

DOI: 10.12731/2227-930X-2023-13-3-163-179
УДК 004.67



Научная статья | Системный анализ, управление и обработка информации

АВТОМАТИЧЕСКОЕ УВЕДОМЛЕНИЕ ОБ АКАДЕМИЧЕСКИХ ЗАДОЛЖЕННОСТЯХ НА ОСНОВЕ TELEGRAM

*А.С. Петросян, А.А. Сурмачевская, А.А. Новичков,
Д.Г. Лукин, Г.А. Гареева*

В данной статье рассмотрен процесс оптимизации времени и удобства сдачи академических задолженностей путем создания Telegram бота, который автоматически присылает обучающимся уведомления о задолженностях и указывает день, когда их можно сдать. Одной из популярных на сегодняшний день платформ для развертывания ботов является мессенджер Telegram. Особенностью системы, реализованной при помощи бота, является простота использования и минимизация ресурсов, затраченных на хранение программного обеспечения, что в значительной степени упрощает взаимодействие с системой.

***Цель** – создание инструмента для сдачи академических задолженностей, взаимодействия с базой данных созданной в Google Таблицы, взаимодействия с участниками учебного процесса в режиме онлайн, что экономит личное время обучающихся и преподавателей.*

***Метод или методология проведения работы:** в статье рассматривается способ, который реализует сдачу академических задолженностей путем создания Telegram бота. Для реализации используется база данных доступная всем преподавателям, обновляемая в режиме реального времени, созданная в Google Таблицы и язык программирования C#.*

Результат: разработан уникальный инструмент, который реализует функции добавления, хранения, взаимодействия и круглосуточного доступа к данным об академических задолженностях.

Область применения результатов: данные о задолженностях, хранящиеся в базе данных и обновляемые преподавателями, целесообразно использовать для мониторинга каждому обучающемуся о своих задолженностях в образовательных учреждениях.

Ключевые слова: google app script; академические задолженности; Telegram бот; автоматизация; база данных; программирование; Google Таблицы

Для цитирования. Петросян А.С., Сурмачевская А.А., Новичков А.А., Лукин Д.Г., Гареева Г.А. Автоматическое уведомление об академических задолженностях на основе Telegram // International Journal of Advanced Studies. 2023. Т. 13, № 3. С. 163-179. DOI: 10.12731/2227-930X-2023-13-3-163-179

Original article | System Analysis, Management and Information Processing

AUTOMATIC NOTIFICATION OF ACADEMIC DEBTS BASED ON TELEGRAM

***A.S. Petrosyan, A.A. Surmachevskaya, A.A. Novichkov,
D.G. Lukin, G.A. Gareeva***

This article considers the process of optimizing the time and convenience of taking academic debts by creating a Telegram bot that automatically sends students notifications about debts and indicates the day when they can be passed. One of today's popular platforms for deploying bots is the Telegram messenger. The peculiarity of a system implemented with the help of a bot is its ease of use and minimization of resources spent on software storage, which greatly simplifies interaction with the system.

Purpose – create a tool for passing academic debts, interaction with the database created in Google Table, interaction with the par-

participants of the educational process online, which saves personal time of students and teachers.

Method or methodology of the work: the article deals with the method which implements the delivery of academic debts by creating Telegram bot. To implement it we use a database available to all teachers, updated in real time, created in Google Tables and the C# programming language.

Result: developed a unique tool that implements functions of adding, storing, interacting and 24/7 access to academic debts data.

Scope of the results: data on debts stored in the database and updated by teachers is useful for monitoring each student about his debts in educational institutions.

Keywords: google app script; academic debts; Telegram bot; automation; database; programming; Google Tables

For citation. Petrosyan A.S., Surmachevskaya A.A., Novichkov A.A., Lukin D.G., Gareeva G.A. Automatic Notification of Academic Debts Based on Telegram. *International Journal of Advanced Studies*, 2023, vol. 13, no. 3, pp. 163-179. DOI: 10.12731/2227-930X-2023-13-3-163-179

Введение

Сдача академических задолженностей является трудоемким процессом как для преподавателей так и для учащихся, требующим большого объема затрат времени, физического присутствия для получения информации о месте, времени, номера аудитории, предмета и темы задолженности, в то же время его необходимость и важность трудно переоценить. Каждое образовательное учреждение имеет определенное количество обучающихся, имеющих академические задолженности, их число значительно увеличивается, если время подходит к сдаче сессии.

Процедура сдачи академических задолженностей является обязательным процессом перед началом сессии. Она необходима для допуска студента к сдаче экзаменов. Процедура сдачи акаде-

мических задолженностей является неслаженным и трудоемким процессом, как для преподавателей, так и для обучающихся.

Для снижения затрат времени и труда целесообразно создать базу данных с записью обучающихся и имеющих академические задолженности. В случае его упрощения будет создана база данных с возможностью редактирования места, времени, номера аудиторной и актуальности данной задолженности на текущий период. Также для удобного заполнения и сокращения времени заполнения базы данных преподавателями, будут автоматически предлагаться уже по умолчанию сохранённые в базе номера обучающихся групп, имена и фамилии обучающихся имеющие академические задолженности, так же будет прикреплено автозаполнение номера телефона обучающихся, имеющие академические задолженности, который будет служить индивидуальным ID в базе данных. Так же по номеру телефона будет производиться выполнение алгоритма поиска задолженностей Telegram ботом в базе данных.

Цель работы – разработать программный продукт для добавления, хранения и взаимодействия с данными академических задолженностей, а также возможности проверить актуальную на сегодняшний день информацию о наличии задолженностей и даты сдачи академического долга.

Для достижения поставленной цели потребуются следующие ресурсы:

1) Базовая информация об обучающихся в данном образовательном учреждении:

- список обучающихся студентов в институте;
- список групп;
- номера контактных телефонов;
- перечень предметов, преподаваемых в данном образовательном учреждении.

Из данных, приведенных выше, будет создан массив, который будет отвечать запросам телеграмм бота об академических задолженностях.

2) В качестве разработки проекта потребуются:

- Создание базы данных в *Google Таблицах*
- Создание Telegram Бота с помощью google app script
- Создание запросов для обратной связи с пользователем Telegram Бота

3) Готовый проект состоит из следующих элементов:

- Массив в базе данных с информацией о каждом обучающемся в образовательном учреждении
- Telegram Бот с функцией обратной связи
- скрипта для обновления базы данных

Для создания Telegram бота был выбран язык программирования google app script. Данный язык был выбран на основании того что база данных, обновляемая в режиме реального времени и имеющая доступ для всех преподавателей с разных устройств посредством предоставления доступа, создана в Google Таблицах. Соответственно в результате данного действия были минимизированы потери времени на запросы и увеличена оптимизация проекта за счет того что она работает на одной платформе. На рисунке 1 показан код для создания Telegram бота.

```

1 |
2 | var token = "6102104957:AAEBLj43LSJaP3y00F9KbRsnqOu0iJ72eWU";
3 | var telegramUrl = "https://api.telegram.org/bot" + token;
4 | var webAppUrl = "https://script.google.com/macros/g/as/fycbirtAgmXSkd-mUu-7RbFEwfk50w7a1PvZmMuc6LFK3RmHU3a13wcTumd1@A6rdnMw/exec";
5 | var spreadsheetId = "1jBw50ZakdE13Lz398o4ZP0jKj1PRXA908K-HvB_T40";
6 |
7 | function setWebhook() {
8 |   var url = telegramUrl + "/setWebhook?url=" + webAppUrl;
9 |   UrIFetchApp.fetch(url);
10 | }
11 |
12 | function sendText(chatId, text, keyboard) {
13 |   var data = {
14 |     method: "post",
15 |     payload: {
16 |       method: "sendMessage",
17 |       chat_id: String(chatId),
18 |       text: text,
19 |       parse_mode: "HTML",
20 |       reply_markup: JSON.stringify(keyboard)
21 |     }
22 |   };
23 |   UrIFetchApp.fetch(`${telegramUrl}/bot${token}/${}`, data);
24 | }
25 |

```

Рис. 1. Код Telegram бота

Алгоритм взаимодействия с данным проектом будет выглядеть следующим образом. Преподаватель, которому предоставлен до-

ступ к базе данных, вносит данные об академических задолженностях определенного студента, при выборе фамилии студента будет включен скрипт автозаполнения сразу всех столбцов, связанных с данными студента. Данный скрипт помогает преподавателю сэкономить время и исключить ошибки при запросах Telegram бота в базу данных. Преподавателю остается заполнить 5 полей:

- 1) дата явки для сдачи академической задолженности;
- 2) время явки для сдачи академической задолженности;
- 3) номер аудитории;
- 4) информацию о задолженностях, которые студент должен сдать;
- 5) название предмета.

На рисунке 2 показан скрипт для автозаполнения форм в базе данных.

```

1 function onEditLists() {
2   let ss = SpreadsheetApp.getActiveSpreadsheet();
3   var TablicaSheet = ss.getSheetByName("Таблица");
4   var StudentsSheet = ss.getSheetByName("Студенты");
5
6   let ar = TablicaSheet.getActiveCell();
7   //console.log(ar.getColumn());
8   let col = 0;
9
10  if (ar.getColumn() == 2) { //если активная ячейка в столбце с группами, то выполняем скрипт
11    let valToFind = ar.getValue();
12    ar.offset(0,1).clearContent().clearDataValidations(); //очищает ячейку справа от активной
13    let values = StudentsSheet.getRange(1,1,StudentsSheet.getLastColumn()).getValues(); //получает первую строку таблицы студентов
14
15    col = values[0].indexOf(valToFind)+1; //ищет колонку с номером группы в таблице студентов
16
17    let fioValues = StudentsSheet.getRange(3,col,48,1); //диапазон с ФИО
18
19    let validation = SpreadsheetApp.newDataValidation().requireValueInRange(fioValues).build(); //создает диапазон с ФИО
20    //ar.offset(0,1).setDataValidation(validation); //вставляет диапазон с ФИО в правую ячейку от активной
21    TablicaSheet.getRange(ar.getRow(),3).setDataValidation(validation);
22  }
23
24  if (ar.getColumn() == 3) { //если выбрали фамилию студента
25    let valToFind = ar.getValue();
26    let stroka = ar.getRow();
27    let colonka = ar.getColumn();
28    let numberOfGroupToFind = TablicaSheet.getRange(stroka, colonka-1).getValue(); //получаем номер группы для поиска
29    let values = StudentsSheet.getRange(1,1,1,StudentsSheet.getLastColumn()).getValues();
30    col = values[0].indexOf(numberOfGroupToFind)+1; //колонка с номером группы
31
32    for (var i = 3; i <= StudentsSheet.getLastRow(); i++){
33      if(valToFind == StudentsSheet.getRange(1,col).getValue()){ //поиск фамилии студента в колонке на листе студентов
34        let nomer = StudentsSheet.getRange(i,col+1).getValue(); //берет номер в правой ячейке от фамилии
35        TablicaSheet.getRange(stroka, colonka+1).setValue(nomer); //заполняет номер студента в таблицу долгов справа от фамилии
36      }
37    }
38  }
39
40 }
41

```

Рис. 2. Автозаполнение форм в базе данных

После заполнения базы данных информация автоматически сохраняется. Информация сохраняется, пока преподаватель не снимет отметку о сдаче данной академической задолженности.

Код функции удаления академических задолженностей показан на рисунке 3.

```

1 function currentPosition(){
2   var spreadsheetId = "1JBwS0ZakdE1SL123G8oAZP0jkjPRXA908K-Hvb_T40";
3   var DataSheet = SpreadsheetApp.openById(spreadsheetId).getSheetByName("Таблица"); //берёт лист с таблицей
4
5   for(var i = DataSheet.getLastRow(); i>=1; i--) { //цикл начинается с последней строки и до строки заголовка
6     if (DataSheet.getRange(1,9).getValue() != true){ //если флажок в состоянии true
7       DataSheet.deleteRow(i); //удалить строку
8     }
9   }
10 }

```

Рис. 3. Функция удаления строк академических задолженностей

На рисунке 4 показана база данных, в которой хранится массив с информацией о студентах и их задолженностях.

	A	B	C	D	E	F	G	H	I	J	K	L
1	Предмет	Группа	ФИО студента	Номер телефона студента	Дата явки	Время явки	Задолженности	Кабинет	Сдал			
2	Законодатель...	Гр_23203	Куряцова Звельона	7953	01.04.2023	10:00	Долгид	102	<input type="checkbox"/>			
3	Информатика	Гр_23204	Сурманавская Анастасия	7939	25.03.2023	12:00	Лабораторные работы	103	<input type="checkbox"/>			
4	История	Гр_23204							<input type="checkbox"/>			
5									<input type="checkbox"/>			
6	Линейная алг...	Гр_23201	Розанова Анна	7927	22.03.2023	10:00	ТК2	102	<input type="checkbox"/>			
7	Математическ...	Гр_23202	Александр Анастасия	7965	22.03.2023	17:00	Практическая работа 1-5	106	<input type="checkbox"/>			
8	Культурология...	Гр_23204	Сурманавская Анастасия	7939	20.03.2023	15:00	Презентация	105	<input type="checkbox"/>			
9	Введение в пр...	Гр_23204	Садонина Екатерина	7960	28.03.2023	15:00	Презентация	105	<input type="checkbox"/>			
10		Гр_23203	Петровиш Ашот	7967					<input type="checkbox"/>			
11									<input type="checkbox"/>			
12									<input type="checkbox"/>			
13									<input type="checkbox"/>			
14									<input type="checkbox"/>			
15									<input type="checkbox"/>			

Рис. 4. База данных

Для взаимодействия между студентом и базой данных служит Telegram бот, который выполняет функцию связующего звена для получения информации.

Алгоритм взаимодействия студента с Telegram ботом будет следующим. Студент отправляет команду Telegram боту для запуска с его взаимодействием командой «/start». Telegram бот запросит разрешение на использование его контактного номера телефона для поиска академических задолженностей. На рисунке 5 показан код для запроса на обработку информации контактного номера телефона.

Если задолженностей нет, то последует ответ от бота «Задолженностей нет». Если задолженности будут найдены, то бот отправит смс сообщение с датой, временем, номером аудитории, предметом и наименованием задолженности. После данного смс сообщения бот предложит создать напоминание на дату, которая указана в поле для явки на сдачу академического долга. На рисунке 6 наглядно показан диалог студента с Telegram ботом.

```

1 function doPost(e) {
2   var contents = JSON.parse(e.postData.contents);
3
4   debug(contents);
5
6   //если пользователь отправил сообщение
7   if (contents.message.text)
8   {
9     var msg = contents.message;
10    var chat_id = msg.chat_id;
11    var text = msg.text;
12    //если пользователь отправил команду старт
13    if (text == "/start")
14    {
15      send_apply(chat_id); //вызывает функцию send эплэй
16    }
17    //если пользователь отправил кнопку узнать задолженности
18    if (text == "Узнать задолженности"){
19      telephone = get_number(contents.chat_id);
20      if (telephone != ""){
21        check_debts(telephone,chat_id); //вызывает функцию проверки задолженностей
22      }
23      else{
24        send_apply(chat_id);
25      }
26    }
27  }
28
29  //если пользователь отправил номер телефона
30  if (contents.message.reply_to_message)
31  {
32    var msg = contents.message.reply_to_message;
33    var chat_id = msg.chat_id;
34    telephone = get_number(contents.chat_id); //вызывает функцию для получения мобильного
35    //sendText (chat_id, telephone);
36    sendText (chat_id, "Спасибо", KEY_DELETE) //удаляет кнопку "Разрешить"
37    if (telephone != ""){
38      check_debts(telephone,chat_id); //вызывает функцию проверки задолженностей
39    }
40  }
41 }

```

Рис. 5. Код запроса на использование номера телефона

Для поиска академических задолженностей Telegram ботом был создан запрос на поиск определенной строки в базе данных по контактному номеру телефона пользователя. Запрос был создан на языке google app script, код которого продемонстрирован на рисунке 7.

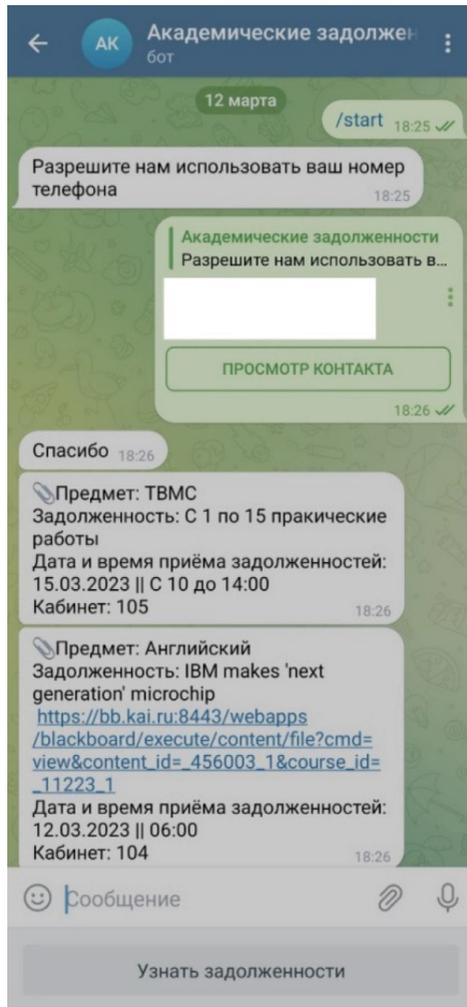


Рис. 6. Результат получения информации о академических задолженностях

Для наглядности алгоритма действий была создана схема взаимодействия всех участвующих в процессе реализации проекта звеньев. Схема продемонстрирована на рисунке 8.

```

1 function timerAlarm() {
2   DataSheet = SpreadsheetApp.openById(spreadSheetId).getSheetByName("Таблица");
3   UsersSheet = SpreadsheetApp.openById(spreadSheetId).getSheetByName("Users");
4   let data = DataSheet.getRange(2, 1, DataSheet.getLastRow()-1, 8).getValues(); //все данные таблицы в массив
5   let curentTime = new Date().getDate() + "." + (new Date().getMonth() + 1) + "." + new Date().getFullYear(); //тек дата
6
7   for (i = 0; i < data.length; i++) {
8     let dataInfo = data[i];
9     var j = 0;
10    //разбиваем все данные по переменным
11    let subject = dataInfo[0];
12    let group = dataInfo[1];
13    let name = dataInfo[2];
14    let number = dataInfo[3];
15    let appointment = dataInfo[4].toString(); //дата явки
16    let time = dataInfo[5];
17    let debt = dataInfo[6];
18    let cabinet = dataInfo[7];
19
20    if (dataInfo[4] !== ""){ //преобразует дату в читабельный формат для сравнения с таблицей
21      appointment = dataInfo[4].getDate() + "." + (dataInfo[4].getMonth() + 1) + "." + dataInfo[4].getFullYear();
22    }
23
24    //if (dataInfo[5] !== ""){ //преобразует время в читабельный формат
25    //  time = dataInfo[5].getHours()+ "." + dataInfo[5].getMinutes();
26    //}
27
28    if (appointment === curentTime){ //если дата в таблице долгов равна тек дате
29      for (j = 2; j <= UsersSheet.getLastRow(); j++){
30        if (UsersSheet.getRange(j, 2).getDisplayValue() === number){ //ищет пользователя по номеру телефона, чтобы взять его id
31          sendText(UsersSheet.getRange(j, 1).getDisplayValue(), "Д сегодня принимается задолженности по предмету " + subject + "\n" +
32            "Задолженность: " + debt + "\n" +
33            "Время приема задолженностей: " + time+ "\n" +
34            "Кабинет: " + cabinet);
35        }
36      }
37    }
38  }

```

Рис. 7. Код создания запроса ботом в базу данных

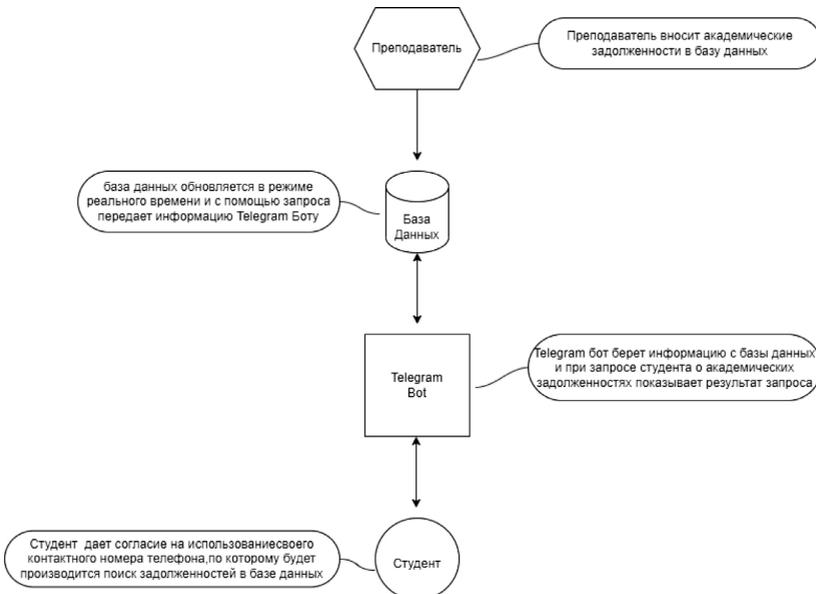


Рис. 8. Процесс получения информации о академических задолженностях

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

Также были проведены тесты для опроса на удобство использования данного проекта на практике. Данный проект мультизадачен и может использоваться в разных образовательных учреждениях. Возможна модернизация базы данных для оптимизации рабочего времени преподавателей.

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 [Электронный ресурс] / 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. [Электронный ресурс] / Н. Л. Аванесян // Научный потенциал XXI века. – 2017.
8. Иванов А. Д., Чат-бот в Telegram и ВКонтакте, как новый канал распространения новостей. [Электронный ресурс] / А. Д. Иванов

- // Волжский университет имени В.Н. Татищева. – 2016. – №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 [Электронный ресурс] – Режим доступа: <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 [Электронный ресурс]: URL: <http://w7phone.ru/chat-boty-istoriya-sovremennost-i-perspektivy-132460>
 15. Telegram Bot API [Электронный ресурс]: URL: <https://core.telegram.org/bots/api>

References

1. Tugov V.V. Designing automated control systems: a training manual / V.V. Tugov, A.I. Sergeev, N.S. Sharov St. Petersburg: Lan', 2019. - 172 p.
2. Models and methods of research of information systems: a monograph / A.D. Khomonenko, A.G. Basyrov, V.P. Bubnov [et al]. - Saint Petersburg: Lan', 2019. - 204 p.

3. Zlatopolsky D.M. Fundamentals of programming in the Python language. - Moscow: DMK Press, 2017. - 284 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 Economy and Public Administration, Moscow State Pedagogical University. Tambov: Consulting company Yukom, 2018. - 120 p.
5. Vinogradova E. Yu. Intelligent information technologies - theory and methodology of building information systems: monograph / Ministry of Education and Science of the Russian Federation, Ural State University of Economics. - Ekaterinburg: Publishing house of the Ural State University of Economics, 2011. - 263 p.
6. Kozlov A. A. Telegram-bot as a simple and convenient way to obtain information [Electronic resource] / A. A. Kozlov, A. V. Batischev // Territory of Science. - 2017. - № 5. - pp. 55-64.
7. Avanesyan N. L., Telegram as an example of messenger: opportunities and prospects for development. [Electronic resource] / N. L. Avanesyan // Scientific Potential of the XXI century. - 2017.
8. Ivanov A. D., Chat-bot in Telegram and VKontakte as a new news distribution channel. [Electronic resource] / A. D. Ivanov // Volga V.N. Tatishchev University. - 2016. -№3 - pp. 126-132.
9. Nasibulin R.O. Creation of a simple moderator bot for Telegram messenger using Python programming language / R.O. Nasibulin, G.A. Gareeva // Youth and system modernization of the country: Collection of scientific articles of the 7th International Scientific Conference of Students and Young Scientists. In five volumes, Kursk, May 19-20, 2022 / Editor-in-chief M.S. Razumov. Volume 3. - Kursk: Southwestern State University, 2022. - pp. 425-427.
10. Gareeva G.A. Development of HR-bot to automate the recruitment process in a manufacturing enterprise / G.A. Gareeva, I.M. Safonov, Z.G. Dzhibladze [et al.] // Scientific and Technical Herald of the Volga Region. - 2023. - № 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. Chat-bots: history, modernity and prospects [Electronic resource]: URL: <http://w7phone.ru/chat-boty-istoriya-sovremennost-i-perspektivy-132460>
15. Telegram Bot API [Electronic resource]: URL: <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.

Новичков А.А.: проведение сбора данных, подготовка начального варианта статьи.

Лукин Д.Г.: анализ и интерпретация полученных данных, литературный анализ.

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

AUTHORS CONTRIBUTION

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.

Andrey A. Novichkov: carrying out data collection, preparation of the initial version of the article.

Daniel G. Lukin: 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*

Новичков Андрей Анатольевич, студент

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

Лукин Даниил Георгиевич, студент

*Казанский национальный исследовательский технический университет им. А.Н. Туполева-КАИ
ул. Академика Королева, 1, г. Набережные Челны, 423814,
Российская Федерация
d.luckin2015@yandex.ru*

Гареева Гульнара Альбертовна, заведующий кафедрой Информационных систем, канд. пед. наук, доцент
Казанский национальный исследовательский технический университет им. А.Н. Туполева-KAI
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, Russian 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, Russian Federation
2db6db@mail.ru

Andrey A. Novichkov, student

Kazan National Research Technical University named after A.N. Tupolev-KAI
1, Akademika Koroleva Str., Naberezhnye Chelny, 423814, Russian Federation
novichkovandrew@mail.ru

Daniel G. Lukin, student

Kazan National Research Technical University named after A.N. Tupolev-KAI

1, Akademika Koroleva Str., Naberezhnye Chelny, 423814, Russian Federation

d.luckin2015@yandex.ru

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 09.06.2023

После рецензирования 25.06.2023

Принята 05.07.2023

Received 09.06.2023

Revised 25.06.2023

Accepted 05.07.2023