

РАЗРАБОТКА ПРОГРАММНОЙ ПЛАТФОРМЫ ДЛЯ ТЕСТИРОВАНИЯ ПРИКЛАДНЫХ РЕШЕНИЙ НА ОСНОВЕ ТЕХНОЛОГИИ КОНТЕЙНЕРНОЙ ВИРТУАЛИЗАЦИИ

Половикова О. Н.¹, канд. физ.-мат. наук, доцент, ✉ polovikovaol@yandex.ru,
orcid.org/0000-0002-9403-1195

Маничева А. С.¹, канд. физ.-мат. наук, доцент, manichevaas@gmail.com,
orcid.org/0000-0002-3820-0062

Журавлева В. В.¹, канд. физ.-мат. наук, доцент, vvzhuravleva@mail.ru,
orcid.org/0000-0002-8789-7476

¹ Алтайский государственный университет, пр. Ленина, д. 61, 656049, Барнаул, Россия

Аннотация

В данной статье анализируется проблема сбора и накопления готовых программных решений студентов с возможностью их непосредственного выполнения (тестирования). Для решения проблемы предлагается подход на основе технологии контейнерной виртуализации. Каждое программное решение автоматически разворачивается в изолированном docker-контейнере. Подход апробирован на практике. Спроектирована и разработана программная платформа, позволяющая размещать готовые прикладные разработки, тестировать варианты существующих решений разного уровня и сложности, анализировать применяемые технологии и подходы, оценивать работу используемых методов и алгоритмов. Программная платформа реализована в виде клиент-серверного web-приложения. Проблема размещения и запуска стека приложений без привлечения дополнительного серверного оборудования и ресурсозатратных технологий решается применением виртуализации на основе гипервизора. Виртуальная машина для разработанного web-приложения создана на базе программного гипервизора. Полученный в рамках исследования анализ видов виртуализации ресурсов позволяет делать выводы о применимости используемых технологий для решения других прикладных задач.

Ключевые слова: виртуализация ресурсов, вычислительные процессы, контейнер, образ, банк программных разработок, клиент-серверное приложение, Docker-система, Docker Hub.

Цитирование: Половикова О. Н., Маничева А. С., Журавлева В. В. Разработка программной платформы для тестирования прикладных решений на основе технологии контейнерной виртуализации // Компьютерные инструменты в образовании. 2023. № 3. С. 51-59. doi: 10.32603/2071-2340-2023-3-51-59

1. ВВЕДЕНИЕ

Информационное обеспечение образовательного процесса напрямую зависит от использования накопленного банка научно-исследовательских проектов студентов. На се-

годняшний день является актуальной задачей по систематизации, хранению и обеспечению доступности готовых проектов. В зависимости от типа и формата разработок можно подобрать и настроить прикладную программу, обеспечивающую часть необходимых функций по работе с проектами, но в целом для такой задачи пока не найдено универсального решения [1, с. 72].

Для управления этапами жизненного цикла разработки программных проектов существуют готовые решения. На рынке программных средств управления проектами в России, наряду с известными зарубежными пакетами, такими как Microsoft Project, Open Plan, Suretrak, Primavera Project Planner, присутствуют и российские пакеты Spider Project и Битрикс24 [2, с. 139]. Такие решения представляют собой комплексное программное обеспечение (ПО) с функциями: проектирования структуры работ и ресурсов, распределения ролей и задач между участниками, оценка и анализ стоимости, средства контроля за ходом работ и др. Данные комплексы программ предназначены для управления этапами моделирования и процессом разработки проектов (с учётом множества факторов), но не обеспечивают инструментами для публикации уже готовых приложений и их выполнения [3, 4].

Учебные заведения различного уровня готовят кадры по компетенциям в области программирования, проектирования и разработки прикладных программ, программной инженерии. Но, несмотря на то, что большинство итоговых проектов студентов являются готовыми прикладными решениями, отсутствуют системы их сбора, представления и тестирования. Нет возможности выполнить программный код и оценить используемые технологии и полученные результаты. Анализ проблемной области показал также отсутствие специализированных сред, позволяющих конструировать платформы для размещения на них прикладных программ.

Чтобы продемонстрировать реальную работоспособность программ, их заранее необходимо устанавливать, настраивать, обеспечивать необходимыми внешними и внутренними условиями выполнения. Такая многоэтапная процедура является трудозатратной и требует привлечения дополнительных технических и программных ресурсов. Поэтому задача по разработке специальной прикладной платформы с возможностью хранения и непосредственного выполнения готовых программ является актуальной и требует решения. Следует разработать сам подход с описанием необходимого стека технологий, на основе которого можно создавать подобные функциональные системы.

Основными функциями платформы с точки зрения пользователей являются: публикация, информационный поиск и выполнение программ. Базовыми этапами разработки платформы являются:

1. Выбор технологии, которая в рамках единой системы будет обеспечивать обособленную работу отдельных программных решений.
2. Проектирование и разработка клиент-серверного приложения для размещения и обслуживания банка программных решений студентов на основе выбранной технологии.
3. Накопление банка программных решений: частичная автоматизация данных процессов средствами web-интерфейса.

2. ВЫБОР ТЕХНОЛОГИЙ ВИРТУАЛИЗАЦИИ

Задача по созданию специализированной программной платформы предполагает использование технологий виртуализации, которые могут размещать приложения в вир-

туальных машинах или контейнерах. Каждое готовое программное решение нужно развернуть в изолированной среде с заранее предустановленным набором ПО и библиотек. В этом случае выполнение программы также осуществляется внутри этой среды и не будет зависеть от настроек и ресурсов исполняющей системы.

Следовательно, нужны технологии, позволяющие генерировать обособленные среды, контролируемые и настраиваемые внешними инструментами. Под виртуализацией будем понимать технологию, которая предоставляет возможность создания набора вычислительных ресурсов, например серверы, устройства хранения данных или сетевые ресурсы, абстрагированные от их аппаратной реализации [5, с. 132].

Виртуализация позволяет создавать изолированные окружения в рамках одного физического устройства, каждое такое окружение может обеспечивать приемлемую среду гостевой операционной системы и обособленную работу отдельной прикладной программы. Технологии виртуализации в настоящее время активно развиваются и прогрессируют [6, с. 182].

Анализ исследований и практических примеров использования технологий виртуализации вычислительных ресурсов позволил выделить основные направления их развития: виртуализация на базе программного гипервизора, виртуализация на базе аппаратного гипервизора, паравиртуализация, полная и контейнерная.

Технология контейнерной виртуализации создает независимые пользовательские окружения на уровне базовой операционной системы. Контейнер — это помещенный на виртуальный диск файл, в который упаковывается приложение со всеми необходимыми для его работы зависимостями: кодом приложения, средой запуска, системными инструментами, библиотеками и настройками. В процессе работы каждый контейнер адаптируется под конкретные потребности установкой нужной ОС и ПО [5, 7, 8].

В отличие от других видов виртуализации, при использовании данной технологии не нужно клонировать новые ОС, разграничивать гипервизорами аппаратные ресурсы компьютера, а также затрачивать ресурсы для эмуляции взаимодействия гостевой ОС с физическим оборудованием. Меньшая ресурсозатратность, работа под управлением ядра основной ОС позволяет создавать намного больше контейнеров, чем гостевых ОС (или виртуальных машин).

Контейнеры скачиваются и запускаются по запросу специальным программным обеспечением: Docker, CRI-O, Railcar, RKT, LXC, используя API-обращения. Опираясь на результаты проведенного анализа для разработки платформы программных решений, было принято решение использовать контейнерную виртуализацию на базе Docker-системы. Набор базовых образов (репозиторий), который поддерживается Docker-системой, непрерывно обновляется и отвечает современным требованиям по разработке ПО. Поэтому можно будет запускать приложения, написанные на разных языках программирования и с применением разнообразных технологий.

Кроме этого, Docker-система обеспечивает весь жизненный цикл работы с контейнерами от загрузки базовых образов до выполнения контейнеров в низкоуровневой среде [9, 10]. Следует также заметить, что все основные контейнерные инструменты поддерживают стандарт OCI (открытый стандарт спецификации образов и контейнеров), что обеспечивает переносимость контейнеров и их платформ.

3. РАЗРАБОТКА КЛИЕНТ-СЕРВЕРНОГО WEB-ПРИЛОЖЕНИЯ

Проблема размещения и запуска стека приложений портала без привлечения дополнительного серверного оборудования и ресурсозатратных технологий решается приме-

нением виртуализации на основе гипервизора. Разновидности виртуализации на базе гипервизора позволяют «создавать» отдельный компьютер со своей операционной системой и обеспечивать его независимую работу, используя при этом ресурсы основной host-машины (процессор, несколько видов памяти и т. д.). Виртуальная машина для клиент-серверного web-приложения создана на базе программного гипервизора с открытым исходным кодом Proxmox [11] на host-машине внутри корпоративной сети. Это позволяет управлять виртуальными машинами средствами самого гипервизора без дополнительного ПО, но обладает достаточно низкой скоростью работы в сравнении с другими видами виртуализации [5].

Для создания, запуска и управления docker-контейнерами на виртуальной машине установлено необходимое ПО: Docker, Docker Compose, Ubuntu server, Nginx, Portainer. Клиентская часть web-приложения написана на языках: JavaScript (библиотека с открытым исходным кодом React) и TypeScript, серверная часть — на языке C# (фреймворк ASP.Net Core). Для хранения настроек пользователей приложения, свойств проектов и параметров контейнеров используется СУБД с открытым исходным кодом PostgreSQL.

4. НАПОЛНЕНИЕ БАНКА ПРОГРАММНЫХ РЕШЕНИЙ

Построение docker-контейнеров происходит в полуавтоматическом режиме при участии авторов проектов. После авторизации на портале и заполнения аннотации о проекте следует перейти в конструктор для создания и размещения стека docker-контейнеров «Деплой проекта» (см. рис. 1).

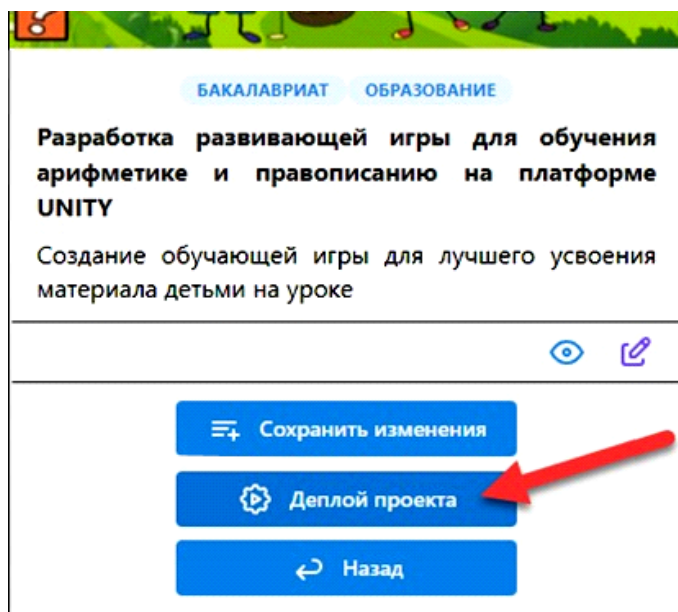


Рис. 1. Интерфейс работы с порталом для размещения прикладной программы

Основные действия, которые должны выполнить пользователи платформы для публикации своего проекта:

- заполнить по шаблону конфигурационный файл (docker-compose.yml) для стека контейнеров (см. рис. 2);

- указать порты для внутреннего веб-сервера, на который будут приходить запросы от приложения.

```
version: "2"
services:
  td1_zilsite:
    hostname: zilsite
    container_name: td1_zilsite
    image: ailesangen/zilsite:latest
    restart: always
  td1_nginx:
    image: nginx:stable-alpine
    container_name: td1_nginx
    restart: always
    environment:
      - "TZ=Asia/Barnaul"
    ports:
      - "6000:80"
    volumes:
      - /var/deploystacks/emilkraepelin/nginx.conf:/etc/nginx/nginx.conf
```

Рис. 2. Пример конфигурационного файла

Остальные этапы выполняются сервисами и службами платформы и выполняются в автоматическом режиме. В конфигурационном файле описывается вся логика взаимодействия контейнеров между собой и внешней средой. Здесь указываются необходимые базовые образы, на основе которых будет выполняться сборка стека (кластера контейнеров). Стек представляет собой несколько взаимосвязанных контейнеров, которыми можно управлять как единым элементом. Необходимые базовые образы размещены на платформе Docker Hub. Пример шаблона для заполнения конфигурационного файла представлен на рисунке 2. Выбор конкретных базовых образов зависит от среды, в которой должна выполняться программа, то есть от требуемых для работы приложения операционной системы, библиотек, утилит, языков программирования и т. д. На официальных ресурсах разработчиков ПО и в открытых репозиториях представлены образы как для распространённых прикладных систем, так и для редких программных средств. По официальным данным в Docker Hub уже загружено свыше 100 000 образов.

После создания конфигурационного файла следует приступить к сборке docker-контейнеров (кластера). Управление сборкой осуществляется через web-интерфейс в личном кабинете на портале:

- 1) пользователь посылает команду на создание триггера на Docker-сервере;
- 2) сервер автоматически разворачивает все контейнеры и сообщает пользователю о результате выполнения.

В результате выполнения этих этапов пользователь получает программу, готовую для использования в специальном окружении, и ссылку для запуска контейнера.

Как правило, студенты для создания своих докеров пользуются уже существующими официальными образами, к которым монтируют свои собственные файлы настройки и данные для приложения (дописывают слои). Дополнительного исследования требует выбор базовых образов для своих реализаций, так как даже в рамках одного используемого прикладного обеспечения сложно определить, с какого базового образа следует начинать собирать свой docker-контейнер. Можно использовать «толстые» образы с широким набором инструментов для разработки или «тонкие», в которых имеются минимальные установки и настройки для операционной системы и ПО. Чтобы сформировать

предпочтения в выборе базовых образов и корректно прописывать конфигурационные файлы для сборки своих образов, нужен опыт решения подобных задач.

В программу нескольких учебных курсов бакалавриата и магистратуры Института математики и информационных технологий Алтайского университета включены задания по созданию docker-контейнеров на базе различного ПО, библиотек, фреймворков и языков (unity-проекты, web-сайты, desktop-приложения и т. д.). В рамках этих курсов следует создать готовый программный продукт (индивидуально или в команде) и продемонстрировать его возможности. Одним из завершающих этапов разработки является упаковка готового приложения в docker-контейнер с необходимыми параметрами среды окружения, чтобы можно было продемонстрировать работу программы в любой системе без каких-либо дополнительных загрузок, установок и настроек. Поэтому процесс контейнеризации приложения для своего выпускного проекта студентами выполняется самостоятельно с минимальным участием преподавателя.

Опыт упаковки программного приложения в контейнер в рамках некоторого набора программного обеспечения позволяет решать подобные задачи уже в полуавтоматическом режиме с использованием сервисов платформы. В помощь студенту также предлагается несколько шаблонов конфигурационных файлов для создания кластера контейнеров под различные платформы и ПО. Несмотря на то, что есть некоторые тонкости описания, сборки и выполнения стека для прикладных программ, основанных на разных технологиях и языках программирования, эти процессы формализуются и имеют единый унифицированный алгоритм реализации.

5. ЗАКЛЮЧЕНИЕ

Выполненный анализ технологий виртуализации позволяет сформировать необходимое представление о специфике их практического применения для решения других прикладных задач. Контейнерная виртуализация, с точки зрения набора необходимых технологий, обеспечивается всеми этапами жизненного цикла, она менее требовательна к используемым ресурсам по сравнению с другими видами [12, с. 23]. Поэтому данный вид виртуализации выбран в качестве основной технологии для разработки специализированной программной платформы.

Предложенный в рамках исследования подход создания платформы программных решений апробирован на практике [13]. Пользователям предоставлена возможность размещать готовые программы, тестировать варианты существующих решений разного уровня и сложности, анализировать применяемые технологии и подходы, оценивать работу используемых методов и алгоритмов. Сервисы платформы обеспечивают выполнение прикладных программ в изолированных контейнерах, отвечают за взаимодействие между ними, контролируют использование docker-контейнерами вычислительных ресурсов. На рис. 3 представлен результат запуска на выполнение проекта (контейнера), созданного на платформе Unity и опубликованного на платформе.

Автоматизация сервисов системы виртуализации позволяет пользователям абстрагироваться от процесса сборки, загрузки и запуска приложений на сервере. Реализуемые сервисы не предоставляют пользователям прямого доступа к ресурсам сервера, что повышает безопасность разработанной платформы. Предложенный подход позволяет проектировать и разрабатывать высокоуровневые системы виртуализации, в которых этапы взаимодействия с пользователями выполняются и контролируются web-интерфейсами.



Рис. 3. Демонстрация работы сервисов платформы

Анализ возникающих ошибок и недочётов в работе всех сервисов платформы позволил сделать вывод, что основным проблемным местом является сборка готовых образов для контейнера согласно конфигурационному файлу. Большинство ошибок связано с использованием студентами в своих реализациях сторонних библиотек и интерфейсов, которые не входят в базовую версию ПО.

Список литературы

1. Осокина Е. В., Брюховских И. Е., Масликова Т. А. Опыт использования банка инновационных бизнес-идей в подготовке инженерных кадров // Вестник Шадринского государственного педагогического университета. 2020. № 4(48). С. 72–74.
2. Супчинский О. П., Капустьян М. Ф. Комплексный подход при организации и планировании производственных процессов на основе сетевого планирования и «Облачных технологий» // Омский научный вестник. 2014. № 3. С. 138–142.
3. Горяев И. С. Критерии выбора программных средств управления ИТ-проектами // Евразийский Союз Ученых. 2018. № 1. С. 8–10.
4. Вайнилович Ю. В., Захарченков К. В. Программный комплекс многоуровневого управления it-проектами // Инновации. 2019. № 8(250). С. 88–96.
5. Михалев П. С. Анализ современной технологии виртуализации // Интеллектуальный потенциал XXI века: ступени познания. 2014. № 25. С. 131–135.
6. Детков Е. А., Коревых А. А. Виртуализация и его особенности // Международный журнал прикладных наук и технологий «Integral». 2019. № 3. С. 181–193.
7. Ibrahim M. H., Sayagh M., Hassan A. E. A study of how Docker Compose is used to compose multi-component systems // Empir Software Eng. 2021. Vol. 26, № 6. P. 128. doi:10.1007/s10664-021-10025-1
8. Квасницкий В. Н., Журавлева Т. Б. Использование технологии виртуализации при создании информационных систем // Вестник РГГУ. Серия «Экономика. Управление. Право». 2012. № 10. С. 162–169.
9. Jangla K. Accelerating Development Velocity Using Docker. Apress Berkeley, CA. 2018.
10. Cook J. Docker Hub. In: Docker for Data Science. Apress, Berkeley, CA. 2017. P. 103–118. doi:10.1007/978-1-4842-3012-1_6
11. Chang B. R., Tsai H. F., Tsai Y. C. Applying authentication and network security to in-cloud enterprise resource planning system // Vietnam Journal of Computer Science. 2014. Vol. 1, № 2. P. 129–140. doi:10.1007/s40595-014-0015-8

12. Тяжельников Д. Н., Токарев П. А., Петров И. Д. Виртуализация рабочего пространства с ускорением 3d-приложений на стороне сервера при помощи docker // Проблемы современной науки и образования. 2017. № 14 С. 21–23.
13. Портал программных разработок «Проекты АЛТГУ». URL: <https://dockermf.asu.ru/projects> (дата обращения: 15.05.2023).

Поступила в редакцию 14.06.2023, окончательный вариант — 27.07.2023.

Половикова Ольга Николаевна, кандидат физико-математических наук, доцент, доцент кафедры информатики, Алтайский государственный университет, ✉ polovikovaol@yandex.ru

Маничева Анастасия Станиславовна, кандидат физико-математических наук, доцент, доцент кафедры теоретической кибернетики и прикладной математики, Алтайский государственный университет, manichevaas@gmail.com

Журавлева Вера Владимировна, кандидат физико-математических наук, доцент, доцент кафедры теоретической кибернетики и прикладной математики, Алтайский государственный университет, vvzhuravleva@mail.ru

Computer tools in education, 2023

№ 3: 51–59

<http://cte.eltech.ru>

[doi:10.32603/2071-2340-2023-3-51-59](https://doi.org/10.32603/2071-2340-2023-3-51-59)

Development of a Software Platform for Testing Application Solutions Based on Container Virtualization Technology

Polovikova O. N.¹, Cand. Sc., Docent, ✉ polovikovaol@yandex.ru, orcid.org/0000-0002-9403-1195

Manicheva A. S.¹, Cand. Sc., Docent, manichevaas@gmail.com, orcid.org/0000-0002-3820-0062

Zuravleva V. V.¹, Cand. Sc., Docent, vvzhuravleva@mail.ru, orcid.org/0000-0002-8789-7476

¹Altai State University, 61 Lenin Ave., 656049, Barnaul, Russia

Abstract

This article analyzes the problem of collecting and storing the program solutions of students that can be directly executed, tested. Container-based virtualization technologies are proposed to solve the problem. Each software development is automatically deployed in one or more isolated docker-containers. The approach has been tested in practice. A software platform for publishing ready-made application solutions has been designed and developed. The platform services allow testing ready-made developments of different level and complexity, analyzing the applied technologies and approaches, evaluating the performance of the used methods and algorithms. The software platform is implemented on the basis of client-server web- application. The problem of deploying and running the portal application stack without additional server hardware and resource-intensive technologies is solved by using hypervisor-based virtualization. The virtual machine for the developed web- application was created on the basis of a software hypervisor. The analysis of resource virtualization types obtained in the framework of the research allows

us to draw conclusions about the applicability of the used technologies for solving other practical tasks.

Keywords: *resource virtualization, computing processes, container, image, software development base, client-server application, Docker-system, Docker Hub.*

Citation: O. N. Polovikova, A. S. Manicheva, and V. V. Zuravleva, "Development of a Software Platform for Testing Application Solutions Based on Container Virtualization Technology," *Computer tools in education*, no. 3, pp. 51–59, 2023 (in Russian); doi:10.32603/2071-2340-2023-3-51-59

References

1. E. V. Osokina, I. E. Bryukhovskikh, and T. A. Maslikova, "Experience in using the Bank of innovative business ideas in training engineering personnel," *Journal of Shadrinsk State Pedagogical University*, no. 4(48), pp. 72–74, 2020 (in Russian).
2. O. P. Supchinsky and M. F. Kapustyan, "An integrated approach to organizing and planning production processes based on network planning and "Cloud technologies" *Scientific journals of OMSTU*, no. 3, pp. 138–142, 2014 (in Russian).
3. I. S. Goryaev, "Criteria of choice of programmatic facilities of management of IT-project," *Eurasian Union of Scientists*, no. 1, pp. 8–10, 2018 (in Russian).
4. V. Vajnilovich and K. V. Zakharchenkov, "Software system for multi-level the IT projects management," no. 8(250), pp. 88–96, 2019 (in Russian); doi:10.26310/2071-3010.2019.250.8.014
5. P. S. Mikhalev, "Analysis of modern virtualization technology," *Intellectual potential of the 21st century: stages of knowledge*, no. 25, pp. 131–135, 2014 (in Russian).
6. E. A. Detkov and A. A. Korevyh, "Virtualization and its features," *International Journal of Applied Science and Technology INTEGRAL*, no. 3, pp. 181–193, 2019 (in Russian).
7. M. H. Ibrahim, M. Sayagh, and A. E. Hassan, "A study of how Docker Compose is used to compose multi-component systems," *Empir Software Eng.*, vol. 26, no. 6, p. 128, 2021 (in Russian); doi:10.1007/s10664-021-10025-1
8. V. Kvasnitsky and T. Zhuravlyova, "Use of virtualization technology for creation of information systems," *RSUH/RGGU BULLETIN. Series Economics. Management. Law*, no.10, pp. 162–169, 2012 (in Russian).
9. K. Jangla, *Accelerating Development Velocity Using Docker*, Berkeley, CA, USA: Apress, 2018; doi:10.1007/978-1-4842-3936-0
10. J. Cook, "Docker Hub," in *Docker for Data Science*, Berkeley, CA, USA: Apress, pp. 103–118, 2017; doi:10.1007/978-1-4842-3012-1_6
11. B. R. Chang, H.-F. Tsai, Y.-C. Tsai, and Y.-S. Chang, "Applying authentication and network security to in-cloud enterprise resource planning system," *Vietnam Journal of Computer Science*, vol. 1, no. 2, pp. 129–140, 2014; doi:10.1007/s40595-014-0015-8
12. D. N. Tyazhelnikov, P. A. Tokarev, and I. D. Petrov, "Virtualization of the workspace with the acceleration of 3d applications on the server side using Docker," *Modern problems of science and education*, no. 14, pp. 21–23, 2017 (in Russian).
13. Altai State University, "Software development portal "Altai State University Projects" in *dockermf.asu.ru*, 2023 (in Russian). [Online]. Available: <https://dockermf.asu.ru/projects>

Received 14-06-2023, the final version — 27-07-2023.

Olga Polovikova, Candidate of Sciences (Phys.-Math.), Docent, Altai State University,
✉ polovikovaol@yandex.ru

Anastasia Manicheva, Candidate of Sciences (Phys.-Math.), Docent, Altai State University,
manichevaas@gmail.com

Vera Zuravleva, Candidate of Sciences (Phys.-Math.), Docent, Altai State University,
vzhuravleva@mail.ru