

DOI: 10.12731/2227-930X-2023-13-4-188-204

УДК 004.67



Научная статья | Системный анализ, управление и обработка информации

## АВТОМАТИЗАЦИЯ ПРОЦЕССА СОСТАВЛЕНИЯ УЧЕБНОГО РАСПИСАНИЯ

*А.С. Петросян, А.А. Сурмачевская, Е.А. Сафонова,  
Э.М. Кузнецова, Г.А. Гареева*

*В данной статье представлена разработка, оптимизирующая процесс составления учебного расписания путем создания алгоритмов, которые автоматически распределяют нагрузку учебного процесса и учитывают все предпочтения преподавателей. Одной из популярных на сегодняшний день платформ для развертывания данных разработок является платформа для создания таблиц google Таблицы и создание кода на базе google app script. Особенностью системы, реализованной при помощи данной платформы, является простота использования и минимизация ресурсов, затраченных на хранение программного обеспечения, что в значительной степени упрощает взаимодействие с системой.*

***Цель** – создание инструмента для составления учебного расписания, взаимодействия с базой данных созданной в Google Таблицы, взаимодействия с участниками учебного заведения в режиме онлайн, что экономит личное время преподавателей и работников деканата.*

***Метод или методология проведения работы:** в статье рассматривается способ, который оптимизирует процесс составления расписания на неделю, учитывая удобства преподавателя в распределении учебных занятий и нагрузку учебного процесса путем создания алгоритмов на базе google app script. Для реализации используются онлайн-таблицы, доступные всем преподавателям*

и работникам деканата, обновляемые в режиме реального времени, созданные в Google Таблицы и при помощи языка программирования *google app script*.

**Результат:** разработан уникальный инструмент, который реализует функции добавления, хранения, взаимодействия и круглосуточного доступа к данным учебного расписания.

**Область применения результатов:** учебное расписание, хранящееся в онлайн-таблицах и обновляемое работниками деканата, которое целесообразно использовать для мониторинга каждому преподавателю, следя за своим временем проведения учебных занятий в образовательных учреждениях.

**Ключевые слова:** *google app script*; учебное расписание; учебный процесс; автоматизация; онлайн-таблицы; программирование; Google Таблицы

**Для цитирования.** Петросян А.С., Сурмачевская А.А., Сафонова Е.А., Кузнецова Э.М., Гареева Г.А. Автоматизация процесса составления учебного расписания // *International Journal of Advanced Studies*. 2023. Т. 13, № 4. С. 188-204. DOI: 10.12731/2227-930X-2023-13-4-188-204

Original article | System Analysis, Management and Information Processing

## **AUTOMATING THE PROCESS OF PREPARING THE SCHOOL TIMETABLE**

***A.S. Petrosyan, A.A. Surmachevskaya, E.A. Safonova,  
E.M. Kuznetsova, G.A. Gareeva***

*This paper presents a development that optimizes the process of educational scheduling by creating algorithms that automatically distribute the load of the educational process and take into account all the preferences of teachers. One of the currently popular platforms for deploying these developments is the google tables platform for creat-*

*ing google tables and creating code based on google app script. The peculiarity of the system implemented with the help of this platform is the ease of use and minimization of resources spent on software storage, which greatly simplifies interaction with the system.*

**Purpose** – *create a tool for drawing up a study schedule, interacting with a database created in Google Tables, interacting with participants of an educational institution online, which saves personal time of teachers and employees of the dean's office.*

**Method or methodology of the work:** *the article discusses a way to optimize the creation of a weekly schedule, taking into account the convenience of the teacher in the distribution of pairs and taking into account the load of the educational process by creating algorithms based on Google app script. For implementation, online tables are used that are accessible to all teachers and employees of the dean's office, updated in real time, Tables created in Google and the google app script programming language.*

**Result:** *a unique tool has been developed that implements the functions of adding, storing, interacting and round-the-clock access to training schedule data.*

**Scope of application of the results:** *the curriculum, stored in online tables and updated by the dean's office staff, should be used for monitoring by each teacher, keeping track of their time of couples in educational institutions.*

**Keywords:** *google app script; learning schedule; learning process; automation; online-tables; programming; Google Tables*

**For citation.** *Petrosyan A.S., Surmachevskaya A.A., Safonova E.A., Kuznetsova E.M., Gareeva G.A. Automating the Process of Preparing the School Timetable. International Journal of Advanced Studies, 2023, vol. 13, no. 4, pp. 188-204. DOI: 10.12731/2227-930X-2023-13-4-188-204*

Составление учебного расписания является трудоёмким процессом для работников деканата, требующим большого количества времени, непосредственной связи с преподавателями для

уточнения удобства распределения учебных занятий и распределения нагрузки на рабочий день преподавателя. В основном каждое образовательное учреждение создает учебное расписание занятий на месяц вперед без учета изменений и гибкости графика. Оно необходимо для слаженной работы всего учебного процесса учреждения. На данный момент процедура создания учебного расписания является неслаженным и трудоемким процессом как для преподавателей, так и для работников деканата. Данный процесс можно оптимизировать благодаря онлайн-таблицам, связанным алгоритмом для оптимального распределения нагрузки и гибкости графика преподавателей, что облегчит задачу работникам учебного заведения.

Каждому преподавателю создается отдельная онлайн-таблица с расписанием на неделю, которое преподаватель заполняет самостоятельно в соответствии со своими предпочтениями. Данные таблицы были разработаны с ограничениями по количеству занятий в неделю для равномерного распределения учебного процесса на весь семестр.

Целью данного исследования является разработка программного продукта для добавления, хранения и взаимодействия с расписанием учебных занятий по каждой дисциплине непосредственно самим преподавателем, а также проверки актуального расписания.

Для достижения поставленной цели потребуются следующие ресурсы:

1. Базовая информация о дисциплине для определенной группы:
  - список групп
  - список преподавателей
  - перечень дисциплин, преподаваемых на каждом курсе, каждой группы
2. В качестве разработки проекта потребуются:
  - создание шаблонных таблиц для преподавателей и итоговой таблицы с расписанием в сервисе для создания электронных таблиц – «Гугл Таблицы» (англ. Google Sheets)

- создание алгоритма распределения занятий с помощью облачного скриптового языка Google Apps Script
3. Готовый проект состоит из следующих элементов:
- Google Таблица с фамилией и дисциплинами преподавателя
  - индивидуальная Google Таблица преподавателя для составления расписания
  - итоговая Google Таблица с оптимальным распределением предметов на неделю обучения, которая строится на основании индивидуальных таблиц преподавателей
  - алгоритм просчета нагрузки преподавателя на неделю
  - алгоритм распределения приоритетов того или иного предмета в случае выбора занятий на одно и то же время разных преподавателей.

Для создания алгоритмов был выбран язык программирования google app script, который является встроенным языком для расширения функциональности Google Apps и создания легких облачных приложений. Данный язык был выбран на основании того, что таблицы, обновляемые в режиме реального времени и предоставляющие доступ для всех преподавателей с разных устройств, созданы в Google Таблицах. Соответственно в результате данного действия были минимизированы потери времени на запросы и увеличена оптимизация проекта за счет того, что он работает на одной платформе.

На рисунке 1 представлено создание таблицы расписания с выпадающими списками и привязкой к почте каждого преподавателя.

Далее будет представлен алгоритм взаимодействия. Для каждого преподавателя будет создана индивидуальная таблица расписания на неделю, в которой они заполняют предметы, которые преподают в той или иной группе. При заполнении таблицы подтверждение выбранной даты и времени занятий производится кнопкой «внести изменения в ячейку» (рис.2). Данная опция была создана для исключения ошибочного выбора и корректной работы таблицы.

Рис. 1. Создание таблицы расписания

A1						
	A	B	C	D	E	F
1	<div style="background-color: #c8e6c9; padding: 5px; text-align: center;"> <b>ВНЕСТИ ИЗМЕНЕНИЯ В ЯЧЕЙКУ</b> </div>		Дисциплина	Ауд.	Дисциплина	Ауд.
2			15.03.05 Конструкторско-технологическое обеспечение машиностроительных производств			
3	Де нед	Направление				
4		Группа	23101		23102	
5	<div style="writing-mode: vertical-rl; transform: rotate(180deg);"> <b>понедельник 27.02.2023</b> </div>	8:00-9:30				
6		9:40-11:10				
7		11:30-13:00				
8		13:10-14:40				
9		15:00-16:30				
10		16:40-18:10				

Рис. 2. Кнопка подтверждения выбора времени и даты

Для оптимизации и разгрузки учебного процесса на одну неделю будет выделяться определенное количество занятий для каждого предмета с расчетом времени нагрузки на обучающихся и согласно нормированному 8 часовому рабочему графику преподавателей. Преподаватель заполняет расписание с определённым предметом в таблице с выбором времени и дня недели, которые

являются для него предпочтительными. После заполнения преподавателями всех предметов, все таблицы будут объединены в одну общую таблицу. Зеленая ячейка означает, что на эту дату и время были заполнены ячейки у нескольких преподавателей. На рисунке 3 продемонстрирована общая таблицы расписания.

The image shows a screenshot of a software application displaying a large, multi-column table. The table is organized into several vertical sections, each representing a different subject or course. The rows represent time slots. Green rectangular highlights are placed in various cells across the table, indicating that these specific subject-time slot combinations have been assigned to multiple teachers. The interface includes a menu bar at the top and a status bar at the bottom.

**Рис. 3.** Объединённая таблица расписания

Для удобного и своевременного создания расписания было принято решение о заполнении таблиц в субботу. Данное действие будет реализовывать уже сформированное расписание к концу этого дня для публикации, распечатки и ознакомления его с преподавательским составом.

Выполнение алгоритма просчета приоритетов и нагрузки на рабочий день будет следующим. При пересечении двух или более занятий на одну и ту же дату приоритет будет отдаваться по расчету удовлетворенности желаемого времени на данной неделе. На данном этапе выполняется сравнение и просчет всех таблиц, что продемонстрировано на рисунке 4.

На следующем этапе кода выполняется функция распределения приоритетов. При одинаковом количестве процентов удовлетворенности между двумя или более предметами программа выполняет еще один цикл просчета, где уже будет учитывать нагрузку на преподавателя на день пересечения предметов с одинаковым процентом удовлетворенности времени. Соответственно, если нагрузка на преподавателя выше, то приоритет будет отдан преподавателю, чей процент загруженности на этот день будет меньше. Данные действия продемонстрированы на рисунках 5 и 6.

```

120 //функция сравнивает два массива и определяет количество совпадений
121 function matchedCellsCount(firstArray, secondArray) {
122     let matchedValFromArray2 = [];
123     for (var line in firstArray) {
124         var isMatched = secondArray.indexOf(firstArray[line]);
125         if (isMatched !== -1) {
126             //var matchedValFromArray2 = secondArray[isMatched]
127             matchedValFromArray2.push(secondArray[isMatched]);
128         }
129     }
130     return matchedValFromArray2.length;
131     //console.log(matchedValFromArray2);
132 }

```

Рис. 4. Сравнение таблиц

```

//функция считает процент удовлетворённости
function percentOfSatisfaction(array, countOfMatch, countOfRed) {
    var chislitel = countOfMatch + countOfRed;
    var percentOfSatisfaction = 100 - (chislitel / array.length * 100);
    return percentOfSatisfaction;
}

```

Рис. 5. Просчет процента удовлетворенности

```

//сравнение процентов удовлетворенности
function comparePosForColor(pos1, id1, pos2, id2, matchedCells) {
    if (pos1 > pos2) {
        colorRedMC(matchedCells, id1); //вызывает функцию, которая окрасит совпавшие ячейки в таблице учителя, чей уровень удовлетворённости выше
        //console.log(matchedCells);
    }
    else if (pos1 < pos2) {
        colorRedMC(matchedCells, id2);
        //console.log(matchedCells);
    }
    else {
        console.log("Одинаковый процент удовлетворённости");
    }
}

```

Рис. 6. Сравнение процентов удовлетворенности

Следующим выполнением кода является создание готового расписания, где ячейки, окрашенные зеленым и белым цветом, остаются неизменными и переносятся в итоговую таблицу, а ячейки, окрашенные красным цветом и имеющие более низкий приоритет, будут перенесены на ближайшие свободные ячейки от тех дат, где они были изначально. Демонстрация выполнения данного кода показана на рисунках 7 и 8.



```
//копирует из таблицы все ячейки, которые содержат значения и не красные и вставляет в итоговую таблицу
function copyPaste(id) {
    var finalSheet = SpreadsheetApp.openById("1vOor82fKhGzh49xTettTxPTBtvBrZpCtQnsPuwvVPTY");
    var finalRange = finalSheet.getSheetByName("Schedule");
    var pr1 = SpreadsheetApp.openById(id);
    var range = pr1.getRange("Schedule!B4:AH40");
    var numRows = range.getNumRows();
    var numCols = range.getNumColumns();

    for (var i = 1; i < numRows; i++) {
        for (var j = 1; j < numCols; j++) {
            if (range.getValues()[i][j] != '' && range.getBackgrounds()[i][j] != '#ff0000') {
                let c11 = range.getValues()[i][j];
                finalRange.getRange(i + 4, j + 2).setValue(c11);
            }
        }
    }
}
}
```

**Рис. 7.** Перенос в итоговую таблицу расписания занятий с не пересекающимися с другими предметами и ячейки с более высокими приоритетами

```
//ВЫЗОВ ФУНКЦИИ КОТОРАЯ ПЕРЕНЕСЕТ ЭТУ КРАСНУЮ ЯЧЕЙКУ В ИТОГОВУЮ ТАБЛИЦУ И ПОКРАСИТ В ЗЕЛЕНый
function copyPasteRedCell(finalList, range2List, row, col, emptyCellRow) {

    var copyCell = range2List.getRange(row, col).getValue(); //значение красной ячейки в таблице преподавателя
    range2List.getRange(row, col).setBackgroundcolor('green'); //перекраска этой яч-ки в зеленый
    var pasteRange = finalList.getRange(emptyCellRow, col); //ячейка в итоговой табл в которую будет переносится значение
    pasteRange.setValue(copyCell); //вставка значения
    pasteRange.setBackgroundcolor('green'); //зеленый цвет для этой яч-ки в итоговой табл

    console.log("адрес красной ячейки");
    console.log(row, col);
    console.log("Значение");
    console.log(range2List.getRange(row, col).getValue());
    console.log("адрес места для вставки");
    console.log(emptyCellRow, col);
}
}
```

**Рис. 8.** Перенос предметов с более низким приоритетом на ближайшие даты от того времени, на которое они были назначены

```
//узнать в первой или второй смене красная ячейка
function getShiftOfRedCell(row, col, lastName, range2List) {
    var arrFirsShift = [5, 6, 7, 11, 12, 13, 17, 18, 19, 23, 24, 25, 29, 30, 31, 35, 36, 37]; //строки первой смены
    var arrSecondShift = [8, 9, 10, 14, 15, 16, 20, 21, 22, 26, 27, 28, 32, 33, 34, 38, 39, 40]; //строки второй смены

    if (arrFirsShift.includes(row)) {
        console.log("Первая смена");
        //ВЫЗОВ ФУНКЦИИ ПОИСКА СВОБОДНЫХ ЯЧЕЕК ЭТОЙ ГРУППЫ В ИХ СМЕНУ
        getEmptyCellsInShift(row, col, arrFirsShift, lastName, range2List);
    }
    else {
        console.log("Вторая смена");
        //ВЫЗОВ ФУНКЦИИ ПОИСКА СВОБОДНЫХ ЯЧЕЕК ЭТОЙ ГРУППЫ В ИХ СМЕНУ
        getEmptyCellsInShift(row, col, arrSecondShift, lastName, range2List);
    }
}
}
```

**Рис. 9.** Разделение расписания на первую и вторую смену обучения

Для распределения умного расписания был написан блок кода, делящий обучение на 1 и 2 смену соответственно. Данная функция не позволяет переносить занятия, установленные в определенный день первой смены на эту же дату второй смены.

Выполнение данного блока кода показано на рисунке 9.

Для наглядности алгоритма действий была создана схема взаимодействия всех участвующих в процессе реализации проекта звеньев. Схема продемонстрирована на рисунке 10.



Рис. 10. Процесс создания умного расписания

На финальной стадии создания разработки была произведена контрольная проверка работоспособности. Все технические аспекты исправно работают.

Также были проведены тесты для опроса на удобство использования данной разработки на практике. Данная разработка является многозадачной и может использоваться в разных образовательных учреждениях. Возможна модернизация данного расписания и упрощения функций просчетов нагрузки для оптимизации рабочего времени работников деканата.

### *Список литературы*

1. Тугов В.В. Проектирование автоматизированных систем управления: учебное пособие / В.В. Тугов, А.И. Сергеев, Н.С. Шаров. Санкт-Петербург: Лань, 2019. 172 с.
2. Модели и методы исследования информационных систем: монография / А.Д. Хомоненко, А.Г. Басыров, В.П. Бубнов [и др.]. Санкт-Петербург: Лань, 2019. 204 с.
3. Златопольский Д.М. Основы программирования на языке Python. М.: ДМК Пресс, 2017. 284 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. 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.
6. Козлов А. А. Телеграм-бот как простой и удобный способ получения информации / А. А. Козлов, А. В. Батищев // Территория науки. 2017. №5. С. 55-64.
7. Аванесян Н. Л. Telegram как пример мессенджера: возможности и перспективы развития // Научный потенциал XXI века: матери-

- алы Международной (заочной) научно-практической конференции. Астана, 2017. С. 61-65.
8. Иванов А. Д. Чат-бот в Telegram и ВКонтакте как новый канал распространения новостей // Вестник Волжского университета им. В.Н. Татищева. 2016. Т. 1, № 3. С. 126-132.
  9. Насибулин Р.О. Создание простого бота модератора для мессенджера Telegram с помощью языка программирования Python / Р.О. Насибулин, Г.А. Гареева // Молодежь и системная модернизация страны: Сборник научных статей 7-й Международной научной конференции студентов и молодых ученых. В 5-ти томах, Курск, 19–20 мая 2022 года / Отв. редактор М.С. Разумов. Том 3. Курск: Юго-Западный государственный университет, 2022. С. 425-427.
  10. Гареева Г.А. Разработка HR-бота для автоматизации процесса подбора персонала в производственном предприятии / Г.А. Гареева, И.М. Сафонов, З. Г. Джигладзе [и др.] // Научно-технический вестник Поволжья. 2023. № 1. С. 52-55.
  11. Alex Root Jr. Aiogram Documentation, November 2022. <https://readthedocs.org/projects/aiogram/downloads/pdf/latest/>
  12. David Love. Tkinter GUI Programming by Example. Packt Publishing, 2018. 340 p.
  13. Baji Shaik, Avinash Vallarapu Beginning PostgreSQL on the Cloud: Simplifying Database as a Service on Cloud Platforms. Apress, 2018. 381 p.
  14. Чат-боты: история, современность и перспективы. <http://w7phone.ru/chat-boty-istoriya-sovremennost-i-perspektivy-132460>
  15. Telegram Bot API. <https://core.telegram.org/bots/api>

### *References*

1. Tugov V.V. *Proektirovanie avtomatizirovannykh sistem upravleniya* [Designing automated control systems] / V.V. Tugov, A.I. Sergeev, N.S. Sharov. St. Petersburg: Lan, 2019, 172 p.
2. *Modeli i metody issledovaniya informatsionnykh sistem* [Models and methods of information systems research] / A.D. Khomonenko, A.G. Basyrov, V.P. Bubnov [et al.]. St. Petersburg: Lan, 2019, 204 p.

3. Zlatopol'skiy D.M. *Osnovy programmirovaniya na yazyke Python* [Fundamentals of programming in the Python language]. M.: DMK Press, 2017, 284 p.
4. Ponomareva L.A., Chiskidov S.V., Ronzhina I.A., Golosov P.E. *Proektirovanie komp'yuternykh obuchayushchikh sistem* [Designing computer-based learning systems]. Tambov: Consulting company Yukrom, 2018, 120 p.
5. Vinogradova E.Yu. *Intellektual'nye informatsionnye tekhnologii – teoriya i metodologiya postroeniya informatsionnykh sistem* [Intellectual information technologies - theory and methodology of building information systems]. Yekaterinburg: Ural State University of Economics, 2011, 263 p.
6. Kozlov A. A., Batishchev A. V. *Territoriya nauki*, 2017, no. 5, pp. 55-64.
7. Avanesyan N. L. *Nauchnyy potentsial XXI veka: materialy Mezhdunarodnoy (zaочноy) nauchno-prakticheskoy konferentsii* [Scientific potential of the XXI century: materials of the International (extramural) scientific-practical conference]. Astana, 2017, pp. 61-65.
8. Ivanov A. D. *Vestnik Volzhskogo universiteta im. V.N. Tatishcheva*, 2016, vol. 1, no. 3, pp. 126-132.
9. Nasibulin R.O., Gareeva G.A. *Molodezh' i sistemnaya modernizatsiya strany: Sbornik nauchnykh statey 7-y Mezhdunarodnoy nauchnoy konferentsii studentov i molodykh uchenykh. V 5-ti tomakh, Kursk, 19–20 maya 2022 goda* [Youth and system modernization of the country: Collection of scientific articles of the 7th International Scientific Conference of Students and Young Scientists. In 5 volumes, Kursk, May 19-20, 2022] / ed. M.S. Razumov. Vol. 3. Kursk: South-West State University, 2022, pp. 425-427.
10. Gareeva G.A., Safonov I.M., Dzhibladze Z.G. [et al.] *Nauchno-tehnicheskii vestnik Povolzh'ya*, 2023, no. 1, pp. 52-55.
11. Alex Root Jr. Aiogram Documentation, November 2022. <https://readthedocs.org/projects/aiogram/downloads/pdf/latest/>

12. David Love. Tkinter GUI Programming by Example. Packt Publishing, 2018, 340 p.
13. Baji Shaik, Avinash Vallarapu Beginning PostgreSQL on the Cloud: Simplifying Database as a Service on Cloud Platforms. Apress, 2018, 381 p.
14. Chatbots: history, modernity and prospects. <http://w7phone.ru/chat-boty-istoriya-sovremennost-i-perspektivy-132460>
15. Telegram Bot API. <https://core.telegram.org/bots/api>

### **ВКЛАД АВТОРОВ**

**Петросян А.С.:** разработка программного продукта, обработка результатов исследований.

**Сурмачевская А.А.:** формулирование основных направлений исследования, разработка теоретических предпосылок, формирование общих выводов.

**Сафонова Е.А.:** проведение сбора данных, подготовка начального варианта статьи.

**Кузнецова Э.М.:** анализ и интерпретация полученных данных, литературный анализ.

**Гареева Г.А.:** научное редактирование текста статьи и окончательное утверждение версии для публикации.

### **AUTHOR CONTRIBUTIONS**

**Ashot S. Petrosyan:** development of the software product, processing of research results.

**Angelica A. Surmachevskaya:** formulation of the main directions of research, development of theoretical assumptions, formation of general conclusions.

**Ekaterina A. Safonova:** carrying out data collection, preparation of the initial version of the article.

**Evelina M. Kuznetsova:** analysis and interpretation of the data obtained, literary analysis.

**Gulnara A. Gareeva:** scientific editing of the text of the article and final approval of the version for publication.

## **ДАННЫЕ ОБ АВТОРАХ**

**Петросян Ашот Сергеевич**, студент

*Казанский национальный исследовательский технический университет им. А.Н. Туполева-КАИ  
ул. Академика Королева, 1, г. Набережные Челны, 423814,  
Российская Федерация  
ashot\_116@mail.ru*

**Сурмачевская Анжелика Александровна**, студент

*Казанский национальный исследовательский технический университет им. А.Н. Туполева-КАИ  
ул. Академика Королева, 1, г. Набережные Челны, 423814,  
Российская Федерация  
2db6db@mail.ru*

**Сафонова Екатерина Алексеевна**, студент

*Казанский национальный исследовательский технический университет им. А.Н. Туполева-КАИ  
ул. Академика Королева, 1, г. Набережные Челны, 423814,  
Российская Федерация  
sea55555@bk.ru*

**Кузнецова Эвелина Максимовна**, студент

*Казанский национальный исследовательский технический университет им. А.Н. Туполева-КАИ  
ул. Академика Королева, 1, г. Набережные Челны, 423814,  
Российская Федерация  
270303ava@gmail.com*

**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*

#### **DATA ABOUT THE AUTHORS**

**Ashot S. Petrosyan**, student

*Kazan National Research Technical University named after  
A.N. Tupolev-KAI  
1, Akademika Koroleva Str., Naberezhnye Chelny, 423814, Rus-  
sian Federation  
ashot\_116@mail.ru*

**Angelica A. Surmachevskaya**, student

*Kazan National Research Technical University named after  
A.N. Tupolev-KAI  
1, Akademika Koroleva Str., Naberezhnye Chelny, 423814, Rus-  
sian Federation  
2db6db@mail.ru*

**Ekaterina A. Safonova**, student

*Kazan National Research Technical University named after  
A.N. Tupolev-KAI  
1, Akademika Koroleva Str., Naberezhnye Chelny, 423814, Rus-  
sian Federation  
sea55555@bk.ru*

**Evelina M. Kuznetsova**, student

*Kazan National Research Technical University named after  
A.N. Tupolev-KAI  
1, Akademika Koroleva Str., Naberezhnye Chelny, 423814, Rus-  
sian Federation  
270303ava@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*  
*SPIN-code: 3279-8465*  
*Scopus Author ID: 36801593200*  
*ResearcherID: M-1728-2015*  
*ORCID: <https://orcid.org/0000-0002-8539-4541>*

Поступила 13.11.2023  
После рецензирования 28.11.2023  
Принята 02.12.2023

Received 13.11.2023  
Revised 28.11.2023  
Accepted 02.12.2023