

АНАЛИЗ ОПТИМИЗАЦИИ ПРОГРАММНОЙ СИСТЕМЫ НА ПРИМЕРЕ СВОБОДНЫХ АВТОМАТИЗИРОВАННЫХ БИБЛИОТЕЧНО-ИНФОРМАЦИОННЫХ СИСТЕМ

О. И. Васильев¹ [0009-0001-2150-5951], В. Ю. Медведев² [0009-0008-3409-5462]

^{1, 2}*Департамент программной инженерии и искусственного интеллекта
Дальневосточного федерального университета, г. Владивосток*

¹vasilev.o@dvfu.ru, ²medvedev.viu@dvfu.ru

Аннотация

Статья посвящена исследованию возможностей оптимизации работоспособности и повышения эффективности функционирования сложных многофункциональных программных систем на примере свободных автоматизированных библиотечно-информационных систем (далее – АБИС).

К 2023 году в мире накоплен ценный опыт создания и эксплуатации интегрированных АБИС различного масштаба и назначения, однако вопросы совершенствования их проектных решений остаются актуальными. В первую очередь это касается необходимости оптимизации структуры исходного программного кода с целью повышения его читаемости и поддерживаемости, снижения времени выполнения отдельных функциональных модулей, уменьшения объёма занимаемой оперативной памяти.

В рамках исследования был проведён сравнительный анализ исходных кодов нескольких действующих открытых АБИС, реализованных на различных языках программирования. Были изучены основные подходы к проектированию структуры кода, выявлены наиболее частотно используемые алгоритмы и паттерны. Для оценки степени оптимизированности исходного кода был разработан комплекс показателей, включающий оценку структуры, читаемости, модульности и других характеристик. На этой основе проведено сравнение отдельных фрагментов кода до и после применения известных техник рефакторинга.

В результате проведённой работы удалось выявить наиболее распространённые ошибки и недочёты в структуризации исходных кодов АБИС, определить

основные направления их оптимизации. Получены данные о возможном снижении затрат на тестирование и техническую поддержку посредством улучшения качества исходных кодов.

Ключевые слова: *исправление программного кода, оптимизация программной системы, рефакторинг, многоязыковая система, оценка качества программных систем, автоматизированные библиотечно-информационные системы, процесс разработки программного обеспечения.*

ВВЕДЕНИЕ

К настоящему времени создано немало многофункциональных информационных систем, предназначенных для автоматизации различных бизнес-процессов в области библиотечного дела и информационного обеспечения. Однако даже самые сложные из них, такие как Koha, NewGenLib или Evergreen, до сих пор не лишены определённых недостатков в плане структуризации исходных кодов.

Так, несмотря на значительный объём функционала и использование множества модульных компонентов, в большинстве случаев исходный код данных систем характеризуется низкой степенью читаемости и поддерживаемости. Это обусловлено, в частности, отсутствием чёткого разделения кода на слои, наличием переплетений логики приложения и представления, сложным вложением функций и процедур. Далеко не все аспекты оптимизации проектной документации отработаны должным образом. Например, в структуре кода зачастую не учитываются особенности языков реализации, отсутствуют стандарты оформления комментариев и docstring, не соблюдаются рекомендации по именованию переменных и функций.

Всё это негативно сказывается как на скорости выполнения программы, так и на затратах на тестирование, доработку и техническую поддержку на этапах жизненного цикла софтвера. Таким образом, проблема оптимизации структуры исходных кодов АБИС требует детального рассмотрения с привлечением известных подходов к повышению качества программного обеспечения.

Для решения поставленной задачи в данной работе был использован комплексный подход, включающий:

- Сравнительный анализ реализации основных функциональных блоков в нескольких открытых АБИС;
- Выделение наиболее распространённых ошибок проектирования на уровне структуры кода;
- Разработку критериев оценки его оптимизированности;
- Тестирование рефакторинга отдельных фрагментов с последующей оценкой результатов.

Для дальнейшего анализа необходимо остановиться на некоторых фундаментальных аспектах разработки программного обеспечения, оказывающих прямое влияние на качество и эффективность кода.

Во-первых, это так называемая архитектура трехуровневого проектирования (см. например, [1]). Согласно этой концепции, программная система должна состоять из трех взаимосвязанных частей: интерфейсного уровня, бизнес-логики (доменной модели) и хранилища данных. Правильное распределение функциональных блоков между этими слоями обеспечивает максимальную независимость компонентов и разделяемость ответственностей.

Вторым важным моментом является отделение представления данных от логики работы с ними. Это позволяет избежать «жесткой» привязки пользовательского интерфейса ко внутренней структуре данных и алгоритмам их обработки. Такая концепция разработки известна как MVC (Model-View-Controller) (см. например, [2]) и широко используется при создании веб-приложений.

Третьим важнейшим положением теории является Декларативное программирование. В отличие от императивного стиля, оно позволяет задавать логику приложения в виде объявлений отношений между данными и правил их преобразования, без указания непосредственной последовательности выполнения операций. Это дает возможность компилятору самостоятельно отыскать оптимальные алгоритмы вычислений.

Следующий важный блок — это наличие четкой и последовательной системы именования всех элементов программы: переменных, функций, классов, методов и других конструкций языка. Она должна отражать смысл элемента однозначно и кратко. Качественная именровка является фундаментальной основой

для понимания логики работы программы. Еще один важный аспект — это использование объектно-ориентированного подхода (ООП) к проектированию. В соответствии с принципами ООП, сложную систему следует структурировать как набор взаимодействующих объектов с чётко определённым интерфейсом и явно выраженными свойствами. Это позволяет повысить модульность кода и сделать его расширяемым.

Также не менее важным является применение различных шаблонов и паттернов проектирования. Подобные универсальные решения известных задач позволяют структурировать код эффективно и повторноиспользуемо, избегая «изобретения велосипеда». Набор рекомендуемых паттернов для конкретной предметной области следует вводить в стандарты проектирования.

МАТЕРИАЛЫ

Для проведения глубокого сравнительного анализа структуры исходных кодов различных автоматизированных библиотечно-информационных систем в настоящем исследовании был сформирован представительный объем материалов. В частности, в качестве объектов изучения были отобраны исходные тексты ряда широко используемых свободных АБИС: Koha, NewGenLib, Evergreen (см. [3, 4, 5]). При сравнении эти проекты обеспечивают полную функциональность библиотечно-информационных процессов, но реализованы с применением различных подходов и технологий программирования. Так, Koha и NewGenLib написаны преимущественно на Perl, в то время как Evergreen представляет собой масштабную платформу, написанную на C++ с использованием веб-фреймворка Django.

Исходные файлы всех рассматриваемых систем были получены из центральных репозиторий (GitHub, Launchpad и др.) в наиболее свежих стабильных версиях (см. [6–8]).

Для проведения детального технического анализа кода был разработан ряд методик, включая:

– Статистический анализ – подсчет количества классов, функций, переменных, блоков кода для оценки его масштаба.

– Анализ вложенности кода – определение глубины вложенности функций и конструкций для оценки его читабельности.

– Анализ когерентности – изучение соответствия именования элементов программы их предназначению.

– Выделение антипаттернов – выявление ошибочных конструкций с точки зрения теории ООП и рефакторинга.

– Метрический анализ – расчет показателей сложности, сцепленности и других метрик для измерения качества дизайна.

Полученные результаты послужили основой для дальнейшей разработки рекомендаций по оптимизации структуры исходных кодов рассматриваемых систем.

РЕЗУЛЬТАТЫ

Проведённое исследование позволило выявить ряд закономерностей в структуризации исходных кодов рассмотренных автоматизированных систем. Одним из основных недостатков оказалось отсутствие чёткого разделения кода на слои в указанном стандартном виде [9]. Так, во многих модулях смешаны функции представления данных, бизнес-логики и доступа к СУБД, что не разделяет ответственности и затрудняет внесение изменений [10].

Выявлена значительная вариативность подходов к именованию элементов программы. Переменные, классы и методы зачастую недостаточно описательны и не отражают их назначения [11], что затрудняет понимание логики функционирования. Это также касается использования комментариев – в ряде модулей они либо отсутствуют, либо носят формальный характер. Анализ структуры кода показал проблемы с соблюдением принципов объектно-ориентированного программирования. Особенно в значительной части функциональных блоков модулей данных отмечался переизбыток процедурных конструкций вместо объектно-ориентированного дизайна [12, 13]. Также зафиксированы случаи неоптимальной вложенности функций и блоков кода, превышающей 3–4 уровня [14, 15]. Это негативно сказывается на читабельности и поддерживаемости. Кроме того, отмечено использование устаревших алгоритмов и конструкций в отдельных модулях систем, что может повлиять на их производительность [16–18].

Метрический анализ также выявил повышенные значения показателей сложности, сцепленности и длины методов в ряде модулей по сравнению с рекомендуемыми нормами. Это указывает на наличие избыточных взаимосвязей между элементами кода.

По результатам проведённого структурного анализа удалось получить ряд количественных показателей, характеризующих состояние исходного кода рассматриваемых информационных систем.

Данные, полученные в результате исследования, свидетельствуют о значительных проблемах в структурировании исходных кодов рассмотренных АБИС, требующих комплексного подхода к их устранению.

Прежде всего, необходимо последовательное внедрение трехуровневой архитектуры, строго разделяющей представление данных от логики их обработки.

Необходимо тщательно проработать именование всех элементов программы в соответствии со стандартами, чтобы максимально отразить их назначение. Также следует исключить излишне глубокую вложенность фрагментов кода и сократить объем индивидуальных функций. Это повысит его читаемость и управляемость.

Необходимо также максимально приблизиться к объектно-ориентированному стилю программирования за счёт широкого использования паттернов и абстракций. Выполнение перечисленных мероприятий позволит существенно повысить качество исследуемых систем на всех этапах их жизненного цикла.

Для систематизации предложенных оптимизаций целесообразно разработать детальные рекомендации по рефакторингу кода рассматриваемых АБИС.

На уровне архитектуры необходимо предусмотреть:

- Выделение в единый слой функций доступа к СУБД и ORM-маппинга.
- Создание слоя представления данных с использованием DTO-объектов.
- Разделение бизнес-логики на сервисы по функциональным областям.
- Вынесение всей валидации в отдельный валидатор.
- Определение контроллеров, реализующих MVC-паттерн.

На уровне технологического кода рекомендуется:

- Внедрить стандарты именования согласно PSR-2, PEAR.
 - Рефакторить перегруженные классы с целью «единственной ответственности».
-

- Вынести вспомогательную логику в хелперы.
- Использовать фасады и сервис-провайдеры.
- Применять паттерны проектирования (фабричный метод и др.).
- Удалять мёртвый и повторяющийся код.

Выполнение рекомендаций на всех этапах жизненного цикла поможет существенно повысить качество исследуемых АБИС.

ЗАКЛЮЧЕНИЕ

На основании проведенного исследования можно сделать ряд заключений:

1. Выявленные значительные проблемы в структурировании исходных кодов исследуемых АБИС указывают на несоответствие международным стандартам проектирования ПО.

2. Высокие показатели вложенности функций негативно влияют на поддерживаемость кода.

3. Недостаточное разделение ответственностей между слоями усложняет внесение изменений и расширение функционала.

4. Нарушение ООП-принципов делают код менее надежным, повторно используемым и гибким.

Таким образом, выявленные недочеты требуют комплексного применения известных методов оптимизации и рефакторинга на всех этапах жизненного цикла для достижения ожидаемого уровня качества ПО. Это позволит снизить затраты на сопровождение и повысить производительность исследуемых АБИС.

СПИСОК ЛИТЕРАТУРЫ

1. Фаулер М. Архитектура корпоративных программных приложений.: Пер. с англ. М.: Издательский дом «Вильямс», 2006. 544 с.: ил. С. 46–48.
2. Фаулер М. Архитектура корпоративных программных приложений.: Пер. с англ. М.: Издательский дом «Вильямс», 2006. 544 с.: ил. С. 347.
3. Koha, описание. URL: <https://koha-community.org>
4. NewGenLib, описание. URL: <https://sourceforge.net/projects/newgenlib/>
5. Evergreen, описание. URL: <https://evergreen-ils.org>

6. Релиз Koha.
URL: <https://github.com/Koha-Community/Koha/releases/tag/v23.11.00>
 7. Релиз NewGenLib.
URL: <https://sourceforge.net/projects/newgenlib/files/NewGenLib/version3.1.5/>
 8. Релиз Evergreen. URL: <https://launchpad.net/evergreen/3.10/3.10.4>
 9. *Alghazali S.M.M., Polshchikov K., Hailan A.M., Svoykina L.* Development of Intelligent Tools for Detecting Resource-intensive Database Queries // International Journal of Advanced Computer Science and Applications. 2021. V. 12, No. 7. P. 32–36.
 10. *Labusch K. et al.* BERT for Named Entity Recognition in Contemporary and Historical German // Proceedings of the 15th Conference on Natural Language Processing. 2019. P. 9–11.
 11. *Гринберг М.* Разработка веб-приложений с использованием Flask на языке Python. Litres, 2022.
 12. *Меркулова А.Ш.* Автоматизированные библиотечно-информационные системы: практикум для обучающихся по направлению 51.03.06 «Библиотечно-информационная деятельность», квалификация (степень) выпускника «бакалавр»: учебное пособие. Кемерово: КемГИК, 2021. 130 с. ISBN 978-5-8154-0590-5. Текст: электронный // Лань: электронно-библиотечная система.
URL: <https://e.lanbook.com/book/250673>
 13. *Аракелян Э.К., Андрюшин А.В., Мезин С.В., Косой А.А.* Оценка оптимального уровня интеллектуальности АСУТП энергоблоков большой мощности на базе современных ПТК // Материалы XII Межд. конф. «Управление развитием крупномасштабных систем» / под общ. ред. С.Н. Васильева. М.: Ин-т проблем управления им. В.А. Трапезникова РАН, 2019. С. 574–576.
 14. *Gorodnyaya L.* Method of paradigmatic analysis of programming languages and systems // CEUR Workshop Proceedings. 2020. V. 2543. P. 149–158.
 15. *Шиняева О.В., Ахметшина Е.Р., Ключева Т.В. и др.* Информационно-цифровое неравенство населения и способы его преодоления в регионе: монография под редакцией О.В. Шиняевой. Ульяновск: УлГТУ, 2020. 201 с. ISBN 978-5-9795-2015-5. Текст: электронный // Лань: электронно-библиотечная система.
URL: <https://e.lanbook.com/book/259799>
-

16. *An Yang, Kai Liu, Jing Liu, Yajuan Lyu, Sujian Li.* Adaptations of ROUGE and BLEU to Better Evaluate Machine Reading Comprehension Task [Электронный ресурс] // arXiv.org. 2018. Дата обновления: 10.06.2018.

URL: <https://arxiv.org/abs/1806.03578>

17. *Воеводин В.В., Воеводин Вл.В.* Параллельные вычисления/ СПб.: БХВ-Петербург, 2002. 608 с.

18. Современные технологии в науке и образовании – СТНО-2018: Сборник трудов международного научно-технического форума: в 11 томах, Рязань, 28 февраля 2018 года / Под общ. ред. О.В. Миловзорова. Рязань: Рязанский государственный радиотехнический университет, 2018. 234 с.

19. *Городняя Л.В.* Перспективы функционального программирования параллельных вычислений // Электронные библиотеки. 2021. № 6. С. 1090–1116.

20. *Изместьева О.В., Матусевич Д.С.* Зарубежное свободное программное обеспечение автоматизированных библиотечно-информационных систем // Научные и технические библиотеки. 2020. № 3. С. 1090–1116.

21. *Ионов М.А.* Опыт миграции БЕН РАН на АБИС Коха // Электронный век науки. Информационное и ресурсное обеспечение научной деятельности в контексте цифровой трансформации: материалы I международной научно-практической конференции, Махачкала, 27 сентября – 1 октября 2021 года / Библиотека по естественным наукам Российской академии наук; Дагестанский государственный университет. Москва: Федеральное государственное бюджетное учреждение науки Библиотека по естественным наукам Российской академии наук, 2021. С. 12–16.

22. *Нещерет М.Ю.* Цифровая библиография: библиотеки в поисках инновационных инструментов библиографической деятельности // Научные и технические библиотеки. 2021. № 7. С. 33–50.

23. *Скрыпников А.В., Денисенко В.В., Хитров Е.Г., Евтеева К.С., Савченко И.И.* Распознавание рукописного текста с использованием нейронных сетей // Современные наукоемкие технологии. 2021. № 6-1. С. 91–95.

24. *Сукиасян Э.Р.* Библиотечно-библиографическая классификация (БК): современное состояние и перспективы // Научные и технические библиотеки. 2020. № 1. С. 60–75.

ANALYSIS OF SOFTWARE SYSTEM OPTIMIZATION USING THE EXAMPLE OF FREE AUTOMATED LIBRARY AND INFORMATION SYSTEMS

O. I. Vasyliiev¹ [0009-0001-2150-5951], V. Y. Medvedev² [0009-0008-3409-5462]

^{1, 2}*Department of Software Engineering and Artificial Intelligence Far Eastern Federal University;*

¹vasilev.o@dvmfu.ru, ²medvedev.viu@dvmfu.ru

Abstract

This article is devoted to the study of the possibilities of optimizing the operability and improving the efficiency of complex multifunctional software systems using the example of free automated library and information systems (hereinafter - ALIS).

By 2023, the world has accumulated valuable experience in the creation and operation of integrated ALIS of various scales and purposes, but the issues of improving their design solutions remain relevant. First of all, this concerns the need to optimize the structure of the source code in order to increase its readability and maintainability, reduce the execution time of individual functional modules, and reduce the amount of RAM used.

As part of the study, a comparative analysis of the source codes of several existing open source databases implemented in various programming languages was carried out. The main approaches to the design of the code structure were studied, the most frequently used algorithms and patterns were identified. To assess the degree of optimization of the source code, a set of indicators was developed, including an assessment of the structure, readability, modularity and other characteristics. On this basis, individual code fragments were compared before and after the use of well-known refactoring techniques.

As a result of the work carried out, it was possible to identify the most common errors and shortcomings in the structuring of the source codes of the ALIS, to determine the main directions of their optimization. Data has been obtained on the possible reduction of testing and technical support costs by improving the quality of source codes.

Keywords: *software code correction, software system optimization, refactoring, multilingual system, software system quality assessment, automated library and information systems, software development process.*

REFERENCES

1. *Fauler M.* Arhitektura korporativnyh programmnyh prilozhenij.: Per. s angl. M.: Izdatel'skij dom "Vil'jams", 2006. 544 p.: il. S. 46–48.
2. *Fauler M.* Arhitektura korporativnyh programmnyh prilozhenij.: Per. s angl. M.: Izdatel'skij dom "Vil'jams", 2006. 544 p.: il. S. 347.
3. Koha, opisanie. URL: <https://koha-community.org>
4. NewGenLib, opisanie. URL: <https://sourceforge.net/projects/newgenlib/>
5. Evergreen, opisanie. URL: <https://evergreen-ils.org>
6. Reliz Koha.
URL: <https://github.com/Koha-Community/Koha/releases/tag/v23.11.00>
7. Reliz NewGenLib.
URL: <https://sourceforge.net/projects/newgenlib/files/NewGenLib/version3.1.5/>
8. Reliz Evergreen. URL: <https://launchpad.net/evergreen/3.10/3.10.4>
9. *Alghazali S.M.M., Polshchikov K., Hailan A.M., Svojkina L.* Development of Intelligent Tools for Detecting Resource-intensive Database Queries // International Journal of Advanced Computer Science and Applications. 2021. V. 12, No. 7. P. 32–36.
10. *Labusch K. et al.* BERT for Named Entity Recognition in Contemporary and Historical German // Proceedings of the 15th Conference on Natural Language Processing. 2019. P. 9–11.
11. *Grinberg M.* Razrabotka veb-prilozhenij s ispol'zovaniem Flask na jazyke Python. Litres, 2022.
12. *Merkulova A.Sh.* Avtomatizirovannye bibliotechno-informacionnye sistemy: praktikum dlja obuchajushhihsja po napravleniju 51.03.06 «Bibliotechno-informacionnaja dejatel'nost'», kvalifikacija (stepen') vypusknika «bakalavr»: uchebnoe posobie. Kemerovo: KemGIK, 2021. 130 p. ISBN 978-5-8154-0590-5. Tekst: jelektronnyj // Lan': jelektronno-bibliotechnaja sistema. URL: <https://e.lanbook.com/book/250673>

13. *Arakeljan Je.K., Andrijushin A.V., Mezin S.V., Kosoj A.A.* Ocenka optimal'nogo urovnja intellektual'nosti ASUTP jenergoblokov bol'shoj moshhnosti na baze sovremennyh PTK // *Materialy XII Mezhd. konf. «Upravlenie razvitiem krupnomasshtabnyh sistem» / pod obshh. red. S.N. Vasil'eva. M.: In-t problem upravlenija im. V.A. Trapeznikova RAN, 2019. S. 574–576.*

14. *Gorodnyaya L.* Method of paradigmatic analysis of programming languages and systems // *CEUR Workshop Proceedings. 2020. V. 2543. P. 149–158.*

15. *Shinjaeva O.V., Ahmetshina E.R., Kljueva T.V. i dr.* Informacionno-cifrovoe neravenstvo naselenija i sposoby ego preodolenija v regione: monografija pod redakciej O.V. Shinjaevoj. Ul'janovsk: UIGTU, 2020. 201 p. ISBN 978-5-9795-2015-5. Tekst: jelektronnyj // Lan': jelektronno-bibliotechnaja sistema.

URL: <https://e.lanbook.com/book/259799>

16. *An Yang, Kai Liu, Jing Liu, Yajuan Lyu, Sujian Li.* Adaptations of ROUGE and BLEU to Better Evaluate Machine Reading Comprehension Task [Jelektronnyj resurs] // *arXiv.org. 2018. Data obnovlenija: 10.06.2018. URL: https://arxiv.org/abs/1806.03578*

17. *Voevodin V.V., Voevodin Vl.V.* Parallel'nye vychislenija/ SPb.: BHV-Peterburg, 2002. 608 s.

18. *Sovremennye tehnologii v nauke i obrazovanii – STNO-2018: Sbornik trudov mezhdunarodnogo nauchno-tehnicheskogo foruma: v 11 tomah, Rjazan', 28 fevralja 2018 goda / Pod obshh. red. O.V. Milovzorova. Rjazan': Rjazanskij gosudarstvennyj radiotehnicheskij universitet, 2018. 234 s.*

19. *Gorodnjaja L.V.* Perspektivy funkcional'nogo programirovanija paralel'nyh vychislenij // *Jelektronnye biblioteki. 2021. № 6. S. 1090–1116.*

20. *Izmest'eva O.V., Matusevich D.S.* Zarubezhnoe svobodnoe programmnoe obespechenie avtomatizirovannyh bibliotechno-informacionnyh sistem // *Nauchnye i tehnicheskie biblioteki. 2020. № 3. S. 69–78.*

21. *Ionov M.A.* Opyt migracii BEN RAN na ABIS Koha // *Jelektronnyj vek nauki. Informacionnoe i resursnoe obespechenie nauchnoj dejatel'nosti v kontekste cifrovoj transformacii: materialy I mezhdunarodnoj nauchno-prakticheskoy konferencii, Mahachkala, 27 sentjabrja – 1 oktjabrja 2021 goda / Biblioteka po estestvennym naukam*

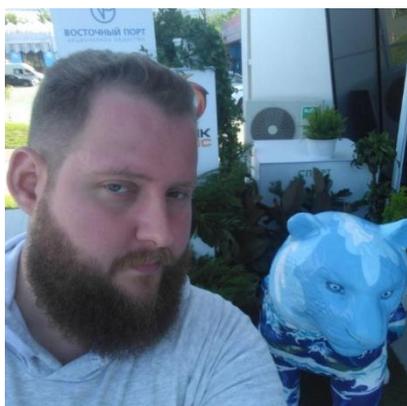
Rossijskoj akademii nauk; Dagestanskij gosudarstvennyj universitet. Moskva: Federal'noe gosudarstvennoe bjudzhetnoe uchrezhdenie nauki Biblioteka po estestvennym naukam Rossijskoj akademii nauk, 2021. S. 12–16.

22. *Neshheret M.Ju.* Cifrovaja bibliografija: biblioteki v poiskah innovacionnyh instrumentov bibliograficheskoj dejatel'nosti // Nauchnye i tehnicheckie biblioteki. 2021. № 7. S. 33–50.

23. *Skrypnikov A. V., Denisenko V. V., Hitrov E. G., Evteeva K.S., Savchenko I.I.* Raspoznavanie rukopisnogo teksta s ispol'zovaniem nejronnyh setej // Sovremennye naukoemkie tehnologii. 2021. № 6-1. S. 91–95.

24. *Sukiasjan Je.R.* Bibliotechno-bibliograficheskaja klassifikacija (BBK): sovremennoe sostojanie i perspektivy // Nauchnye i tehnicheckie biblioteki. 2020. № 1. S. 60–75.

СВЕДЕНИЯ ОБ АВТОРАХ



ВАСИЛЬЕВ Олег Иванович – аспирант, ассистент Департамента программной инженерии и искусственного интеллекта Института математики и компьютерных технологий Дальневосточного федерального университета.

Oleh Vasylijev – graduate student, assistant of the Department software engineering and artificial intelligence of the Institute of Mathematics and Computer Technologies Far Eastern Federal University.

email: vasilev.o@dvfu.ru

ORCID: 0009-0001-2150-5951

МЕДВЕДЕВ Валентин Юрьевич – аспирант Департамента программной инженерии и искусственного интеллекта Института математики и компьютерных технологий Дальневосточного федерального университета.

Valentin Medvedev – graduate student of the Department software engineering and artificial intelligence of the Institute of Mathematics and Computer Technologies Far Eastern Federal University.

email: medvedev.viu@dvfu.ru

ORCID: 0009-0008-3409-5462



Материал поступил в редакцию 25 марта 2024 года
