ниже 18°C , тогда включаем обогреватель (Вывод: Обогреватель включен).

Если температура в комнате выше 25°C , тогда включаем кондиционер (Вывод: Кондиционер включен).

Если температура в комнате выше или равна 18°C и ниже или равна 25°C , тогда отключаем все приборы (Вывод: Приборы отключены).

Используйте конструкцию `if..else` для проверки значений в переменной t и принятия решений. Для вывода текстового значения принятого решения используйте команду `echo`.

Ввод	Вывод
12	Обогреватель включен
32	Кондиционер включен
20	Приборы отключены
hi	Неверный ввод

Рис. 4. Формулировка задания по теме «Условные операторы»

Также следует отметить, что в данном случае пользователю нужно следить за точностью вывода информации, так как в случае опечатки в текстовом значении вывода результата работы программы ответ будет считаться неверным.

Проверяющий сценарий запускает предложенное обучающимся решение, передавая различные значения t и сопоставляя выдаваемый ответ с образцом. В случае успешного выполнения задания система тестирования произведет вызов внешней функции от веб-службы «Moodle mobile web service» и отправит результат тестирования для сохранения в базе данных, таким образом преподавателю всегда будет доступен исходный код ответа слушателя, который он прислал. После сохранения результата тестирования происходит вызов внешней функции, которая позволяет программным способом выставить оценку за задание.

5. ЗАКЛЮЧЕНИЕ

К недостаткам системы «IServer» можно отнести сложность разработки практических и самостоятельных работ, которая требует от преподавателя навыков работы с PHPUnit и написания тестов для осуществления тестирования исходного кода студентов, но данный недостаток окупается экономией времени преподавателя на проверку практических работ.

В то же время архитектура системы нацелена на возможность масштабирования и независимого использования, что является несомненным преимуществом, особенно в условиях роста числа обучающихся в дистанционном режиме. В течение срока эксплуатации внедрённой системы преподавателями отмечено увеличение числа положительных отзывов слушателей. Количество договоров на обучение также возросло по сравнению с предыдущим периодом. Рост чистой прибыли от функционирования ОЦ КПиД приблизительно оценивается в 15–20 % по сравнению с периодом, когда использовалась преимущественно ручная проверка и, частично, CodeRunner.

Для оценки эффективности применения внедренной системы был проведен анализ скорости обработки ответов слушателей.

Для анализа были взяты две группы:

Группа А — это группа, в которой обучение проводилось стандартными средствами без применения описываемой в статье системы автоматической проверки ответов.

Группа Б — это контрольная группа, в которой обучение проходило с применением автоматической проверки ответов слушателей.

Для измерения был выбран промежуток с 1 сентября 2019 года по 30 июня 2020 года по одной из тем курса Web-программирование PHP. В соответствии с таблицей 1 приведены данные в виде временных промежутков, отображающих скорость проверки заданий от поступления ответа в систему до выставления оценки преподавателем. Следует отметить тот факт, что система проверки не предоставляет финальную оценку, для утверждения которой ответ еще должен проанализировать преподаватель.

Для оценки результатов эксперимента был выбран критерий Манна-Уитни. В соответствии с требованиями данного критерия все промежутки были отсортированы и проранжированы.

В соответствии с формулой 1 был проведен расчет эмпирического значения критерия.

$$U_{\text{эмп}} = (n_1 \cdot n_2) + n_x \cdot \frac{(n_x + 1)}{2} - R_{\text{max}}, \quad (1)$$

где n_1 и n_2 — это объем первой и второй выборок, R_{max} — наибольшая по величине сумма рангов, n_x — количество измерений в группе с наибольшей суммой рангов.

Сформулируем статистические гипотезы:

H_0 — Время проверки ответов слушателей с применением системы IServer не превосходит время проверки без применения системы. H_1 — Время проверки ответов слушателей с применением системы достоверно превосходит время проверки без применения системы.

На рисунке 5 приведена ось значимости, а также данные для эмпирического значения критерия и для критических значений при уровнях значимости 0,05 и 0,01.

Исходя из полученных результатов следует отвергнуть H_0 и принять гипотезу H_1 . Можно сделать вывод о том, что применение системы автоматической проверки заданий позволило повысить скорость обработки ответов слушателей.

Таблица 1. Скорость оценки ответов в анализируемых группах, мин.

Группа А	Ранг (группа А)	Группа Б	Ранг (группа Б)
37	2	12	1
67	6	46	3
175	11	59	4
185	12	61	5
198	14	78	7
207	15	88	8
262	19	102	9
465	24	116	10
526	25	194	13
549	28	235	16
619	31	246	17
666	34	250	18
672	35	282	20
728	38	306	21
811	39	349	22
812	40	357	23
852	42	540	26
885	43	544	27
959	44	553	29
1133	46	559	30
1136	47	641	32
1147	50	647	33
1342	54	689	36
1405	55	689	37
1487	57	826	41
1566	58	999	45
1688	59	1142	48
1820	60	1145	49
1924	61	1151	51
2043	62	1185	52
2139	63	1317	53
2154	64	1438	56
Сумма А	1238	Сумма Б	842

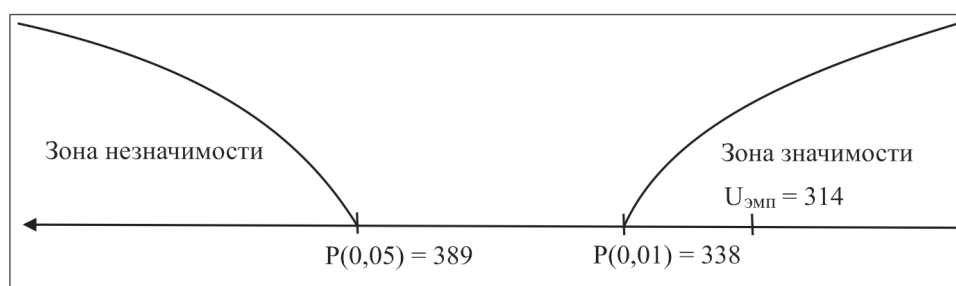


Рис. 5. Ось значимости критерия Манна-Уитни

Как показывает практика, обычно при использовании автоматизированных тестов в системе оставляют возможность ручной проверки и корректировки баллов, выставлен-

ных по результатам работы алгоритма. Если при апробации оказывается, что преподавателю приходится прибегать к ручной проверке настолько часто, что временные затраты начинают обесценивать внедрение автоматизированной проверки, то это является поводом пересмотреть формулировки заданий и/или конструкцию тестов; качественное проектирование тестов приводит к тому, что такая потребность возникает в единичных случаях. Например, такое возможно, если обучающийся использует какую-либо версию языка, которая не поддерживается системой проверки.

Следует также отметить, что на базе образовательного центра кафедры проектирования и дизайна (design.ifmo.ru) успешно функционирует система автоматизированного тестирования [4, 5], направленная на автоматизацию проверки практических заданий слушателей в рамках обучения на образовательных курсах «Веб-разработчик. От Middle до Senior» и «Frontend разработчик. От Middle до Senior», автор курсов и платформы Государев Илья Борисович.

Список литературы

1. Специалист в области компьютерной графики и веб-дизайна URL: https://design.ifmo.ru/courses/web_design_review/ (дата обращения: 29.10.2020).
2. Готская И. Б., Сивинский С. А. К проблеме оценки компетенций при реализации дополнительных профессиональных программ с использованием СДО Moodle // Письма в Эмиссия.Оффлайн (The Emissia.Offline Letters): электронный научный журнал. 2018. № 6 (июнь). ART 2627. URL: <http://www.emissia.org/offline/2018/2627.htm> (дата обращения: 29.10.2020).
3. Web service API functions. URL: https://docs.moodle.org/dev/Web_service_API_functions (дата обращения: 29.10.2020).
4. Государев И. Б., Флеров А. В., Перепелица Ф. А. Проблемы и принципы повышения квалификации дипломированных специалистов в области веб-технологий и компьютерной графики на платформе дистанционного обучения // Современные проблемы науки и образования. 2019. № 2. URL: <http://www.science-education.ru/ru/article/view?id=28651> (дата обращения: 29.10.2020).
5. λ-Проверят URL: <https://node-server.online/r> (дата обращения 29.10.2020).

Поступила в редакцию 27.05.2020, окончательный вариант — 29.10.2020.

Готская Ирина Борисовна, доктор педагогических наук, профессор кафедры компьютерных технологий и электронного обучения РГПУ им. А. И. Герцена; профессор кафедры компьютерного проектирования и дизайна Санкт-Петербургского национального исследовательского университета ИТМО, ✉ iringot@mail.ru

Муромцев Дмитрий Ильич, кандидат технических наук, доцент факультета программной инженерии и компьютерной техники Санкт-Петербургского национального исследовательского университета ИТМО, ✉ mouromtsev@itmo.ru

Государев Илья Борисович, кандидат педагогических наук, доцент кафедры компьютерных технологий и электронного обучения РГПУ им. А. И. Герцена; доцент факультета программной инженерии и компьютерной техники Санкт-Петербургского национального исследовательского университета ИТМО, ✉ goss@corp.ifmo.ru

Сивинский Станислав Андреевич, аспирант факультета программной инженерии и компьютерной техники Санкт-Петербургского национального исследовательского университета ИТМО, ✉ stanislav_sivin@mail.ru

Computer tools in education, 2020

№ 4: 69–78

<http://cte.eltech.ru>

doi:10.32603/2071-2340-2020-4-69-78

Automating the Verification of Practical Tasks in the Course “PHP Web Programming”

Gotskaya I. B.^{1,2}, Doctor of Pedagogical Sciences, Professor, ✉ iringot@mail.ru

Mouromtsev D. I.², PhD, ✉ mouromtsev@itmo.ru

Gosudarev I. B.^{1,2}, PhD, ✉ goss@corp.ifmo.ru

Sivinskiy S. A.², Postgraduate, ✉ stanislav_sivin@mail.ru

¹Herzen University, 48 Moika river embankment, 191186, Saint Petersburg, Russia

²ITMO University, 49 Kronverksky, bldg. A, 197101, Saint Petersburg, Russia

Abstract

The paper deals with the main aspects of designing a system of automated testing to test the practical and independent work of students in the discipline “PHP Web-programming” based on the educational center of Computer Engineering and Design, ITMO University. The primary idea is to use unit testing to evaluate practical and independent student works. The paper describes the architecture of interaction between LMS Moodle, IServer testing system and docker system. The use of docker system allows to achieve safe execution of the students’ code without risk of hacking the testing system and disrupting the learning process.

Keywords: *PHPUnit, unit test, web-programming, PHP, automate testing.*

Citation: I. B. Gotskaya, D. I. Mouromtsev, I. B. Gosudarev, and S. A. Sivinskiy, “Automating the Verification of Practical Tasks in the Course “PHP Web Programming”,” *Computer tools in education*, no. 4, pp. 69–78, 2020 (in Russian); doi:10.32603/2071-2340-2020-4-69-78

References

1. ITMO University, “Specialist in computer graphics and web-design. Course program,” in *design.itmo.ru*. [Online] (in Russian). Available: https://design.ifmo.ru/courses/web_design_review/
2. I. B. Gotskaya and S. A. Sivinskiy, “To the problem of competencies assessment in the implementation of additional professional programs using LMS Moodle,” *Emissia. Offline Letters*, no. 6, 2018 (in Russian).
3. “Web service API functions,” in *Moodle*. [Online] (in Russian). Available: https://docs.moodle.org/dev/Web_service_API_functions
4. I. B. Gosudarev, A. V. Flerov, and F. A. Perepelitsa, “The problems and principles of advanced training of the postgraduate students in the field of web technologies and computer graphics on the platform of distant learning,” *Modern problems of science and education*, no. 2, 2019 (in Russian); doi: 10.17513/spno.28651
5. I. B. Gosudarev, *λ-Proveriator*, [Soft]. [Online] (in Russian). Available: <https://node-server.online/r>

Received 27.05.2020, the final version — 29.10.2020.

Irina Gotskaya, Doctor of Pedagogical Sciences, Professor of the Computer technology and e-learning Department, Institute of Computer Science and Technological Education, Al.Herzen State Pedagogical University of Russia, St. Petersburg; Professor of the Program Engineering and Computer Technics Faculty, Saint-Petersburg National Research University of Information Technologies, Mechanics and Optics, St. Petersburg, ✉ iringot@mail.ru

Dmiiitii Mouromtsev, PhD, Candidate of technical sciences, Associate Professor of the Program Engineering and Computer Technics Faculty, Saint-Petersburg National Research University of Information Technologies, Mechanics and Optics, St. Petersburg, ✉ mouromtsev@itmo.ru

Ilya Gosudarev, PhD, Candidate of Pedagogical Sciences, Associate Professor of the Computer technology and e-learning Departmen, Institute of Computer Science and Technological Education, The Herzen State Pedagogical University of Russia, St. Petersburg; Associate Professor of the Program Engineering and Computer Technics Faculty, Saint-Petersburg National Research University of Information Technologies, Mechanics and Optics, St. Petersburg, ✉ goss@corp.ifmo.ru

Stanislav Sivinskiy, Postgraduate of the Faculty of Software Engineering and Computer Systems, Saint Petersburg National Research University of Information Technologies, Mechanics and Optics, St. Petersburg, ✉ stanislav_sivin@mail.ru