DOI: 10.12731/2227-930X-2023-13-2-33-46
УДК 004.62

Научная статья | Системный анализ, управление и обработка информации

АВТОИМПОРТ БОЛЬШОГО ОБЪЕМА ИНФОРМАЦИИ В БАЗУ ДАННЫХ С ПОМОЩЬЮ ЯЗЫКА ПРОГРАММИРОВАНИЯ PYTHON

Р.Р. Крапивин, Г.А. Гареева, Ю.М. Филатов, А.Г. Файзуллина, И.Ю. Мышкина

В статье рассматривается эффективный и автоматизированный способ импортирования больших объемов данных из таблиц Excel в базу данных. В различных проектах присутствуют задачи, в которых поток огромных данных, таких как лог-файлы программных операций или ручные операции, совершаемые на рабочих участках, жизненно необходим для эффективного анализа.

***Цель** – разработка модуля для автоматического импортирования большого объема данных из формата Excel в базу данных.*

***Метод или методология проведения работы:** в статье рассматривается способ, который реализует автоматическое импортирование данных из таблиц Excel в базу данных Postgresql.*

***Результат:** разработан собственный уникальный модуль, который способен обрабатывать огромные Excel таблицы и импортировать их в базу данных Postgresql без ручных операций.*

***Ключевые слова:** Python; pandas; библиотека; запрос; Postgresql; xlsx; Excel*

***Для цитирования.** Крапивин Р.Р., Гареева Г.А., Филатов Ю.М., Файзуллина А.Г., Мышкина И.Ю. Автоимпорт большого объема информации в базу данных с помощью языка программирования Python // International Journal of Advanced Studies. 2023. Т. 13, № 2. С. 33-46. DOI: 10.12731/2227-930X-2023-13-2-33-46*

Original article | System Analysis, Management and Information Processing

AUTOIMPORT OF A LARGE VOLUME INFORMATION INTO A DATABASE USING THE PYTHON PROGRAMMING LANGUAGE

R.R. Krapivin, G.A. Gareeva, Y.M. Filatov, A.G. Faizullina, I.Yu. Myshkina

The article discusses an efficient and automated way to import large amounts of data from Excel tables into a database. In various projects, there are tasks in which the flow of huge data, such as logs of program operations or manual operations performed at work sites, is vital for effective analysis.

***Purpose** – development of a module for automatic import of a large amount of data from Excel format into a database.*

***Method or methodology of work:** the article discusses a method that implements automatic import of data from Excel tables into a Postgresql database.*

***Result:** developed its own unique module that is able to process huge Excel tables and import them into a Postgresql database without manual operations.*

***Scope of the results:** the data obtained, which are stored in the database, should be used to identify high-yield accounts for subsequent investment.*

***Keywords:** Python; pandas; library; query; Postgresql; xlsx; Excel*

***For citation.** Krapivin R.R., Gareeva G.A., Filatov Y.M., Faizullina A.G., Myshkina I.Yu. Autoimport of a Large Volume Information into a Database Using the Python Programming Language. International Journal of Advanced Studies, 2023, vol. 13, no. 2, pp. 33-46. DOI: 10.12731/2227-930X-2023-13-2-33-46*

Введение

В любых компаниях существует информационный поток данных, зачастую реализуемый в ручном режиме. Под потоком

данных подразумевается обмен или передача больших таблиц с данными. Так же такой большой объем данных используется для аналитики, используемой в профессии Data science для выявления закономерностей, позволяющие улучшить или оптимизировать работу компании.

В компаниях с большим потоком информации затрачиваются большие средства для выполнения импорта таблиц тех же Excel книг в базу данных.

Автоматизация импортирования такого потока, заметно снижает нагрузку на работников, которые оперируют данными.

Из-за постоянного и активного роста IT сектора, подобные задачи по автоматизации импорта Excel книг в базы данных довольно часто встречаются и ложатся на плечи обычных разработчиков.

В статье рассматривается способ реализации механизма импорта на популярном и распространённым высокоуровневом языке программирования Python, что может помочь сэкономить время и деньги на разработке.

Цель работы: разработать программное обеспечение для автоматизированного импорта таблиц Excel в базу данных Postgresql.

Для автоматизации импорта требуется рабочее пространство, например рабочая папка, в которой будут находиться исполняющий Excel файл и Excel таблицы, которые необходимо импортировать в базу данных. Для реализации подобного исполняющего файла необходимо выполнить три шага:

Первое, в исполняющем файле необходимы библиотеки для работы (рис. 1).

Второе, научить исполняющий файл находить Excel книги в ближайшем окружении.

Третье, импортировать каждую Excel книгу в базу данных.

Для первого шага достаточно прописать 4 строки, вызывая эти библиотеки в рабочее окружение исполняющего файла.

```
import pandas as pd
import sqlalchemy
from sqlalchemy import create_engine

import os
```

Рис. 1. Библиотеки для импортирования данных

Рассмотрим каждый из импортированных модулей. Pandas – высокоуровневая Python библиотека для анализа данных. Это высокоуровневый модуль, потому что она построена поверх более низкоуровневой библиотеки NumPy. В экосистеме Python Pandas является наиболее продвинутой и быстроразвивающейся библиотекой для обработки и анализа данных. Библиотека была создана в 2008 году компанией AQR Capital, в 2009 году она стала проектом с открытым исходным кодом. По большей части эта библиотека используется для аналитики, но в рамках данной статьи она используется для создания DataFrame массивов, которые с помощью sqlalchemy импортируются в базу данных. DataFrame – основной тип данных в Pandas, вокруг которого строится вся работа. Его можно представить в виде обычной таблицы с любым количеством столбцов и строк. Внутри ячеек такой «таблицы» могут быть данные самого разного типа: числовые, булевы, строковые и так далее.

SQLAlchemy – это фреймворк для работы с реляционными базами данных в Python. Он был создан Майком Байером в 2005 году. SQLAlchemy позволяет работать с базами данных MySQL, MS-SQL, PostgreSQL, Oracle, SQLite и другими. В рамках статьи она используется для формирования подключения к базе данных PostgreSQL.

Create_engine это модуль SQLAlchemy или же пул соединений.

Пул соединений – это стандартный способ кэширования соединений в памяти, что позволяет использовать их повторно. Соз-

давать соединение каждый раз при необходимости связаться с базой данных – затратно. А пул соединений обеспечивает неплохой прирост производительности.

При таком подходе приложение при необходимости обратится к базе данных и затем получить пул подключения. После выполнения запросов подключение освобождается и возвращается в пул. Новое создается только в том случае, если все остальные связаны. В исполняющем файле это основной механизм.

Библиотека os – это стандартный встроенный модуль языка программирования Python. Эта библиотека функций для работы с операционной системой. Методы, включенные в неё, позволяют определять тип операционной системы, получать доступ к переменным окружения, управлять директориями и файлами. Для исполняющего файла эта библиотека позволит находить Excel файлы.

Для выполнения первого шага достаточно прописать import библиотек в начале исполняющего файла (рис. 1).

Для второго шага требуется использовать библиотеку os. Необходимо просматривать все Excel файлы в папке, где находится исполняющий файл, и игнорировать другие форматы (рис. 2). У библиотеки есть вызываемый метод listdir, с помощью которого исполняющий файл получает список файлов в папке, в котором сам находится. После получения списка файлов, каждый из них проверяется на наличие строки .xlsx, которая означает Excel формат.

```
def get_excel_list():
    excel_list = []
    for file in os.listdir():
        if '.xlsx' in file:
            excel_list.append(file)
    return excel_list
```

Рис. 2. Функция помещение Excel таблиц в массив excel_list

Для реализации третьего шага, самого главного, полученный ранее список Excel таблиц необходимо преобразовать в DataFrame и создать с помощью engine пул подключения для импортирования DataFrame в таблицы базы данных (рис. 3).

```
def import_excel_in_bd(excel_list):
    engine = create_engine(f'postgresql://{user_info}:{password_info}@{host_info}:{port}/{dbname_info}')
    for excel_file in excel_list:# Начало прохода по массиву excel list
        df = pd.read_excel(excel_file)# Чтение excel
        df.to_sql(excel_file, engine, if_exists = 'replace', index = False)# Создание таблицы, если она
```

Рис. 3. Импортирование Excel файлов в базу данных

Метод read_excel показанный на рисунке 3, создает Dataframe из Excel таблицы, а следующий метод to_sql выполняет функцию импортирования, но для его работы требуется engine, пул подключения, который создается методом create_engine и на вход получает аргументы из данных для подключения, таких как:

1) user_info – переменная которая хранит имя пользователя базы данных;
2) password_info – переменная хранящая пароль для подключения;
3) host_info – переменная содержащая имя хоста, где располагается база данных;
4) port – переменная хранящая номер порта;
5) dbname_info – переменная, хранящая имя базы данных, к которой нужно подключение.

Такие переменные являются очень важными и объявляются в отдельной функции как глобальные переменные, чтобы можно было вызывать их в любом участке кода (рис. 4).

Каждый Excel файл будет импортирован в таблицу, имена таблиц берутся из названия самих Excel файлов.

Например, поместим два Excel файла в папку с исполняющим файлом и запустим его (рис. 5).

Каждый Excel содержит ежемесячные расходы одного человека (рис. 6).

```python
def create_global_params():
    'Задаются глобальные параметры'
    global user_info
    global password_info
    global host_info
    global port
    global dbname_info

    user_info = 'postgres'
    password_info= 'pass123'
    host_info = 'localhost'
    port = 5432
    dbname_info = 'postgres'
```

Рис. 4. Создание и объявление глобальных переменных для подключения к базе данных

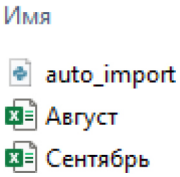

Рис. 5. Excel таблицы в папке с исполняющем файлом

	A	B	C	D	E	F	G	H	I
1	Дата	Аренда	Коммуналка	Кафе, прогулки	Продукты/Еда	Техника/Оборудова	Внечеков	Прочее	Итого
2	01.08.2022			296,99 ₽		246,89 ₽			40 167,98 ₽
3	02.08.2022				717,90 ₽				
4	03.08.2022			1 143,25 ₽			33,5		
5	04.08.2022								
6	05.08.2022	12 000,00 ₽	2 459,00 ₽						
7	06.08.2022			1 848,85 ₽					
8	07.08.2022			1 287,67 ₽					
9	08.08.2022						121,00 ₽		
10	09.08.2022				861,94 ₽				
11	10.08.2022			240,00 ₽					
12	11.08.2022								
13	12.08.2022			644,86 ₽				420,00 ₽	
14	13.08.2022							177,50 ₽	
15	14.08.2022			1 735,50 ₽					
16	15.08.2022								
17	16.08.2022								
18	17.08.2022								
19	18.08.2022				300,00 ₽				
20	19.08.2022					3 662,00 ₽	100,00 ₽		
21	20.08.2022			780,86 ₽					
22	21.08.2022								
23	22.08.2022				1 722,13 ₽		700,00 ₽		
24	23.08.2022			1 952,99 ₽					

Лист1

Рис. 6. Содержимое Excel таблицы Август.xlsx

После выполнения работы исполняющего файла, появятся две таблицы в базе данных (рис. 7, 8).

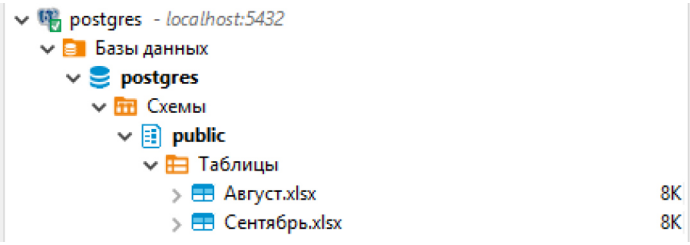

Рис. 7. Созданные таблицы в Postgresql

Август.xlsx

	Дата	123 Аренда	123 Коммуналка	123 Кафе, прогулки	123 Продукты/Еда	123 Техника/Оборудование
1	2022-08-01 00:00:00.000	[NULL]	[NULL]	296,99	[NULL]	246,89
2	2022-08-02 00:00:00.000	[NULL]	[NULL]	[NULL]	717,9	[NULL]
3	2022-08-03 00:00:00.000	[NULL]	[NULL]	1 143,25	[NULL]	[NULL]
4	2022-08-04 00:00:00.000	[NULL]	[NULL]	[NULL]	[NULL]	[NULL]
5	2022-08-05 00:00:00.000	12 000	2 459	[NULL]	[NULL]	[NULL]
6	2022-08-06 00:00:00.000	[NULL]	[NULL]	1 848,85	[NULL]	[NULL]
7	2022-08-07 00:00:00.000	[NULL]	[NULL]	1 287,67	[NULL]	[NULL]
8	2022-08-08 00:00:00.000	[NULL]	[NULL]	[NULL]	[NULL]	[NULL]
9	2022-08-09 00:00:00.000	[NULL]	[NULL]	[NULL]	861,94	[NULL]
10	2022-08-10 00:00:00.000	[NULL]	[NULL]	240	[NULL]	[NULL]
11	2022-08-11 00:00:00.000	[NULL]	[NULL]	[NULL]	[NULL]	[NULL]
12	2022-08-12 00:00:00.000	[NULL]	[NULL]	644,86	[NULL]	[NULL]
13	2022-08-13 00:00:00.000	[NULL]	[NULL]	[NULL]	[NULL]	[NULL]
14	2022-08-14 00:00:00.000	[NULL]	[NULL]	1 735,5	[NULL]	[NULL]
15	2022-08-15 00:00:00.000	[NULL]	[NULL]	[NULL]	[NULL]	[NULL]
16	2022-08-16 00:00:00.000	[NULL]	[NULL]	[NULL]	[NULL]	[NULL]
17	2022-08-17 00:00:00.000	[NULL]	[NULL]	[NULL]	[NULL]	[NULL]
18	2022-08-18 00:00:00.000	[NULL]	[NULL]	[NULL]	300	[NULL]
19	2022-08-19 00:00:00.000	[NULL]	[NULL]	[NULL]	[NULL]	3 662
20	2022-08-20 00:00:00.000	[NULL]	[NULL]	780,86	[NULL]	[NULL]
21	2022-08-21 00:00:00.000	[NULL]	[NULL]	[NULL]	[NULL]	[NULL]
22	2022-08-22 00:00:00.000	[NULL]	[NULL]	[NULL]	1 722,13	[NULL]
23	2022-08-23 00:00:00.000	[NULL]	[NULL]	1 952,99	[NULL]	[NULL]
24	2022-08-24 00:00:00.000	[NULL]	[NULL]	1 204,69	[NULL]	[NULL]
25	2022-08-25 00:00:00.000	[NULL]	[NULL]	893,98	[NULL]	[NULL]
26	2022-08-26 00:00:00.000	[NULL]	[NULL]	[NULL]	[NULL]	[NULL]
27	2022-08-27 00:00:00.000	[NULL]	[NULL]	[NULL]	1 278,88	[NULL]

Рис. 8. Содержимое таблицы Август

Результаты работы

В ходе проведенного исследования разработано программное средство для автоматического импортирования Excel таблиц в базу данных Postgresql на основе языка программирования Python и библиотек pandas, sqlalchemy, os и модуля create_engine. Подобная разработка может стать основой или важной частью для будущих проектов в разных компаниях. Полученные результат

является примером, доработка которого под конкретные задачи компаний позволит обрабатывать огромный объем информации в компаниях или различных веб сервисов.

Список литературы

1. Логинова Е.В. Необходимость изучения информационных потоков предприятия / Е.В.Логинова, Т.А. Сарыева // Проблемы современной науки и образования, 2017. – № 2. С. 45-48.
2. Методы и модели исследования сложных систем и обработки больших данных: Монография / И. Ю. Парамонов, В. А. Смагин, Н. Е. Косых, А. Д. Хомоненко; под редакцией В. А. Смагина и А. Д. Хомоненко. – Санкт-Петербург: Лань, 2020. – 236 с.
3. Бенгфорт, Б. Прикладной анализ текстовых данных на Python. Машинное обучение и создание приложений обработки естественного языка / Б. Бенгфорт. – СПб.: Питер, 2019. – 368 с.
4. Пономарева Л.А., Чискидов С.В., Ронжина И.А., Голосов П.Е. Проектирование компьютерных обучающих систем: Монография. М-во образования и науки РФ, РАНХиГС, МГПУ ИЦО. Тамбов: Консалтинговая компания Юком, 2018. 120 с.
5. Прокофьева Е.Н. Оценка качества управления информационными потоками в организациях / Е.Н. Прокофьева, А.В. Вострикова // Вестник РМАТ, 2017. – 330 с.
6. Прохоренок Н.А. Python 3 и PyQt. Разработка приложений. – СПб.: БХВ-Петербург, 2012. – 704 с.
7. Самойлова И. А. Технологии обработки больших данных / Молодой ученый. - 2017. - № 49 (183). - С. 26-28.
8. Модели и методы исследования информационных систем: монография / А.Д. Хомоненко, А.Г. Басыров, В.П. Бубнов [и др.]. - Санкт-Петербург: Лань, 2019. - 204 с.
9. Канаев К.А., Фалеева Е.В., Пономарчук Ю.В. Сравнительный анализ форматов обмена данными, используемых в приложениях с клиент-серверной архитектурой // Фундаментальные исследования. – 2015. – № 2-25. – С. 5569-5572.

10. Златопольский Д.М. Основы программирования на языке Python. – М.: ДМК Пресс, 2017. – 284 с.
11. Виноградова Е.Ю. Интеллектуальные информационные технологии – теория и методология построения информационных систем: монография / М-во образования и науки РФ, Урал. гос. экон. ун-т. – Екатеринбург: Изд-во Урал. гос. экон. ун-та, 2011. – 263 с.
12. Белкова А. Л. Осваиваем работу с реляционными базами в MS Excel 2013 / А.Л. Белкова, С.Н. Леора // Теория и практика образования в современном мире: материалы VI Междунар. науч. конф. – Санкт-Петербург: Заневская площадь, 2014. – С. 349-356.
13. Уорсли, Дж. PostgreSQL. Для профессионалов / Дж. Уорсли, Дж. Дрейк. - М.: СПб: Питер, 2002. - 496 с.
14. Hans-Jürgen Schönig Mastering PostgreSQL 13 - Fourth Edition: Build, administer, and maintain database applications efficiently with PostgreSQL 13. - Packt Publishing, – 2020. - 476 p.
15. Baji Shaik, Avinash Vallarapu Beginning PostgreSQL on the Cloud: Simplifying Database as a Service on Cloud Platforms. – Apress, - 2018. - 381 p.

References

1. Loginova E.V. Necessity of studying information flows of an enterprise / E.V. Loginova, T.A. Sarieva // Problems of Modern Science and Education, 2017. - № 2. - *pp*. 45-48.
2. Methods and models of research of complex systems and big data processing: Monograph / I.Y. Paramonov, V.A. Smagin, N.E. Kosykh, A.D. Khomonenko; edited by V. A. Smagin and A. D. Khomonenko. - St. Petersburg: Lan', 2020. - 236 *p.*
3. Bengforth, B. Applied textual data analysis in Python. Machine learning and creating natural language processing applications / B. Bengforth. - St. Petersburg: Peter, 2019. - 368 *p.*
4. Ponomareva L.A., Chiskidov S.V., Ronzhina I.A., Golosov P.E. Designing computer learning systems: Monograph. Ministry of Education and Science of the Russian Federation, Russian Academy of National Econ-

omy and Public Administration, Moscow State Pedagogical University. Tambov: Consulting company Yukom, 2018. 120 *p.*

5. Prokofieva E.N. Assessment of the quality of information flow management in organizations / E.N. Prokof'eva, A.V. Vostrikova // Vestnik RMAT, 2017. - 330 *p.*
6. Prohorenok N.A. Python 3 and PyQt. Development of applications. - St. Petersburg: BHV-Peterburg, 2012. - 704 *p.*
7. Samoylova I. A. Technologies of big data processing / Young scientist. - 2017. - № 49 (183). - *pp*. 26-28.
8. Models and methods of research of information systems: monograph / A.D. Khomonenko, A.G. Basyrov, V.P. Bubnov [et al]. - Saint Petersburg: Lan', 2019. - 204 *p.*
9. Kanaev K.A., Faleeva E.V., Ponomarchuk Y.V. Comparative analysis of data exchange formats used in applications with client-server architecture // Fundamental Research. - 2015. - № 2-25. - *pp*. 5569-5572.
10. Zlatopolsky D.M. Fundamentals of programming in the Python language. - Moscow: DMK Press, 2017. - 284 *p.*
11. Vinogradova E. Yu. Intelligent information technology - theory and methodology of building information systems: monograph / Ministry of Education and Science of the Russian Federation, Ural State. Economics University. - Ekaterinburg: Publishing house of the Ural State University of Economics, 2011. - 263 *p.*
12. Belkova A.L. Mastering the work with relational databases in MS Excel 2013 / A.L. Belkova, S.N. Leora // Theory and practice of education in the modern world: proceedings of the VI International. scientific. conf. - St. Petersburg: Zanevskaya Square, 2014. - *pp*. 349-356.
13. Worsley, J. PostgreSQL. For professionals / J. Worsley, J. Drake. - M.: SPb: Peter, 2002. - 496 *p.*
14. Hans-Jürgen Schönig Mastering PostgreSQL 13 - Fourth Edition: Build, administer, and maintain database applications efficiently with PostgreSQL 13. - Packt Publishing, – 2020. - 476 p.
15. Baji Shaik, Avinash Vallarapu Beginning PostgreSQL on the Cloud: Simplifying Database as a Service on Cloud Platforms. – Apress, - 2018. - 381 p.

ВКЛАД АВТОРОВ

Крапивин Р.Р.: разработка программного обеспечения, тестирование существующих компонентов кода, обработка результатов исследований.

Гареева Г.А.: формулирование основных направлений исследования, разработка теоретических предпосылок, формирование общих выводов.

Филатов Ю.М.: проведение сбора данных, подготовка начального варианта статьи.

Файзуллина А.Г.: анализ и интерпретация полученных данных, литературный анализ.

Мышкина И.Ю.: научное редактирование текста статьи и окончательное утверждение версии для публикации.

AUTHOR CONTRIBUTIONS

Krapivin R.R.: software development, testing existing code components, processing research results.

Gareeva G.A.: formulation of the main directions of research, development of theoretical assumptions, formation of general conclusions.

Filatov Y.M.: Carrying out data collection, preparing the initial version of the article.

Fayzullina A.G.: analysis and interpretation of the data obtained, literary analysis.

Myshkina I.Yu.: scientific editing of the text of the article and final approval of the version for publication.

ДАННЫЕ ОБ АВТОРАХ

Крапивин Роман Русланович, студент

Казанский национальный исследовательский технический университет им. А.Н. Туполева-КАИ

ул. Академика Королева, 1, г. Набережные Челны, 423814, Российская Федерация

Jerichotyrant1@yandex.ru

Гареева Гульнара Альбертовна, заведующий кафедрой Информационных систем, канд. пед. наук, доцент
Казанский национальный исследовательский технический университет им. А.Н. Туполева-КАИ
ул. Академика Королева, 1, г. Набережные Челны, 423814, Российская Федерация
gagareeva1977@mail.ru

Филатов Юрий Михайлович, студент
Казанский национальный исследовательский технический университет им. А.Н. Туполева-КАИ
ул. Академика Королева, 1, г. Набережные Челны, 423814, Российская Федерация
Uraura111222@gmail.com

Файзуллина Айгуль Гинатулловна, преподаватель инженерно-экономического колледжа
Казанский федеральный университет, Набережночелнинский институт
проспект Мира, 68/19, г. Набережные Челны, 423812, Российская Федерация
dlya_pisem_t@mail.ru

Мышкина Ирина Юрьевна, доцент кафедры системного анализа и информатики, кандидат тех. наук, доцент
Казанский федеральный университет, Набережночелнинский институт
проспект Мира, 68/19, г. Набережные Челны, 423812, Российская Федерация
mirinau@mail.ru

DATA ABOUT THE AUTHORS

Roman R. Krapivin, student
Kazan National Research Technical University named after A.N. Tupolev-KAI

1, Akademika Koroleva Str., Naberezhnye Chelny, 423814, Russian Federation
Jerichotyrant1@yandex.ru

Yuri M. Filatov, student
Kazan National Research Technical University named after A.N. Tupolev-KAI
1, Akademika Koroleva Str., Naberezhnye Chelny, 423814, Russian Federation
Uraura111222@gmail.com

Gulnara A. Gareeva, Head of the Department of Information Systems, Candidate of Pedagogical Sciences, Associate Professor
Kazan National Research Technical University named after A.N. Tupolev-KAI
1, Akademika Koroleva Str., Naberezhnye Chelny, 423814, Russian Federation
gagareeva1977@mail.ru
Scopus Author ID: 36801593200
ResearcherID: M-1728-2015
SPIN-code: 3279-8465

Aigul G. Faizullina, Lecturer, College of Engineering and Economics
Kazan Federal University Naberezhnochelninsk Institute
68/19, Prospekt Mira, Naberezhnye Chelny 423812, Russian Federation
dlya_pisem_t@mail.ru

Irina Yu. Myshkina, Associate Professor, Department of System Analysis and Informatics
Kazan Federal University Naberezhnochelninsk Institute
68/19, Prospekt Mira, Naberezhnye Chelny 423812, Russian Federation
mirinau@mail.ru

Поступила 10.02.2023
После рецензирования 25.02.2023
Принята 09.03.2023

Received 10.02.2023
Revised 25.02.2023
Accepted 09.03.2023