УДК 004.9

# Пример использования информационных технологий для решения повседневных задач

*Николаев Сергей Валерьевич*
*Приамурский государственный университет им. Шолом-Алейхема*
*магистрант*

*Пасюков Александр Андреевич*
*Приамурский государственный университет им. Шолом-Алейхема*
*магистрант*

*Якимов Антон Сергеевич*
*Приамурский государственный университет им. Шолом-Алейхема*
*магистрант*

*Баженов Руслан Иванович*
*Приамурский государственный университет им. Шолом-Алейхема*
*к. п. н., доцент, зав. кафедрой информационных систем, математики и методик обучения*

**Аннотация**
Цель данной статьи продемонстрировать как применение информационных технологий может упростить задачу сотруднику и повысить качество работы. В качестве примера взята задача, в которой нам необходимо ввести мониторинг файловой папки в операционной системе Windows 10 и при определенных условиях оповещать об изменениях, которые в ней произошли. В решении использовались язык программирования C# и платформа .NET, встроенные функции операционной системы, несколько сторонних библиотек и API мессенджера Telegram.
**Ключевые слова**: Информационные технологии, языки программирования, API, C#, .NET, Telegram bot.

# An example of the use of information technology for solving everyday tasks in labor activity

*Nikolaev Sergey Valerievich*
*Sholom-Aleichem Priamursky State University*
*Undergraduate*

*Pasyukov Alexandr Andreevich*
*Sholom-Aleichem Priamursky State University*
*Undergraduate*

*Yakimov Anton Sergeevich*
*Sholom-Aleichem Priamursky State University*
*Undergraduate*

*Bazhenov Ruslan Ivanovich*
*Sholom-Aleichem Priamursky State University*
*Candidate of pedagogical sciences, associate professor, Head of the Department of Information systems, Mathematics and teaching methods*

**Abstract**
The purpose of this article is to demonstrate how the application of information technology can simplify the task of an employee and improve the quality of work. As an example, the task is taken in which we need to enter the monitoring of the file folder in the operating system Windows 10 and, under certain conditions, notify about the changes that occurred there. The solution used the C # programming language and the .NET platform, built-in functions of the operating system, several third-party libraries and the Telegram messenger API.
**Keywords**: Information technologies, programming languages, API, C #, .NET, Telegram bot.

В информационном обществе почти каждая профессия в той или иной степени связана с работой на компьютере. А наличие смартфонов у большинства членов такого общества не подвергается сомнению.

Подобная среда открывает возможности для решения различных задач, связанных с повседневной жизнью, использую современные технологии, такие как языки программирования, различные фреймворки и библиотеки.

О таких технологиях рассказывают в своих книгах Д. Рихтер [1], Э. Троелсен [2], Г.Шилдт [3]. В работе В. И. Кручинина излагаются методические материалы для изучения основ программирования [4].

Цель данной работы продемонстрировать на примере подход к решению обычной рутинной задачи для повышения качества труда и комфорта сотрудника.

В качестве примера возьмем следующую задачу: У нас есть сотрудник, чьей обязанностью является публикация материала на сайт. Условно назовем его контент менеджер (далее КМ) На сервере организации есть каталог с общим доступом (далее «папка обмена»), куда может поместить информацию любой из служащих. КМ должен регулярно проверять «папку обмена» на наличие нового материала и публиковать его на сайт. Задачу усложняет, что подобных папок несколько. При плохом мониторинге данных папок появляются несколько проблем: срочная информация выставляется не вовремя, образуются скопления информации и как в следствии нагрузка неравномерна, другие сотрудники нервничают и заставляют нервничать КМ, что в дальнейшем влияет на работоспособность и тех, и других.

Рассмотрим задачу. В условии ярко выражены действия, которые повторяются регулярно из раза в раз. Это мониторинг «папки обмена».

Необходимо просмотреть каждую папку, отсеять информацию, которая находится там всегда (например, правила оформления материалов для сайта), отметить новые файлы и забрать себе. Автоматизируем данный процесс. Можно выделить две подзадачи, это собственно сам мониторинг каталогов и оповещение о размещении в них новых файлов. В первой планируется реализовать метод(функцию) для извлечения файлов, помещенный в замкнутый цикл. Но при этом необходимо установить определенную периодичность. Во второй имеются несколько вариантов:

- Выводим информацию на экран;
- Подаем звуковой сигнал на компьютер;
- Высылаем письмо на электронную почту;
- Отправляем сообщение в мессенджер, например, в Telegram.

Все эти способы не противоречат друг другу и ничего не мешает реализовать их все в одной версии программы. Как это реализовать, будет продемонстрировано далее.

В качестве реализации данной задачи автоматизации был выбран язык программирования C# и платформа .Net Framework, в следствии чего все примеры кода будут приведены на нем. Стоит уточнить, что данный выбор не обусловленным каким-то преимуществом технологии, данной задачу вполне способен реализовать практически любой язык программирования высокого уровня с примерно тем же уровнем трудозатрат.

Метод для извлечения списка файлов и подкаталогов из необходимых нам каталогов представлена на рисунке 1.

```
1  public static IEnumerable<string> GetFilesFrom(List<string> paths, List<string> ignoreList)
2  {
3    var filesList = new List<string>();
4    foreach (var path in paths)
5    {
6      filesList.AddRange(Directory.GetDirectories(path)
7        .Except(ignoreList));
8    }
9    return filesList;
10 }
```

Рисунок 1. Метод извлечение списка каталогов и файлов из целевых директорий

Метод принимает на вход список полных путей файлов и каталогов, а также, список полных путей файлов и каталогов, которые нужно игнорировать (файлы и каталоги, которые всегда присутствуют в «папке обмена»). Возвращает метод объединенный список файлов и каталогов со всех целевых директорий в виде полных путей.

Для того, чтобы выполнение было цикличным использовался класс Timer располагающийся в пространстве имен «System.Timers» .Net Framework-а. Пример его применения можно увидеть на рисунке 2.

```
1   static void Main(string[] args)
2   {
3     var timer = new Timer(1000 * 60 * 10);
4     timer.Elapsed += Timer_Elapsed;
5     timer.Start();
6
7     Console.ReadKey();
8   }
9
10  private static void Timer_Elapsed(object sender = null, ElapsedEventArgs e = null)
11  {
12    //Здесь должны быть описаны команды,
13    //которые будут исполняться раз в 1000*60*10 мс (10 минут)
14  }
```

Рисунок 2. Использование класса Timer

В конструкторе класса мы задаем параметр в виде времени в мс, это и есть частота исполнения события. Тело события должно быть описана чуть ниже (в данном случае событие имеет имя Timer_Elapsed). В нашем случае это должен метод описанный на рисунке 1 и другие команды, реализующие функции оповещения пользователя.

Перейдем к оповещению. Начнем с простого, вывода на экран и оповещения звуком. Для этого достаточно использовать методы консоли: Console.WritleLine Console.Beep. Но что если КМ находится не рядом со своим компьютером? Тогда данные оповещения будут бесполезны, так как не смогут привлечь его внимания.

Намного эффективнее, в данном случае, использовать другие виды оповещения. Например, письмо на электронную почту, или сообщение в мессенджер. И то, и другое требует аккаунт или почтовый ящик, но создать их не составляет особого труда.

Создадим класс Mail, который будет отвечать за отправку письма. Пример реализации представлен на рисунке 3. В качестве почтового сервиса используется Gmail. В 13 строке используется собственный метод для генерации содержимого. Вместо него вполне можно использовать обычную переменную типа string. На вход конструктор принимает какой-то текст для отправки и дату, которая так же может быть использована при отправке, например, в заголовке письма. Для отправки сообщения необходимо создать экземпляр класса Mail и вызвать метод Send.

Перейдем к реализации оповещения через мессенджер. В качестве мессенджера был выбран Telegram. Отправка будет осуществляться с использованием Telegram Bot. Его разработчики предоставляют неплохой API для взаимодействия и документацию по нему. С документацией можно ознакомиться на официальном сайте: https://tlgrm.ru/docs/bots/api. Так же для .Net Framework имеется готовая библиотека «Telegram.Bot» от Робина Мюллера, которая сильно упрощает взаимодействие с Telegram Bot. Найти, установить и использовать данную библиотеку можно с помощью NuGet Manager (менеджер пакетов в .Net Framework).

```
public class Mail
{
  private MailMessage message;
  private SmtpClient smtp;

  public Mail(List<string> text, DateTime dateTime)
  {
    var from = new MailAddress("<Отпрвитель>", $"<Заголовок письма> {dateTime}");
    var to = new MailAddress("<Электронный ящик получателя>");
    message = new MailMessage(from, to)
    {
      Subject = "В папке обмена присутствует неопубликованный материал",
      Body = MailBodyBuilder.Build(text), // содержимое письма
      IsBodyHtml = true
    };
    smtp = new SmtpClient("smtp.gmail.com", 465)
    {
      Credentials = new NetworkCredential("<электронный ящик для доступа к почтовому сервису>",
      "<пароль электронного ящика>"),
      EnableSsl = true
    };
  }
  public void Send()
  {
    smtp.Send(message);
  }
}
```

Рисунок 3. Реализация класса Mail

Но для того, чтобы воспользоваться библиотекой, нам необходимо создать бота. Для этого необходимо через Telegram отправить команду «/newbot» специальному боту «BotFather» (telegram.me/BotFather). После выполнения инструкций BotFather вышлет информацию по созданному боту, в которой будет токен доступа к HTTP API Telegram. На рисунке 4 представлен диалог с «BotFather».

Теперь созданного бота можно найти по адресу: t.me/Demo_Article_Bot. А токен «382804046:AAEYo8TRV7inEZyAtu-sY1Fs9zvb3ANjqtg» использовать для работы с API. Важно никому не раскрывать этот токен, так как злоумышленник может использовать его в своих целях. Через «BotFather» производятся и другие настройки бота, например, создание команд. Чаще всего подобные боты используются для диалога с пользователем. Где пользователь вводит какую-то команду, а бот реагирует необходимым способом. За алгоритм обработки ответа отвечает какой-либо веб-сервис, который и взаимодействует с Telegram Bot. Подобный веб-сервис, должен быть где-нибудь развернут и принимать запросы из сети интернет, иначе бот не сможет с ним взаимодействовать.

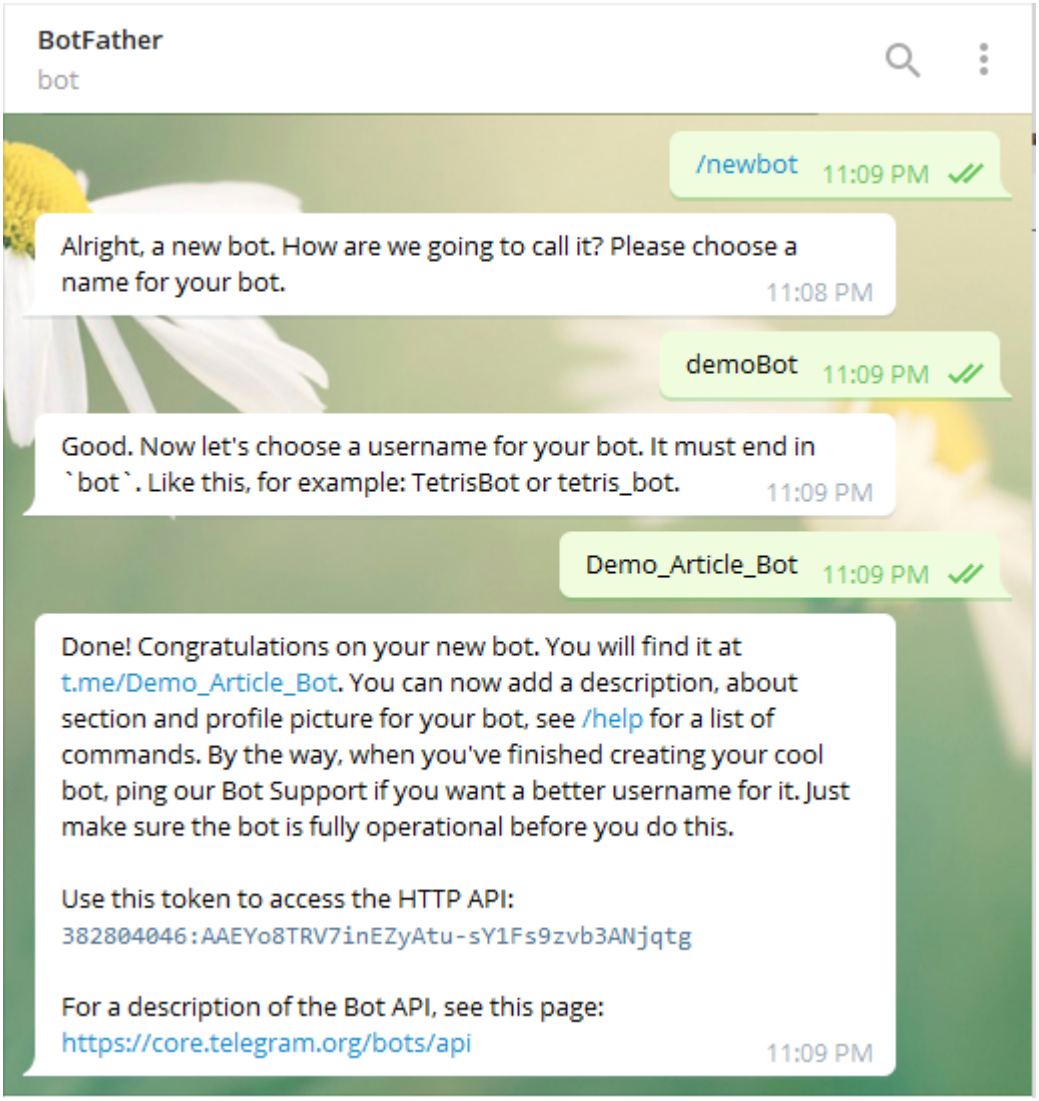


Рисунок 4. Процесс создания нового бота в Telegram

Но, для поставленной цели не нужен диалог с ботом. Поток сообщений будет односторонний, от бота к пользователю. Поэтому разрабатывать веб-сервис нет необходимости. Все, что необходимо боту – это доступ в интернет и идентификационный номер (Id) пользователя в Telegram, для возможности отправки сообщений. Чтобы узнать свой Id необходимо написать любой сообщение на t.me/userinfobot. Данный Id будет использоваться в разрабатываемом классе для отправки сообщений. В свою очередь, чтобы получать сообщения от бота, пользователь должен добавить его себе в контакты, иначе сообщения не будут приходить. Реализация класса для отправки сообщений через Telegram представлена на рисунке 5.

В конструкторе используется токен доступа для создания объекта бота. Отправка сообщения организуется с помощью метода Send.

```
1   public class TelegramSender
2   {
3     private TelegramBotClient bot;
4
5     public TelegramSender()
6     {
7       bot = new TelegramBotClient("<Токен доступа бота>");
8     }
9
10    public void Send(string text)
11    {
12      bot.SendTextMessageAsync("<Id пользователя получателя>", text).GetAwaiter();
13    }
14  }
```

Рисунок 5. Класс, реализующий отправку сообщений в Telegram

Теперь можно использовать созданные классы оповещения в событии Timer_Elapsed. Его итоговый вид представлен на рисунке 6.

В данном классе целевые директории и файлы, которые необходимо игнорировать заданы жестко в исходном коде. Это не лучшее решение. Хорошей практикой будет либо считывать эти списки с текстовых файлов, либо предоставить пользователю возможность через интерфейс менять данные списки. В данном случае это сделано для упрощения.

В итоге была написана программа, которая автоматизировала процесс мониторинга определенных директорий в операционной системе с определенной периодичностью. Были реализованы несколько способов оповещения пользователя. В процессе разработки возникали спорные решения: периодичность мониторинга, включать ли в процесс копирования файлов из папки обмена в папку КМ, каким образом хранить списки целевых директорий, списки игнорируемых файлов. Все данные решения принимались на усмотрение КМ, так как решался частный случай поставленной задачи. Преобразование данного частного решения к общему отдельный процесс исследования, который будет осуществлен в будущем.

```
private static void Timer_Elapsed(object sender = null,
    ElapsedEventArgs e = null)
{
  var telegram = new TelegramSender();
  var monitoringFolders = new List<string>
  {
    "C:/папка обмена", "C:/папка обмена2"
  };
  var files = DirectoryProvider.GetFilesFrom(monitoringFolders,
    new List<string>
    {
      "C:/папка обмена/правила офрмления.doc",
      "C:/папка обмена2/правила офрмления.doc"
    })
    .Except(DefaultFiles.Get(pathIgnoreList)).ToList();
  Console.Clear();
  Console.WriteLine(DateTime.Now);

  if (files.Count() == 0)
    Console.WriteLine("Новых файлов нет!");
  else
  {
    var output = string.Join("\n", files);
    // Вывод строк на экран консоли
    Console.WriteLine(output);

    //Звуковое оповещение (частота сигнала, длительность в мс)
    Console.Beep(3000, 3000);

    //Отправка письма на почту
    var mail = new MailLib.Mail(files, DateTime.Now);
    mail.Send();

    //Отправка через Telegram
    telegram.Send(output);
  }
}
```

Рисунок 6. Реализация Timer_Elapsed

**Библиографический список**


1. Рихтер Дж. CLR via C#. Программирование на платформе Microsoft .NET Framework 4.0 на языке C#. М.: Питер, 2013. 928 с.
2. Троелсен Э. Язык программирования C# 5.0 и платформа .NET 4.5 /. М.: Вильямс, 2015. 486 с.
3. Шилдт Г. C# 4.0: полное руководство. М.: Вильямс, 2011. 1056 с.
4. Кручинин В. И. Лабораторный практикум по основам программирования: учеб. пособие. Волгоград: ИУНЛ ВолгГТУ, 2014. 92 с.