

РАЗРАБОТКА СИСТЕМЫ РЕКОМЕНДАЦИИ И АВТОМАТИЗАЦИИ РАЗВЕРТЫВАНИЯ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ НА УДАЛЕННЫХ ТЕРМИНАЛАХ*

Е.А. ОМЕЛЬЯНЕНКО

630073, РФ, г. Новосибирск, пр. Карла Маркса, 20, Новосибирский государственный технический университет, магистрант кафедры вычислительной техники. E-mail: egaom@mail.ru

Последние несколько десятилетий можно наблюдать быстрое развитие компьютерной техники и информационных технологий. Компьютеры настолько крепко вошли в нашу жизнь, что уже сегодня невозможно представить себе работу во многих отраслях (например, в архитектуре, банковском деле, медицине) без их использования.

С компьютеризацией новых отраслей и сфер жизни будет расти количество компьютеров на рабочих местах и количество программного обеспечения (ПО), специфичного для той или иной профессии. Также в большинстве случаев при работе используется не единственный инструмент, а некоторая их комбинация, которая может отличаться для специалистов в одной области и даже для специалистов, работающих над одним проектом. Так, например, при разработке веб-сайта разработчик серверной части будет использовать один набор ПО, а разработчик клиентской части – другой.

Из описанного выше встает проблема конфигурирования и установки программных продуктов в различных комбинациях для большого количества компьютеров (рабочих мест). В больших компаниях за это отвечают специализированные отделы инженеров, но среди мелкого и среднего бизнеса часто бывает невыгодно содержать для этих целей целый отдел или даже одного специалиста.

Описанные проблемы могли бы решить существующие системы управления конфигурациями, но они слишком сложны в освоении и использовании, поскольку разрабатывались для управления серверами и не могут быть использованы человеком без глубоких знаний в сфере информационных технологий. Разрабатываемая система решает описанные проблемы.

Ключевые слова: компьютеризация, автоматизация, конфигурирование, рекомендательная система, асинхронность, микросервисная архитектура, контейнеризация, Ansible

* Статья получена 15 ноября 2018 г.

ВВЕДЕНИЕ

Целью настоящей работы является разработка и тестирование системы рекомендации и автоматизации развертывания программного обеспечения на удаленных машинах.

Актуальность работы заключается в том, что с повсеместной компьютеризацией и ростом числа компьютеров на рабочих местах возникает необходимость в конфигурировании большого количества удаленных рабочих мест людьми без глубоких знаний в сфере информационных технологий.

Практическая ценность работы заключается в простоте, удобстве и гибкости использования разрабатываемой системы, что выгодно выделяет ее среди существующих решений. Данную систему сможет использовать широкий круг пользователей, что поспособствует ускорению автоматизации и компьютеризации различных сфер жизни.

Научная ценность работы заключается в комбинировании современных технологий разработки ПО, способов построения архитектуры приложения и приемов машинного обучения, а так же в исследовании рынка информационных технологий для выделения наиболее часто используемых программных продуктов и стеков-технологий при разработке информационных систем.

1. ВВЕДЕНИЕ В ПРЕДМЕТНУЮ ОБЛАСТЬ

Компьютеризация – широкое внедрение компьютеров в различные сферы человеческой деятельности [1].

Автоматизация – процесс передачи выполняемых человеком функций компьютеру или компьютерной системе [2].

Конфигурирование системы – применение изменений к системе с целью приведения ее в требуемое состояние.

Микросервисная архитектура – вариант сервис-ориентированной архитектуры программного обеспечения, направленной на взаимодействие, насколько это возможно, небольших, слабо связанных и легко изменяемых модулей – микросервисов [3].

Асинхронность в программировании – выполнение процесса в неблокирующем режиме системного вызова, что позволяет потоку программы продолжать обработку.

Рекомендательные системы – программы, которые пытаются предсказать, какие объекты (фильмы, музыка, книги, новости, веб-сайты) будут интересны пользователю, собирают определенную информацию о его профиле.

Две основные стратегии создания рекомендательных систем – фильтрация на основе содержания и коллаборативная фильтрация [4].

2. КОЛЛАБОРАТИВНАЯ ФИЛЬТРАЦИЯ

Коллаборативная фильтрация – это метод, позволяющий предсказать неизвестные предпочтения пользователя на основе известных оценок и/или поведения других пользователей. Работа всех видов данного метода основывается на утверждении о том, что пользователи, одинаково оценившие предметы системы, имеют склонность одинаково оценить и другие предметы системы [4].

В классическом случае строится матрица пользователей-предметов, значения в которой являются оценками конкретного пользователя конкретного предмета. Те ячейки, в которых нет значений, являются неизвестными. Это значит, что для данного предмета пользователь не выставил оценку (см. таблицу).

Пример матрицы пользователей-предметов

Пользователь	«Топ»	«Топ 2»	«Топ 3»	«Мстители»
Иван	5	4	–	3
Сергей	–	–	4	3
Дмитрий	3	–	–	5

Для определения соседства пользователей применяются несколько различных алгоритмов.

- Манхэттенское расстояние. Данный метод имеет недостаток в точности при малом заполнении матрицы, но также имеет простую реализацию и высокую скорость выполнения.
- Евклидово расстояние. Данный метод имеет проблемы при незаполненной матрице, но прост в разработке и дешев в выполнении.
- Коэффициент корреляции Пирсона. Данный метод имеет большую точность относительно двух других, но более трудоемок.

Основным недостатком коллаборативной фильтрации является проблема холодного старта, суть которой заключается в том, что система выдает неточные рекомендации при малом количестве пользователей в системе либо при малом количестве оценок пользователей [5]. Для решения этой проблемы в рамках данной работы были проведены исследования, описанные в следующем разделе.

3. ИССЛЕДОВАНИЕ ПОПУЛЯРНОСТИ ЯЗЫКОВ ПРОГРАММИРОВАНИЯ И СТЕКОВ ТЕХНОЛОГИЙ

В качестве источников данных для исследования были выбраны следующие ресурсы, поскольку они дают широкий охват аудитории IT-сообщества и предоставляют открытую статистику:

1) hh.ru – HeadHunter – один из крупнейших в России сайтов для размещения вакансий и резюме. Статистика HeadHunter позволяет оценить реальную востребованность технологии на рынке труда;

2) github.com – GitHub – крупнейший веб-сервис для хостинга IT-проектов и их совместной разработки. Статистика GitHub интересна тем, что она рассчитывается на основании языков, используемых в реальных проектах, выложенных на этом хостинге;

3) stackoverflow.com – Stack Overflow – популярная система вопросов и ответов о программировании. Статистика Stack Overflow интересна тем, что она основывается на количестве вопросов и ответов, заданных по какому-либо языку программирования или технологии, и на голосовании участников сообщества.

Перед исследованием популярности языков программирования среди работодателей и соискателей был выделен общий список языков программирования, откуда были исключены различные «вспомогательные технологии» (например, TypeScript, CSS и т. д.), что позволило отразить реальный результат, поскольку эти технологии используются исключительно в связке с каким-либо языком программирования.

Статистика популярности среди работодателей составлялась двумя способами:

- когда название языка встречается в любом месте вакансии (описание, название, требования);
- когда название языка встречается только в названии вакансии.

Такой подход необходим, чтобы отделить основное требование от сопутствующих навыков и иметь более полное представление о ситуации на рынке.

Ниже представлены результаты исследования популярности технологий среди работодателей. Исследование вакансий проводилось за первое полугодие 2018 г.

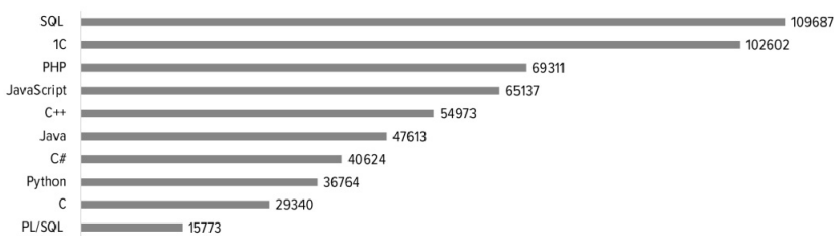


Рис. 1. 10 самых популярных языков по упоминанию в любом пункте вакансии.
График отображает дополнительные навыки программистов

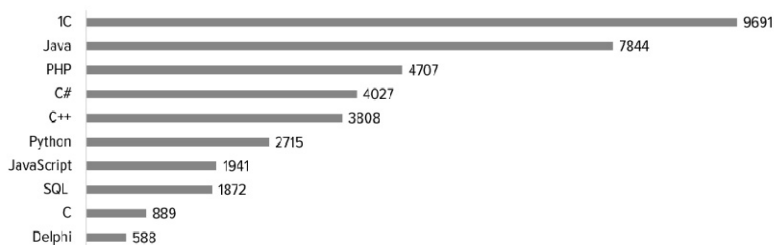


Рис. 2. 10 самых популярных языков по упоминанию в названии вакансии.
График отображает основные навыки программистов

Ниже представлены результаты исследования среди программистов, основанные на статистике, представленной сайтом Stack Overflow. Данную статистику формируют пользователи ресурса посредством опроса и других средств сбора информации.

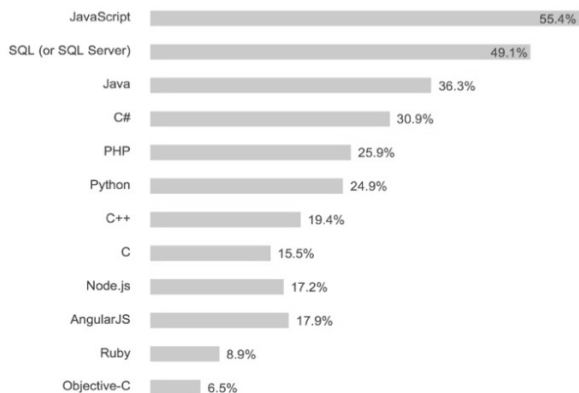


Рис. 3. Рейтинг популярности языков программирования
среди разработчиков за 2018 г.

Можно сделать вывод, что самыми востребованными для работодателей языками программирования являются 1С и Java, это объясняется большим количеством продуктов, использующих эти языки. Самым востребованным дополнительным навыком ожидаемо является язык программирования баз данных SQL, что объясняется наличием баз данных в большинстве проектов. Сами же разработчики предпочитают язык JavaScript, что объясняется его широким применением в веб-разработке, и SQL по описанной выше причине.

4. ОПИСАНИЕ РАЗРАБАТЫВАЕМОЙ СИСТЕМЫ

Разрабатываемая система получила название «EasyToSetup», или «Легко настроить», и описывает основное назначение и преимущество разрабатываемой системы: простая и удобная конфигурация удаленных машин.

Для разработки системы были выбраны следующие инструменты: Python 3.6.4, Django 1.9, PostgreSQL 9.6, MongoDB 3.4, Docker 18.03, Ansible 2.6.

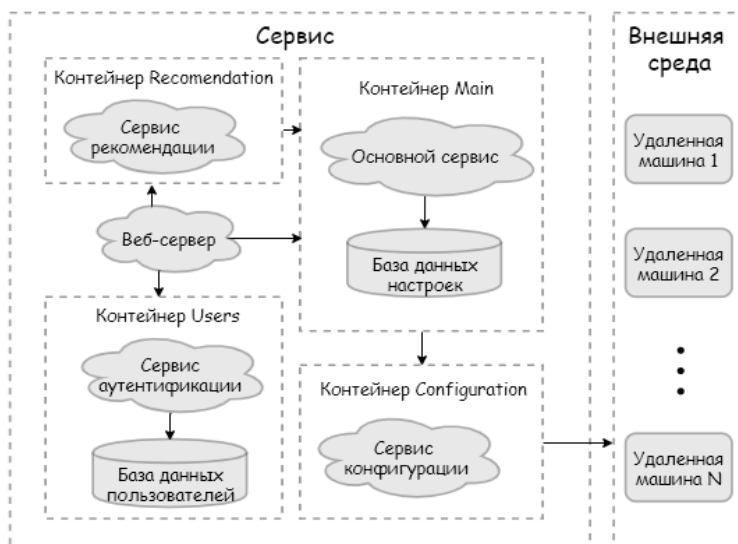


Рис. 4. Диаграмма компонентов разрабатываемой системы

EasyToSetup является распределенной системой, которая состоит из четырех микросервисов и точки входа. Каждый из этих микросервисов представляет собой небольшое, независимое от других микросервисов приложение, решающее собственный набор бизнес-задач.

Данная архитектура позволяет развернуть различные модули на разных серверах и легко масштабировать решение с помощью запуска дополнительных копий микросервисов, на которые оказывается высокая нагрузка, тем самым балансируя ее.

Каждый из указанных на рис. 4 контейнеров-микросервисов и является docker-контейнером, который выполняет собственный набор бизнес-задач.

- Веб-сервер – точка входа в приложения, которая связывает между собой отдельные микросервисы (рис. 5);

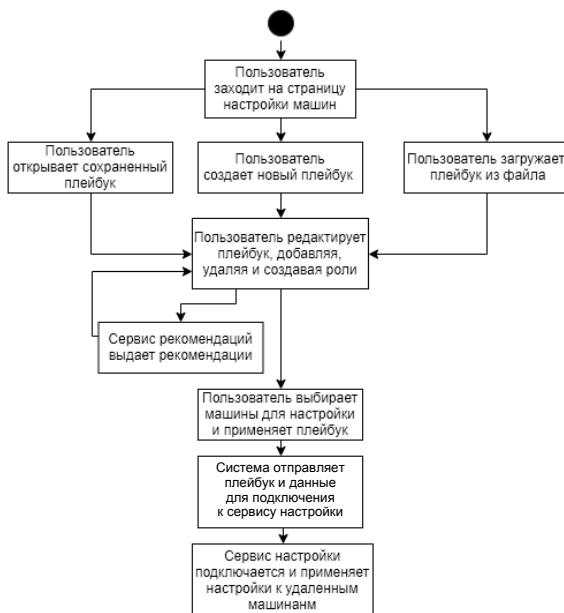


Рис. 5. Алгоритм применения плейбука к удаленным машинам

- Контейнер Main – основной микросервис, в котором реализована внешняя составляющая системы и реализована работа с настройками, которые позже будут применяться к удаленным компьютерам.

- Контейнер Users – микросервис для работы с пользователями, через него выполняется регистрация и авторизация пользователей, там же хранятся данные о пользователях и их права.

- Контейнер Configuration – микросервис по конфигурированию удаленных рабочих мест. Представляет собой асинхронный веб-сервер, который

принимает конфигурацию (плейбук) и данные для подключения к компьютеру, выдает тикет о принятии задачи, после чего выполняет конфигурирование рабочего места с помощью Ansible и возвращает результат.

- Контейнер Recommendation – микросервис, выдающий рекомендации по составлению плейбука, который будет применен к рабочему месту (рис. 6).

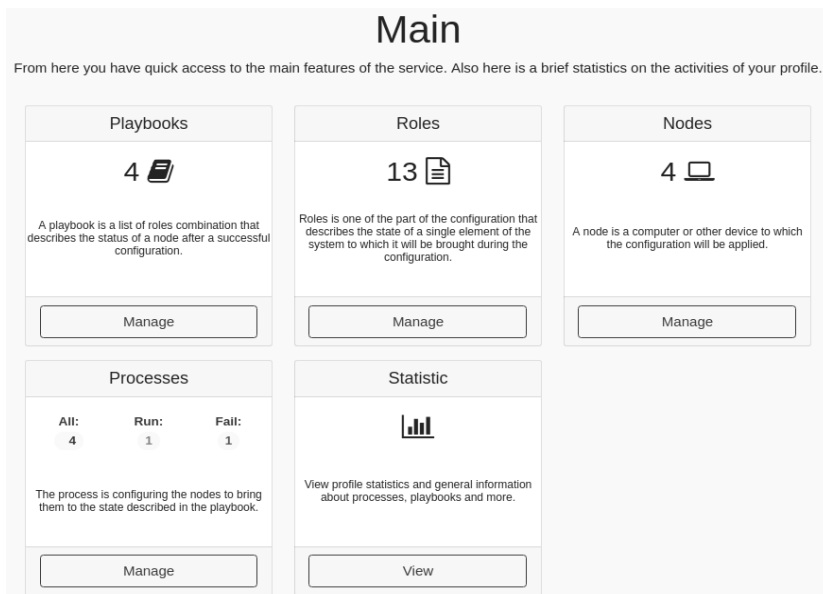


Рис. 6. Главная страница EasyToSetup

Вся система имеет интерфейс на английском языке, чтобы ее могли использовать пользователи по всему миру. С главной страницы можно перейти к управлению следующими элементами:

- Role – это элемент конфигурации, описывающий один элемент системы, который будет сконфигурирован при настройке системы. Например, роль может описывать, что в системе будет установлена СУБД PostgreSQL версии 9.6;

- Playbook – это своего рода слепок системы, состоящий из ролей и описывающий установленные пакеты и состояние, которое должна иметь система после применения к ней плейбука;

- Node – это удаленная машина, к которой будет применяться конфигурация; для подключения необходимо, чтобы машина принимала соединение по протоколу ssh;

- Processes – страница с активными и выполненными процессами по конфигурированию удаленных машин;
- Statistics – это статистика, где можно просмотреть самые популярные плейбуки и роли, которые применялись всеми пользователями.

При создании или редактировании плейбука (рис. 7) пользователю будет выдана рекомендация, которая основывается на том, что если пользователь добавил себе, например, роль Docker, которая часто используется в связке с ролью Docker-compose, то ему может быть интересна роль Docker-compose (рис. 8).



Рис. 7. Страница управления плейбуками



Рис. 8. Окно редактирования плейбука с примером рекомендации роли на основании добавленных рекомендаций

ЗАКЛЮЧЕНИЕ

В ходе данной работы была разработана система рекомендации и автоматизации развертывания программного обеспечения на удаленных терминалах. Система полностью решает поставленные на нее задачи, основными ее преимуществами являются низкий порог вхождения и простота использования.

Было проведено исследование популярности языков программирования и стеков технологий, используемых для разработки ПО, что позволило решить проблему холодного старта в разработанной рекомендательной системе.

В будущем планируется расширить функционал системы с предоставлением рекомендаций по составлению конфигураций. Также планируется реализовать применение конфигураций к операционным системам семейства Windows.

СПИСОК ЛИТЕРАТУРЫ

1. Компьютеризация [Электронный ресурс] // Большой энциклопедический словарь. – URL: <https://dic.academic.ru/dic.nsf/enc3p/161134> (дата обращения: 28.05.2019).
2. *Хлебениких Л.В., Зубкова М.А., Саукова Т.Ю.* Автоматизация производства в современном мире [Электронный ресурс] // Молодой ученый. – 2017. – № 16. – С. 308–311. – URL <https://moluch.ru/archive/150/42390/> (дата обращения: 28.05.2019).
3. *Balalaie A., Heydarnoori A., Jamshidi P.* Microservices architecture enables DevOps: migration to a cloud-native architecture // IEEE Software. – 2016. – Vol. 33 (3). – P. 42–52. – DOI: 10.1109/MS.2016.64.
4. *Koren Y., Bell R., Volinsky C.* Matrix factorization techniques for recommender systems // Computer. – 2009. – Vol. 42 (8). – P. 30–37.
5. *Melville P., Sindhvani V.* Recommender systems // Encyclopedia of Machine Learning / ed. by C. Sammut, J. Webb. – Boston, MA: Springer, 2010. – P. 829–838. – ISBN 978-0-387-30768-8.
6. *Melville P., Mooney R., Nagarajan R.* Content-boosted collaborative filtering for improved recommendations // Eighteenth national conference on artificial intelligence AAAI-02. – Austin, TX, USA, 2002. – P. 187–192.
7. *O'Mahony M.* Collaborative recommendation: a robustness analysis // ACM Transactions on Internet Technology. – 2004. – Vol. 4, N 4. – P. 344–377.
8. *Гурвиц Г.А.* Разработка реального приложения в среде клиент-сервер. – Хабаровск: ДВГУПС, 2005. – С. 32–34.
9. *Кнут Д.* Искусство программирования. Т. 1. Основные алгоритмы. – 3-е изд. – М.: Вильямс, 2006.

10. *Лутц М.* Программирование на Python. Т. 1. – М.: Символ, 2016. – С. 108–213.
11. *Омеляненко Е.А., Басыня Е.А., Сафронов А.В.* Разработка системы контроля и управления доступом персонала в рудник на основе биометрической идентификации // Перспективное развитие науки, техники и технологий: сборник трудов научно-практической конференции, Курск, 13–14 окт. 2017 г. – Курск: Изд-во ЮЗГУ, 2017. – С. 142–149. – ISBN 978-5-9905939-3-0.
12. *Хахаев И.А.* Практикум по алгоритмизации и программированию на Python: учебник. – М.: Альт Линукс, 2010.
13. *Саммерфилд М.* Программирование на Python 3: подробное руководство. – М.: Символ, 2016.
14. *Кристин Л., Грегори Д.* Гибкое тестирование: практическое руководство для тестировщиков ПО и гибких команд. – М.: Вильямс, 2010. – 464 с.
15. *Бейзер Б.* Тестирование черного ящика: технологии функционального тестирования программного обеспечения и систем. – СПб.: Питер, 2004.

Омеляненко Егор Анатольевич, магистрант кафедры вычислительной техники Новосибирского государственного технического университета.
E-mail: egaom@mail.ru

DOI: 10.17212/2307-6879-2019-1-42-54

Developing a system for recommending and automating software deployment to remote terminals*

E.A. Omelyanenko

Novosibirsk State Technical University, 20 K. Marx Prospekt, Novosibirsk, 630073, Russian Federation, doctor of Technical Sciences, undergraduate of the department of computer engineering. E-mail: egaom@mail.ru

The past few decades saw the rapid development of computer technology and information technology. Computers are so firmly established in our lives that it is already impossible to imagine work in many industries, for example, in architecture, banking, medicine, without using them.

With the new industries computerization and areas of life, the number of computers in the workplace and the number of software (software) specific to a particular profession will grow.

* Received 15 November 2018.

Also, in most cases, not a single working tool is used, but some combination of them, which may differ for specialists in one area, and even for specialists working on one project. For example, in a website developing, the server-side developer will use one set of software, and the client-side developer another.

It appears from the facts, as described above, that there is the problem of configuring and installing software products in various combinations for a large number of computers (workstations). In large companies, specialized departments of engineers are responsible for this, but, it is often unprofitable or not necessary among small and medium businesses to maintain an entire department, or even a single specialist, for these purposes.

The described problems could solve existing configuration management systems, but they are too difficult to learn and use, because they were designed to manage servers and cannot be used by people without substantive knowledge in the field of information technology. The developed solution solves the problems described.

Keywords: computerization, automation, configuration, recommender system, asynchrony, microservice architecture, containerization, Ansible

REFERENCES

1. Komp'yuterizatsiya [Computerize]. *Bol'shoi entsiklopedicheskii slovar'* [Big encyclopaedic dictionary]. Available at: <https://dic.academic.ru/dic.nsf/enc3p/161134> (accessed 28.05.2019).
2. Khlebenskikh L.V., Zubkova M.A., Saukova T.Yu. Avtomatizatsiya proizvodstva v so-vremennom mire [Automation of production in the modern world]. *Molodoi uchenyi – Young Scientist*, 2017, no. 16, pp. 308–311. Available at: <https://moluch.ru/archive/150/42390/> (accessed 28.05.2019).
3. Balalaie A., Heydarnoori A., Jamshidi P. Microservices architecture enables DevOps: migration to a cloud-native architecture. *IEEE Software*, 2016, vol. 33 (3), pp. 42–52. DOI: 10.1109/MS.2016.64.
4. Koren Y., Bell R., Volinsky C. Matrix factorization techniques for recommender systems. *Computer*, 2009, vol. 42 (8), pp. 30–37.
5. Melville P., Sindhvani V. Recommender systems. *Encyclopedia of Machine Learning*. Ed. by C. Sammut, J. Webb. Boston, MA, Springer, 2010, pp. 829–838. ISBN 978-0-387-30768-8.
6. Melville P., Mooney R., Nagarajan R. Content-boosted collaborative filtering for improved recommendations. *Eighteenth national conference on artificial intelligence AAAI-02*, Austin, TX, USA, 2002, pp. 187–192.
7. O'Mahony M. Collaborative recommendation: a robustness analysis. *ACM Transactions on Internet Technology*, 2004, vol. 4, no. 4, pp. 344–377.
8. Gurvits G.A. *Razrabotka real'nogo prilozheniya v srede klient-server* [Developing a real-world client-server application]. Khabarovsk, DVGUPS Publ., 2005, pp. 32–34.
9. Knuth D.E. *The art of computer programming*. Vol. 1. *Fundamental algorithms*. Reading, MA, Addison-Wesley, 1997 (Russ. ed.: Knut D.E. *Iskusstvo programmirovaniya*. T. 1. *Osnovnye algoritmy*. 3rd ed. Moscow, Vil'yams Publ., 2006.

10. Lutz M. *Programmirovaniye na Python*. T. 1 [Programming in Python]. Moscow, Simvol Publ., 2016, pp. 108–213. (In Russian).
11. Omel'yanenko E.A., Basynya E.A., Safronov A.V. [Development of a system of control and management of personnel access to the mine based on biometric identification]. *Perspektivnoye razvitie nauki, tekhniki i tekhnologii: sbornik trudov nauchno-prakticheskoi konferentsii* [Perspective development of science, engineering and technology: transaction of papers of the scientific-practical conference], Kursk, 13–14 October 2017, pp. 142–149. ISBN 978-5-9905939-3-0. (In Russian).
12. Khakhaev I.A. *Praktikum po algoritmizatsii i programmirovaniyu na Python* [Workshop on algorithms and programming in Python]. Moscow, Al't Linuks Publ., 2010.
13. Summerfield M. *Programmirovaniye na Python 3: Podrobnoe rukovodstvo* [Programming in Python 3: a complete introduction to the Python language]. Moscow, Simvol Publ., 2016.
14. Crispin L., Gregory J. *Gibkoe testirovaniye: prakticheskoe rukovodstvo dlya testirovshchikov PO i gibkikh komand* [Agile testing: a practical guide for testers and agile teams]. Moscow, Vil'yams Publ., 2010. 464 p.
15. Beizer B. *Testirovaniye chernogo yashchika: tekhnologii funktsional'nogo testirovaniya programmogo obespecheniya i sistem* [Black-box testing: techniques for functional testing of software and systems]. St. Petersburg, Piter Publ., 2004.

Для цитирования:

Омеляненко Е.А. Разработка системы рекомендации и автоматизации развертывания программного обеспечения на удаленных терминалах // Сборник научных трудов НГТУ. – 2019. – № 1 (94). – С. 42–54. – DOI: 10.17212/2307-6879-2019-1-42-54.

For citation:

Omel'yanenko E.A. Razrabotka sistemy rekomendatsii i avtomatizatsii razvertyvaniya programmogo obespecheniya na udalennykh terminalakh [Developing a system for recommending and automating software deployment to remote terminals]. *Sbornik nauchnykh trudov Novosibirskogo gosudarstvennogo tekhnicheskogo universiteta* – Transaction of scientific papers of the Novosibirsk state technical university, 2019, no. 1 (94), pp. 42–54. DOI: 10.17212/2307-6879-2019-1-42-54.