

ТЕХНОЛОГИЯ И ТЕХНИКА ПОЛИГРАФИЧЕСКОГО И УПАКОВОЧНОГО ПРОИЗВОДСТВА

УДК 671:083.73:739.2

А. А. Молдованов, И. А. Астафьев, Л. С. Корочкин, М. С. Шмаков
Белорусский государственный технологический университет

РАСШИРЯЕМЫЕ ИНФОРМАЦИОННЫЕ СИСТЕМЫ КОНТРОЛЯ ЛОГИСТИЧЕСКИХ ЕДИНИЦ

В статье рассматриваются программно-системные решения учета стадий жизненного цикла логистических единиц (производимой, импортируемой и экспортируемой продукции). Представлены результаты исследования возможных способов и технологий построения расширяемых информационных систем хранения и обработки больших объемов данных

Программно-системные решения учета стадий жизненного цикла продукции должны предполагать под собой универсальность в отношении типа изделий, к которым будет применена система контроля. Последняя должна быть адаптирована к постоянному номенклатурному расширению единиц продукции, способна обрабатывать многотысячные потоки данных в единицу времени, предполагать резервирование исходных и обработанных данных. Качество и стабильность функционирования должны быть закреплены за юридическим гарантом, так называемым Центром обработки данных (ЦОД), который, в свою очередь, должен быть лицензирован государством. Повышение общей стабильности работы с технической стороны должно реализовываться современными аппаратными решениями, качество исходных кодов и возможность их сопровождения реализуется через шаблоны проектирования и покрытие автоматизированными тестами.

Предложены технологии построения серверной и клиентской частей в рамках системы. Рекомендованы подходы в построении программных архитектур. Использование предлагаемых программно-системных решений позволит обеспечить расширяемость, сопровождаемость и стабильность работы рассматриваемых информационных системы учета.

Ключевые слова: программно-системные решения, учет, жизненный цикл, продукция, защита, программная структура, виртуальные кабинеты, сервер, клиент.

A. A. Moldovanov, I. A. Astaf'yev, L. S. Korochkin, M. S. Shmakov
Belarusian State Technological University

EXTENDABLE INFORMATION SYSTEMS FOR LOGISTIC UNIT CONTROL

The article discusses the system-software solutions for accounting the stages of the life cycle of logistics units (manufactured, imported and exported products). The results of a study of possible methods and technologies for building expandable information systems for storing and processing large amounts of data are presented

Software and system solutions for accounting for the stages of the product life cycle should assume universality in relation to the type of products to which the control system will be applied. The latter should be adapted to the constant nomenclature expansion of product units. Be able to process thousands of data streams per unit of time, assume the backup of the source and processed data. The quality and stability of functioning should be assigned to a legal guarantor, the so-called Data Processing Center (DPC), which in turn must be licensed by the state. Improving the overall stability of work from the technical side should be implemented with modern hardware solutions, the quality of source codes and the possibility of their maintenance is implemented through design patterns and coverage with automated tests.

Technologies for building server and client parts within the system are proposed. Recommended approaches to building software architectures. Using the proposed software and system solutions will ensure the extensibility, maintainability and stability of the considered information accounting systems.

Key words: software and system solutions, accounting, life cycle, products, protection, program structure, virtual cabinets, server, client.

Введение. На данный момент учет стадий жизненного цикла продукции необходим в силу специфики развития современных производственно-потребительских отношений в рамках государственного контроля качества выпускаемой продукции наряду с совершенствованием системы налогообложения, необходимостью контроля уровня теневых рыночных отношений.

Цель концептуальной идеи – повысить уровень доверия потребителя к производителю, обезопасить обоих, снизить уровень фальсификации изделий. Остается актуальным вопрос необходимости виртуализации рабочих пространств в рамках описываемых хозяйственных отношений. Современная практика предлагает к использованию расширяемые программно-системные комплексы для решения поднятых вопросов.

Основная часть. Основной причиной создания и развития автоматизированных информационных систем (АИС) является необходимость ведения учета информации о состоянии и динамике объекта, которому посвящена система. На основании информационной картины, создаваемой системой, руководители различного звена могут принимать решения об управляющих воздействиях с целью решения текущих проблем.

По результатам анализа предлагается к использованию трехзвенная архитектура АИС с возможностью расширения, характерными особенностями которой являются высокая степень гибкости и масштабируемости, высокая безопасность (так как защиту можно определить для каждого сервиса или уровня), высокая производительность (так как задачи распределены между серверами).

Общая архитектура трехзвенной клиент-серверной системы представлена на рис. 1.

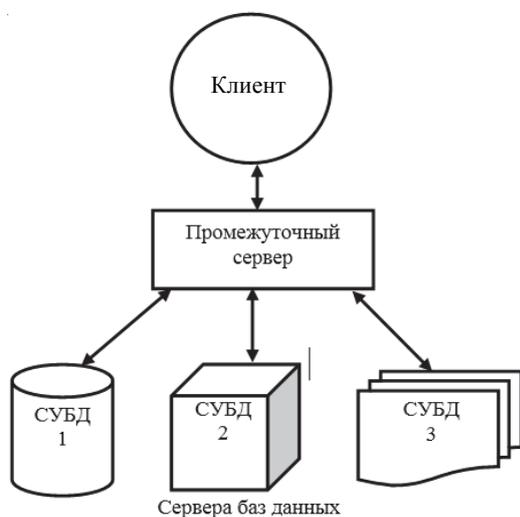


Рис. 1. Трехзвенная архитектура «клиент – сервер»

Основная идея архитектуры «клиент – сервер» состоит в разделении сетевого элемента архитектуры на несколько компонентов, каждый из которых реализует специфический набор сервисов. Подобные компоненты могут выполняться на разных компьютерах, обрабатывая серверные и / или клиентские функции. Это позволяет повысить надежность, безопасность и производительность сетевых приложений и сети в целом. В разрезе международных практик построения распределенных архитектур подобное решение закреплено за стандартами SPA/WebApp и PWA & Mobile. Под первым подразумеваются полнофункциональные браузерные приложения без перезагрузки страниц (клиент-серверное взаимодействие согласно AJAX-подходу), под вторым – Progressive Web Application и мобильные приложения (протокол HTTPS).

Произведен анализ современных подходов в построении программных структур информационных систем [1]. Задача нахождения рационального программно-структурного решения преследует цели установить взаимосвязи между компонентами системы, определить ее общие принципы и правила функционирования, концепцию разработки программных сред так, чтобы те оставались тестируемыми и сопровождаемыми. Также необходимо обеспечить расширяемость и стабильность работы системы. Само понятие структурности в информационных системах подразумевает под собой наличие взаимосвязанных аппаратных и программных компонентов.

Среди программных компонентов выделяют:

- операционные системы (ОС);
- операционные оболочки (ОО);
- программные сервисы;
- (интегрированные) среды разработки;
- пакеты прикладных программ широкого и частного назначения;
- системы управления базами данных;
- системы контроля версий, непрерывного интегрирования и разворачивания.

Среди аппаратных компонентов выделяют:

- сервера баз данных;
- средства резервного хранения и распределенной обработки данных;
- клиентские вычислительные устройства;
- наличие средств коммуникации по взаимнообратному направлению «клиент – сервер».

В качестве примера рассмотрим детализированную структуру серверной части информационной системы с применением современных технологий контейнеризации Docker, оркестровки контейнеров Kubernetes и распределенной базой данных. В представленном случае ETCD является распределенной базой данных (рис. 2).

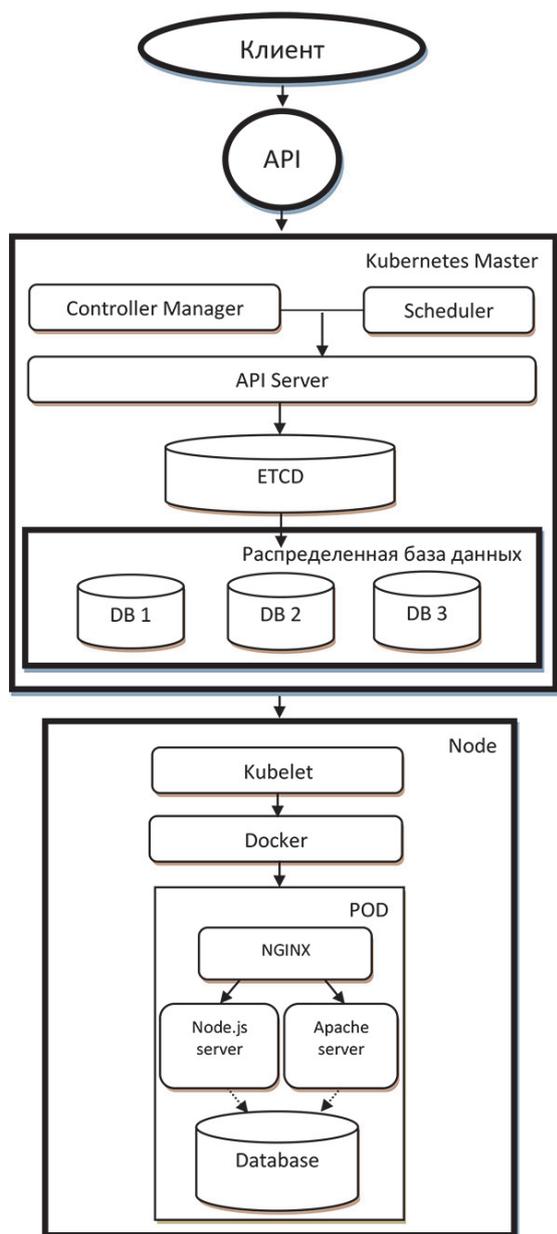


Рис. 2. Структура серверной части отказоустойчивой АИС

API server отвечает за политики, управляет потоками данных. Scheduler является планировщиком задач. Controller управляет состояниями кластеров. Node представляет собой рабочую машину (компьютер, сервер), на которой установлен Kubelet, управляющий POD (базовой единицей, запускающей контейнера Docker). Подобный подход позволяет добиться высокой отказоустойчивости системы и обеспечить автоматическую балансировку нагрузки на распределенные аппаратные средства.

Согласно анализу серверных операционных систем, на данный момент наиболее приоритетной по таким показателям, как скорость работы – стабильность – отсутствие проприетарной ответственности перед авторами, является

ОС Linux. На рис. 3 представлена диаграмма процентного соотношения веб-серверов (согласно статистике сервиса W3Schools на 2019 г.), использующих различные операционные системы [2].

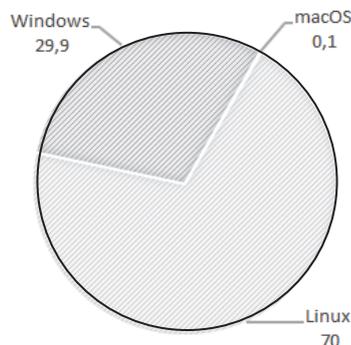


Рис. 3. Процентное соотношение использования веб-серверами ОС

Исследование рынка клиентских операционных систем для desktopных вычислительных машин (персональных электронно-вычислительных машин (ПЭВМ)) позволило сделать вывод об их процентном соотношении. Наиболее востребованной является ОС Windows 10 версии. Результаты (согласно статистике аналитического сервиса NET MARKETSHARE на 2019 г.) представлены на рис. 4 [3]. В данную статистику не включены операционные системы мобильных устройств, так как последние относятся к отдельному рыночному сегменту.

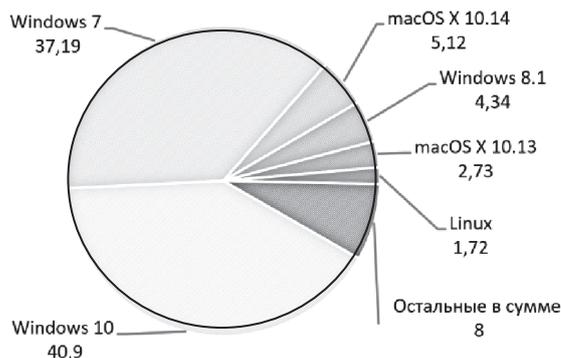


Рис. 4. Процентное соотношение ОС на клиентских desktopных машинах

Согласно концепции рассматриваемой распределенной информационной системы, клиентская сторона может реализовываться не только на платформе операционных систем, предназначенных для работы на ПЭВМ, но и на мобильных платформах. Мобильная операционная система (мобильная ОС) – операционная система для смартфонов, планшетов, карманных переносных компьютеров (КПК) или дру-

гих мобильных устройств. Мобильные операционные системы сочетают в себе функциональность ОС для персонального компьютера (ПК) с функциями для мобильных и карманных устройств: сенсорный экран, сотовая связь, Bluetooth, Wi-Fi, GPS-навигация, камера, видеокамера, распознавание речи, диктофон, музыкальный плеер, NFC и инфракрасное дистанционное управление.

Пользователь может получить доступную относительно его роли информацию из системы по средствам мобильных приложений, будь он контролирующим органом либо покупателем. Согласно результатам анализа рынка мобильных операционных систем, наибольшее развитие и востребованность среди пользователей в последние годы показывает платформа Android. Процентное соотношение мобильных операционных систем (по данным сервиса NET MARKETSHARE) отражено на рис. 5 [2].

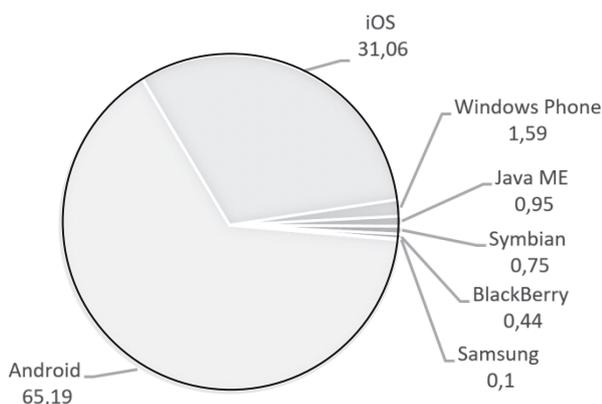


Рис. 5. Процентное соотношение использования мобильных ОС пользователями

Можно сделать вывод, что целевыми платформами для клиентской стороны являются платформы на операционных системах Android, iOS и Windows. Также возможна реализация клиентских рабочих пространств в виде веб-приложения. В данном случае браузер пользователя является исполняющей рабочей средой для клиентского приложения. Таким образом, отпадает необходимость целенаправленной разработки под конкретную платформу, однако подобные приложения имеют более низкий показатель производительности в отличие от нативных программных сред. На рис. 6 представлена статистика применения пользователями различных браузеров [3].

Использование браузера в качестве исполняющей среды для работы клиентского приложения является рациональным решением по следующим причинам: отсутствие привязки к конкретной аппаратно-программной платформе

клиента, отсутствие необходимости в установке дополнительного программного обеспечения (кроме самого браузера), доступ в режиме реального времени к последней актуальной рабочей версии клиентского приложения. Однако при отсутствии специального сканера штрих-кодов пользователь не сможет считать информацию с контрольного знака. Приложения же на базе мобильных платформ имеют доступ к камере гаджета, которая используется в качестве сканера

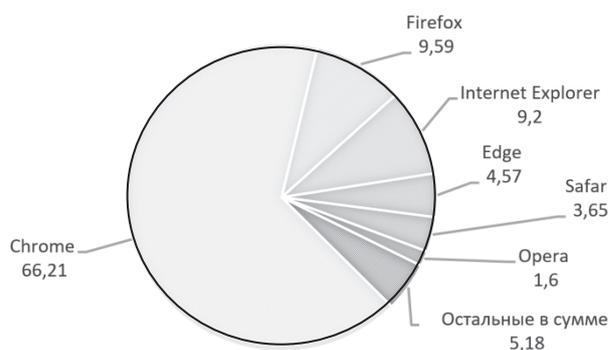


Рис. 6. Процентное соотношение использования браузеров пользователями

В качестве ОС на серверной стороне предлагается Linux; в качестве программной платформы клиентской стороны стек браузеров – Chrome, Firefox, Internet Explorer; в качестве мобильных платформ исполнения – Android, iOS.

Виртуальные кабинеты участников необходимы для интеграции их в информационную систему контроля и обеспечения проверки правомерности их доступа к данным. По средствам виртуальных кабинетов должно реализовываться централизованное взаимодействие между участниками системы с возможностью учета всех операций и инцидентов. Подобный подход позволяет отслеживать первопричину и источник потенциальных и возникших ошибок в системе. Виртуальные кабинеты участников системы должны разделяться концептуально выделенным ролям.

Виртуальный кабинет пользователя представляет собой одностраничное приложение (Single Page Application (SPA)). Для виртуализации рабочего пространства используются следующие программные компоненты [4]:

- Модель – Представление – Модель Представления (MVVM), Модель – Представление – Контроллер (MVC) фреймворки. Согласно результатам анализа подобного рода фреймворков, лидирующие позиции с высоким уровнем поддержки разработчиков занимают React, Vue, Angular;

- Роутинг (Router) – компонент, который отвечает за навигацию пользователя между представлениями;

– Шаблонизатор – программное обеспечение, позволяющее использовать html-шаблоны для генерации конечных html-страниц. Его цель – это отделение представления данных от исполняемого кода;

REST – ориентированный бэкенд со стороны сервера. Под этим компонентом подразумевается прикладной интерфейс приложения (API) для клиент-серверного взаимодействия.

На рис. 7 представлена структура реализации виртуальных кабинетов.

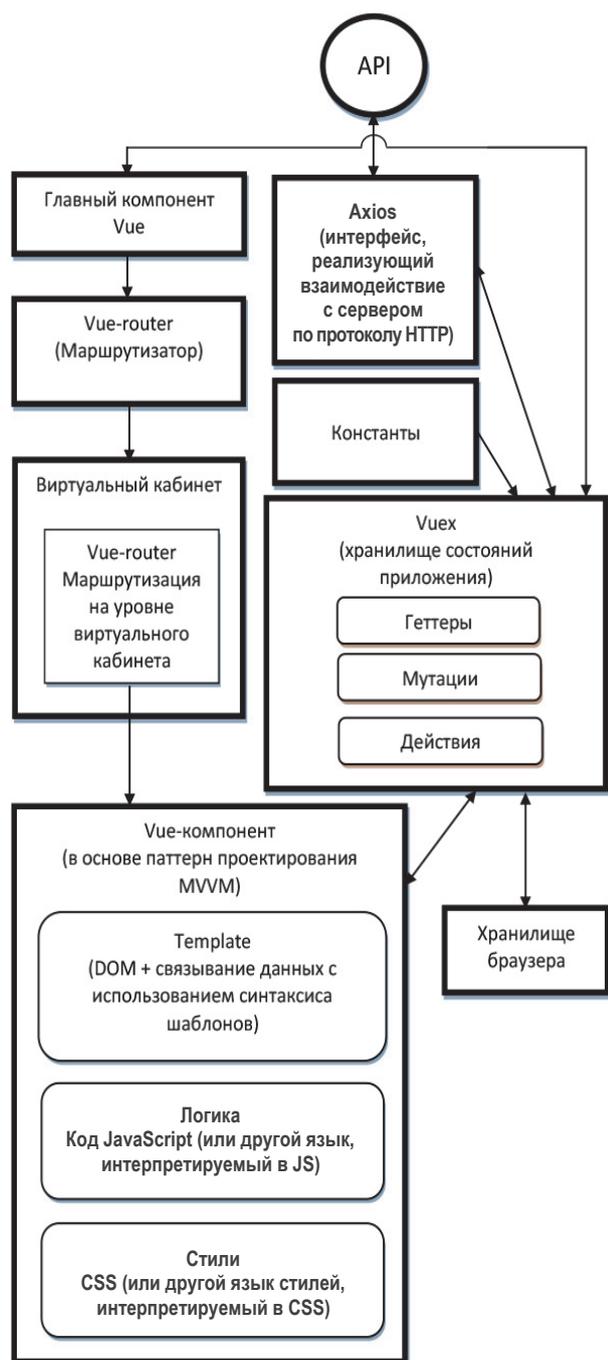


Рис. 7. Структура реализации виртуальных кабинетов

Предлагаемая структура реализации виртуальных кабинетов на базе фреймворка Vue, а также плагинов Vuex (хранилище состояний приложения) и Vue-router (маршрутизатор, роутинг).

Структура исходного кода рекомендована к реализации согласно концепции «чистой архитектуры (clean architecture)» [4]. Для стороны клиента на базе веб-технологий предлагается структура Vue, основанная на функциональном подходе двустороннего связывания данных (two-way data binding). Подобный SPA виртуальный кабинет должен обеспечивать реализацию всех функций, предоставленных пользователю согласно концепции системы, по средствам взаимодействия с пользовательским интерфейсом.

Для обеспечения расширяемости, сопровождаемости и стабильности системы применяется ряд средств [4]:

- шаблоны проектирования: MVP, MVC, MVVM;
- автоматизированное тестирование (Unit-тесты, интеграционные и инструментальные тесты);
- системы контроля версий;
- системы непрерывной интеграции и развертывания (CI/CD systems).

Заключение. В результате проведенной работы можно заключить, что в качестве программно-системных решений учета стадий жизненного цикла продукции предлагается к использованию распределенная масштабируемая информационная система с типом взаимодействия «тонкий клиент – сервер», в которой каждый товар идентифицируется уникальным ключом в виде символьной последовательности, хранимой в банке данных.

Каждый этап ЖЦ товара и его текущий статус фиксируется в базе данных ЦОД через средство интеграции приложений API вплоть до выбытия отслеживаемой позиции из оборота. Доступ к информации разделяется по ролям пользователей согласно общей концепции системы.

Виртуализация рабочих пространств участников системы является рекомендуемым и обязательным для централизации обработки данных ЦОД и предоставления участникам последних актуальных версий системы. Взаимодействие между виртуальным кабинетом и сервером должно обеспечиваться с поддержкой протоколов шифрования данных.

Использование предлагаемых программно-системных решений позволит обеспечить расширяемость, сопровождаемость и стабильность работы рассматриваемой информационной системы учета.

Литература

1. Таненбаум Э., ван Стеен М. Распределенные системы. Принципы и парадигмы. СПб.: Питер, 2003. 877 с.
2. Статистика ОС платформ для веб-технологий. URL: <https://www.w3schools.com/> (дата обращения: 01.07.2019).
3. Статистика рынка интернет-технологий. URL: <https://netmarketshare.com/> (дата обращения: 02.07.2019).
4. Мартин Р., Гренниг Д., Браун С. Чистая архитектура. Искусство разработки программного обеспечения. СПб.: Питер, 2018. 352 с.

References

1. Tanenbaum E., van Steen M. *Raspredelennyye sistemy. Printsipy i paradigmy* [Distributed systems. Principles and paradigms]. St. Petersburg, Piter Publ., 2003. 877 p.
2. *Statistika OS platform dlya web-tekhnologiy* [OS Platform Statistics]. Available at: <https://www.w3schools.com/> (accessed 01.07.2019).
3. *Statistika rynka internet-tekhnologiy* [Market Share Statistics for Internet Technologies]. Available at: <https://netmarketshare.com/> (accessed 02.07.2019).
4. Martin R., Grenning J., Brown S. *Chistaya arkhitektura. Iskusstvo razrabotki programmnogo obespecheniya* [Clean Architecture. Principles and Paradigms]. St. Petersburg, Piter Publ., 2018. 352 p.

Информация об авторах

Молдованов Артем Андреевич – аспирант кафедры полиграфического оборудования и систем обработки информации. Белорусский государственный технологический университет (220006, г. Минск, ул. Свердлова, 13а, Республика Беларусь). E-mail: amaldavanau@gmail.com

Астафьев Игорь Александрович – аспирант кафедры полиграфического оборудования и систем обработки информации. Белорусский государственный технологический университет (220006, г. Минск, ул. Свердлова, 13а, Республика Беларусь). E-mail: sektor-sektor@mail.ru

Корочкин Леон Сергеевич – доктор технических наук, доцент, профессор кафедры полиграфического оборудования и систем обработки информации. Белорусский государственный технологический университет (220006, г. Минск, ул. Свердлова, 13а, Республика Беларусь). E-mail: lenyakorochkin@gmail.ru

Шмаков Михаил Сергеевич – кандидат технических наук, доцент, заведующий кафедрой полиграфического оборудования и систем обработки информации. Белорусский государственный технологический университет (220006, г. Минск, ул. Свердлова, 13а, Республика Беларусь). E-mail: contr7@tut.by

Information about the authors

Moldovanov Artseom Andreyevich – PhD student, the Department of Printing Equipment and Information Processing Systems. Belarusian State Technological University (13a, Sverdlova str., 220006, Minsk, Republic of Belarus). E-mail: amaldavanau@gmail.com

Astaf'yev Ihar Aleksandrovich – PhD student, the Department of Printing Equipment and Information Processing Systems. Belarusian State Technological University (13a, Sverdlova str., 220006, Minsk, Republic of Belarus). E-mail: sektor-sektor@mail.ru

Korochkin Leon Sergeevich – DSc (Engineering), Associate Professor, Professor, the Department of Printing Equipment and Information Processing Systems. Belarusian State Technological University (13a, Sverdlova str., 220006, Minsk, Republic of Belarus). E-mail: lenyakorochkin@gmail.ru

Shmakov Mikhail Sergeevich – PhD (Engineering), Associate Professor, Head of the Department of Printing Equipment and Information Processing. Belarusian State Technological University (13a, Sverdlova str., 220006, Minsk, Republic of Belarus). E-mail: contr7@tut.by

Поступила 03.07.2019