

# РАЗРАБОТКА И РЕАЛИЗАЦИЯ ПРОГРАММЫ ДЛЯ БИОМЕТРИИ ПОЛЬЗОВАТЕЛЯ ПЕРСОНАЛЬНОГО КОМПЬЮТЕРА НА БАЗЕ ОПРЕДЕЛЕНИЯ ПАРАМЕТРОВ КЛАВИАТУРНОГО ПОЧЕРКА

А.А. Вязигин, Н.Ю. Тупикина, Е.В. Сыпин

*Бийский технологический институт (филиал) ФГБОУ ВО «Алтайский государственный технический университет им. И.И. Ползунова», г. Бийск*

В данной статье рассматривается разработка и реализация программного средства для определения параметров клавиатурного почерка пользователя персонального компьютера, определены функциональные и технические требования к программному средству для определения параметров клавиатурного почерка пользователя персонального компьютера, изучены способы определения клавиатурного почерка пользователей, а также способы идентификации пользователя с его использованием. Разработана программа для персонального компьютера для определения клавиатурного почерка пользователя и его идентификации.

*Ключевые слова:* биометрия, идентификация, клавиатурный почерк, программирование, алгоритмизация.

## ВВЕДЕНИЕ

С развитием информационных технологий происходило их массовое внедрение, что привело к практически повсеместному использованию персональных компьютеров, особенно для таких рутинных задач, как набор текста. Появление термина «клавиатурный почерк» связано с тем, что каждый человек, работающий за компьютером, имеет уникальные параметры, характеризующие процесс набора символов [1].

Основной интерес клавиатурный почерк представляет в области биометрии, как дешёвый способ идентификации и аутентификации пользователей. Использование клавиатурного почерка в системах аутентификации пользователей имеет ряд преимуществ:

не требуется дополнительное оборудование, достаточно стандартной клавиатуры, что обуславливает относительную дешевизну системы распознавания пользователей;

не требуются дополнительные факторы аутентификации, что позволяет полностью отказаться от использования сложных паролей;

появляется возможность контролировать доступ пользователя к ресурсам системы, а также его физиологическое состояние.

## ПОСТАНОВКА ЗАДАЧИ

В рамках данной работы было необходимо решить следующие задачи:

- 1) изучить особенности и способы определения клавиатурного почерка пользователей;
- 2) разработать структуру программного средства;
- 3) разработать формат хранения пользовательских данных и алгоритм работы программы;

4) реализовать программу для определения параметров клавиатурного почерка пользователя.

## ПОНЯТИЕ КЛАВИАТУРНЫЙ ПОЧЕРК. ПАРАМЕТРЫ КЛАВИАТУРНОГО ПОЧЕРКА

Клавиатурный почерк – это подробная динамическая информация, которая точно описывает особенности набора текста на клавиатуре конкретным человеком.

Клавиатурный почерк представляет собой биометрический шаблон, используемый для идентификации человека, основанный на схеме ввода, ритме и скорости ввода на клавиатуре [1]. Например, установлено, что время, необходимое для нажатия клавиши, время поиска клавиши, и время удержания клавиши может быть очень характерным для человека, независимо от того, насколько быстро он печатает. У большинства людей есть конкретные буквы, поиск и нажатие на которые занимает больше времени. Правши статистически быстрее в наборе букв, которые они нажимают пальцами правой руки, чем они же пальцами левой руки.

Использование клавиатурного почерка для идентификации пользователя имеет много плюсов:

алгоритмическая реализация алгоритмов определения компьютерного почерка достаточно проста и может быть реализована как в виде программы, так и отдельного устройства;

дешевизна реализации, для работы алгоритма достаточно обычной клавиатуры, которая по умолчанию входит в комплект любого компьютера;

алгоритм можно использовать неявно, пользователь может и не подозревать что используется дополнительно средство идентификации;

достаточно высокая эффективность.

Однако, данный метод имеет недостатки:

требуется этап обучения и получения эталонных значений для сравнения;

при смене клавиатуры, с высокой вероятностью понадобится заново собирать статистику;

так как поведенческая биометрия напрямую зависит от состояния человека, при его изменении (болезнь, усталость, волнение), система может не опознать пользователя.

Динамика нажатия клавиш обладает широким набором информации для сбора, в том числе [1]:

интервал между последовательными нажатиями клавиш;

время удержания клавиш;

время «полёта»;

время ожидания;

общая скорость набора (количество символов в единицу времени);

частота ошибок набора;

анализ использования характерных клавиш управления и клавиш управления курсором.

На рисунке 1 на временной шкале изображены основные параметры для сбора информации о динамике нажатий клавиш пользователем.

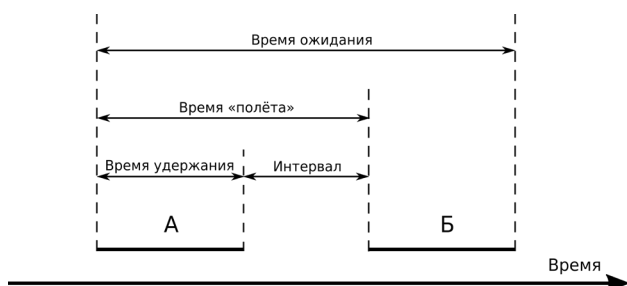


Рис. 1. Параметры динамики нажатий клавиш

При реализации алгоритма, для наибольшей достоверности оценки показателей пользователя необходимо учитывать все вышеуказанные параметры.

### ОБЗОР АЛГОРИТМОВ ОПРЕДЕЛЕНИЯ КЛАВИАТУРНОГО ПОЧЕРКА

Задача распознавания пользователя решается либо путем перебора информации из базы данных, либо с помощью сравнения с неким шаблоном поведения. Существуют несколько подходов по определению клавиатурного почерка.

1) Наиболее простой метод определения компьютерного почерка состоит в сборе информации о динамических характеристиках наиболее часто встречающихся для данного языка диграмм, триграмм или в более общем случае n-грамм из всего текста [1]. В этом случае параметры собираются на основе ввода фиксированных текстов (статически) или переменных текстов (динамически). В случае последнего метода статистику можно собирать непрерывно, чтобы

постоянно идентифицировать пользователя. Собранные данные усредняются и на их основе создаётся эталон для сравнения. Верификация заключается в вычислении разницы сравнения тестовой подписи (полученной в момент сбора) и эталоном. Положительное решение принимается в случае, если разница не превышает принятое пороговое значение.

2) Метод, разработанный Риком Джойсом и Гопалом Гупта, который заключается в представлении динамических данных в виде вектора M, вычисляемого по формуле [2]:

$$M = \{M_{\text{имя пользователя}}, M_{\text{пароль}}, M_{\text{имя}}, M_{\text{фамилия}}\} \quad (1)$$

Пусть  $M = (m_1, m_2, \dots, m_n)$  и  $T = (t_1, t_2, \dots, t_n)$ , где n – общее количество задержек в подписи. Верификация заключается в сравнении тестовой подписи T (полученной при входе в систему) с M, вычисляя норму  $L_1$  по формуле:

$$L_1 = \|M - T\|_1 \quad (2)$$

полученной из

$$\sum_{i=1}^n |m_i - t_i| \quad (3)$$

Положительное решение принимается в случае, если величина  $L_1$  не превышает принятое пороговое значение.

3) Метод разработанный Фабианом Монроузом, базирующийся на исследования Джойса и Гупта и использовании нейронных сетей [3, 4]. Данные о пользователях представлялись в виде N-мерных векторов характеристик, которые обрабатываются методом Джойса и Гупта. Полученные данные разделялись на наборы для обучения и проверки. Для идентификации пользователя используются следующие классификаторы:

Евклидово расстояние между двумя N-мерными векторами U и R. Пусть  $U = [u_1, u_2, \dots, u_n]$ ,  $R = [r_1, r_2, \dots, r_n]$  тогда Евклидово расстояние между ними определяется как:

$$D(R, U) = \left[ \sum_{i=1}^n (r_i - u_i)^2 \right]^{1/2} \quad (4)$$

Для неизвестного U из тестового набора  $D(R, U)$ ,  $i = 1, 2, \dots, n$ , где n – количество проверочных наборов. В результате выбирается набор с минимальным расстоянием U.

невзвешенная вероятность: пусть U и R – N-мерные вектора, определенные выше. Каждая компонента векторов состоит из четырёх компонент и включает в себя набор  $\{\mu_i, \sigma_i, o_i, X_i\}$ , состоящий из средней величины  $\mu_i$ , стандартного отклонения  $\sigma_i$ , частоты события  $o_i$  и значения X для i-ой характеристики. Предполагая, что каждая компонента подчиняется нормальному распределению, оценка между

опорным профилем  $R$  и неизвестным профилем  $U$  вычисляется как:

$$Score(R, U) = \sum_{i=1}^n S_{ui} \quad (5)$$

где

$$S_{ui} = \frac{1}{\sigma_{ui}} \left[ \sum_{j=1}^{\sigma_{ui}} Prob \left( \frac{x_{ij}^{(u)} - \mu_{ri}}{\sigma_{ri}} \right) \right] \quad (6)$$

где  $x_{ij}^{(u)}$  –  $j$ -ое появление  $i$ -ой характеристики в  $U$ .

Другими словами, баллы для каждого  $u_i$  зависят от вероятности наблюдения величины  $u_{ij}$  в профиле  $R$  при средней ( $\mu_{ri}$ ) и стандартном отклонении ( $\sigma_{ri}$ ) для данной характеристики в  $R$ . Большую вероятность имеют значения  $u_i$ , которые находятся ближе к  $\mu_{ri}$ , меньшую – те, что дальше. Вектор  $U$  ассоциируется с ближайшим соседом в базе данных, т. е. с оператором, для которого вектор характеристик будет максимальным;

взвешенная вероятность: некоторые параметры являются более надежными, чем другие, потому что они представлены в большем количестве профилей или имеют более высокий процент в письменном языке; например, английские *er, th, ge* имеют больший вес чем *qu* или *ts*. Исходя из этого вводится понятие веса, тогда оценка между профилями  $R$  и  $U$  вычисляется как

$$Score(R, U) = \sum_{i=1}^n (S_{ui} * w_{ui}) \quad (7)$$

где вес характеристики  $u_i$  определяется как соотношение частоты её появления к частоте появления всех характеристик в  $U$ . Характеристики с большим весом рассматриваются как более надежные. Предположим, что каждая характеристика пользователя подвержена нормальному распределению, тогда баллы в пользу профиля  $R$  при введенном  $U$  определяются максимальными баллами и ближайшим вектором характеристик.

Использование данных классификаторов позволяет распознать человека в 87,18 % случаев. При использовании Байесовых классификаторов процент распознавания вырастает до 92,14 %, ещё 5 % можно получить подбирая весовые коэффициенты.

#### РАЗРАБОТКА СТРУКТУРЫ ПРОГРАММНОГО СРЕДСТВА ОПРЕДЕЛЕНИЯ КЛАВИАТУРНОГО ПОЧЕРКА ПОЛЬЗОВАТЕЛЕЙ ПЕРСОНАЛЬНОГО КОМПЬЮТЕР

Структура – совокупность устойчивых связей объекта, обеспечивающих его целостность и тождественность самому себе при различных внешних и внутренних изменениях. В общем случае под структурой понимается совокупность составных частей некоторого объекта, обеспечивающих его целостность и тождественность самому себе, т.е. сохранение основных

свойств при различных внешних и внутренних изменениях.

На рисунке 2 представлена структура программы для определения клавиатурного почерка пользователей.

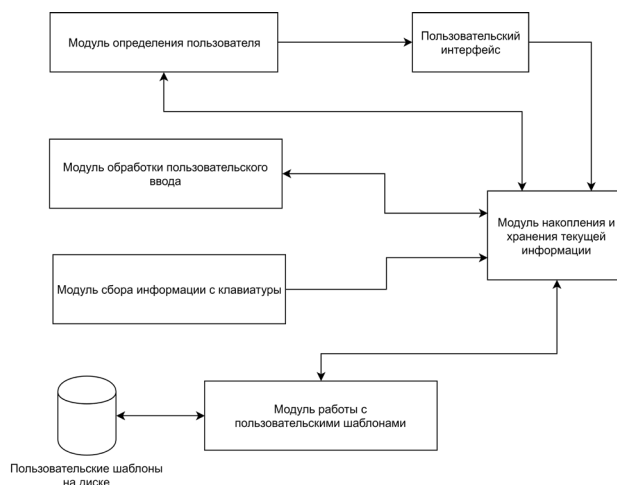


Рис. 2. Структура программы

Для удобства разработки и поддержки, программа была разделена на несколько модулей. Все модули запускаются в параллельных потоках.

#### РАЗРАБОТКА ФОРМАТА ХРАНЕНИЯ ПОЛЬЗОВАТЕЛЬСКИХ ШАБЛОНОВ

Для хранения пользовательских шаблонов был выбран XML-формат. XML (eXtensible Markup Language) – расширяемый язык разметки, созданный консорциумом всемирной паутины (W3C). XML создавался как язык с простым синтаксисом, удобный как для создания и обработки программными средствами, так и удобный для чтения человеком. Язык называется расширяемым, поскольку он не фиксирует разметку, используемую в документах: разработчику необходимо её создать в соответствии с конкретной предметной областью, будучи ограниченным лишь синтаксическими правилами языка. Сочетание простого синтаксиса, удобства для человека, расширяемости, а также использование Юникод кодировок привело к широкому использованию как собственно XML, так и множества производных языков на его базе в самых разнообразных программных средствах [6]. Полученную структуру пользовательского шаблона можно представить в виде следующей древовидной модели (рисунок 3).

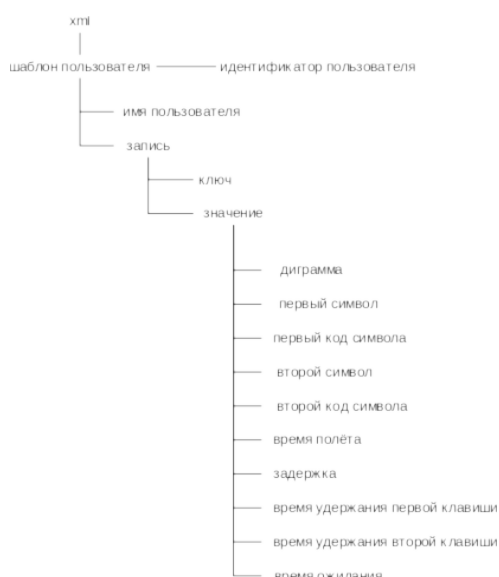


Рис. 3. Структура пользовательского шаблона

Пример шаблона пользователя представлен на рисунке 4.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<userPattern userId="fffff2c2e-cc8d-4d0b-95b0-67349da77cff">
  <userDigramStat>
    <entry>
      <key>aб</key>
      <value>
        <digram>aб</digram>
        <firstChar>a</firstChar>
        <firstDwellTime>80453219</firstDwellTime>
        <firstKeyCode>70</firstKeyCode>
        <flightTime>264794929</flightTime>
        <latency>184341710</latency>
        <secondChar>б</secondChar>
        <secondDwellTime>85971451</secondDwellTime>
        <secondKeyCode>188</secondKeyCode>
        <waitTime>350766380</waitTime>
      </value>
    </entry>
  </userDigramStat>
  <userName>Андрей</userName>
</userPattern>
```

Рис. 4. Пример шаблона пользователя

На рисунке 4 изображено конечное представление структуры пользовательского шаблона в соответствии со структурой шаблона (рисунок 3) с полным соответствием полей, например: поле диграмма соответствует полю digram в конечном представлении, аннотация идентификатор пользователя соответствует аннотации userId и так далее.

### МОДУЛЬ СБОРА ИНФОРМАЦИИ С КЛАВИАТУРЫ

Модуль предназначен для непосредственного сбора информации о клавиатурном вводе пользователя, вся полученная информация обрабатывается в соответствии с «белым» списком приложений (Notepad++, Microsoft Word, LibreOffice Writer) и накапливается в модуле накопления и хранения текущей информации. Данный модуль базируется на библиотеках ядра операционной системы Windows:

библиотека kernel32.dll для установки системного клавиатурного хука и получения идентификаторов активных процессов;

библиотека user32.dll для получения заголовков окон активных приложений, получения текущей раскладки клавиатуры и параметров нажатых клавиш;

библиотека psapi.dll для получения имен процессов и путей до их исполняемых файлов.

### МОДУЛЬ РАБОТЫ С ПОЛЬЗОВАТЕЛЬСКИМИ ШАБЛОНАМИ

Модуль работы с пользовательскими шаблонами содержит в себе два подмодуля:

модуль маршализации и записи статистики на диск. Данный модуль занимается представлением информации о текущем пользователе в виде маршализованной информации и последующей её записи на диск;

модуль чтения и демаршализации информации с диска. Данный модуль занимается чтением пользовательской информации с диска и последующим её сохранением в модуле накопления.

Для создания маршализованной информации использовалась стандартная библиотека Java – JAXB (Java Architecture for XML Binding) [7, 8]. Данная библиотека позволяет осуществлять простую конверсию Java-объектов в маршализованные данные и наоборот используя специальные аннотации.

Для осуществления файлового ввода/вывода была использована стандартная библиотека Java – Java NIO.2 предоставляющая возможности асинхронного ввода/вывода, что позволило осуществить одновременное чтение/запись пользовательских данных из разных потоков в один файл [9].

### МОДУЛЬ ОБРАБОТКИ ПОЛЬЗОВАТЕЛЬСКОГО ВВОДА

Модуль обработки пользовательского ввода содержит в себе два подмодуля:

модуль усреднения пользовательской статистики, данный модуль занимается обработкой «сырого» клавиатурного ввода, разделяя его на отдельные диграмы и передавая их в модуль накопления и хранения текущей информации.

модуль обработки усредненных значений, данный модуль занимается финальной обработкой усреднённой информации и формирования шаблона текущего пользователя, на анализе которого происходит определение пользователя.

### МОДУЛЬ НАКОПЛЕНИЯ И ХРАНЕНИЯ ТЕКУЩЕЙ ИНФОРМАЦИИ

Данный модуль представляет собой простой статический класс, содержащий несколько структур данных, необходимых для полноценной работы приложения, большинство из этих структур базируются на стандартной библиотеке Java – Concurrent API, для

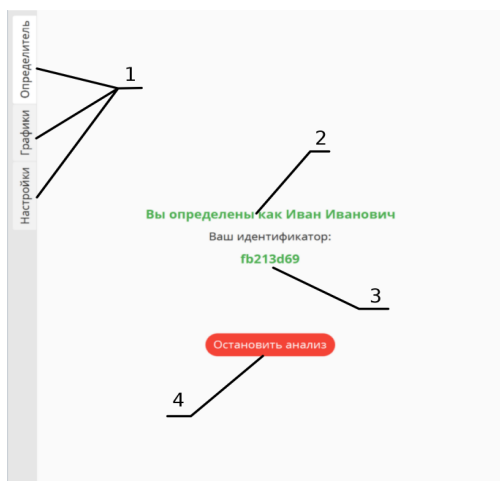
обеспечения возможности доступа из различных потоков приложения.

Данный модуль также осуществляет взаимодействие с пользовательским интерфейсом для отображения в процесса определения пользователя.

### ОСНОВНЫЕ ЭЛЕМЕНТЫ ИНТЕРФЕЙСА ПРОГРАММНОГО СРЕДСТВА ДЛЯ ОПРЕДЕЛЕНИЯ ПАРАМЕТРОВ ПОЛЬЗОВАТЕЛЯ ПЕРСОНАЛЬНОГО КОМПЬЮТЕРА

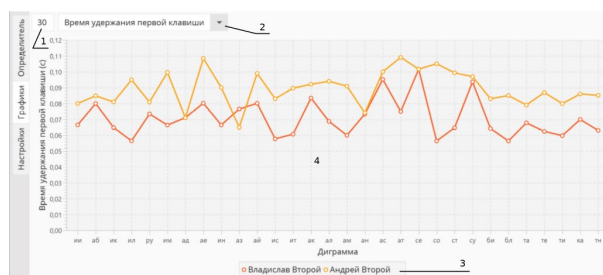
В ходе реализации интерфейса приложения было принято разделить его на несколько вкладок:

- 1) вкладка «*Определитель*», которая содержит информацию о текущем состоянии анализа клавиатурного почерка пользователя и возможностью его остановки (рисунок 5);
- 2) вкладка «*Графики*», служащая для отображения разницы между отдельными категориями шаблона пользователей в графическом виде, (рисунок 6);
- 3) вкладка «*Настройки*», позволяющая осуществлять базовую настройку программы и управлять пользователями (рисунок 7).



1 – переключатели вкладок; 2 – имя пользователя; 3 – идентификатор пользователя; 4 – кнопка остановки анализа  
Рис. 5. Вкладка «*Определитель*»

С помощью переключателя вкладок (поз. 1 на рисунке 5) происходит смена вкладок. В области «имя пользователя» (поз. 2 на рисунке 5) отображается имя пользователя или текущее состояние анализа если пользователь ещё не определён. Кнопка остановки анализа (поз. 4 на рисунке 5) служит для остановки процесса анализа пользователя, а также очистки всех внутренних полей в модуле хранения и накопления текущей информации. Ещё одно нажатие на данную кнопку запустит процесс анализа вновь.



1 – числовое поле для выбора количества отображаемых значений; 2 – выпадающий список; 3 – имена пользователей; 4 – линейная диаграмма

Рис. 6. Вкладка «*Графики*»

С помощью текстового поля (поз. 1 на рисунке 6) производится выбор количества отображаемых точек на графике. С помощью выпадающего списка (поз. 2 на рисунке 6) производится выбор отображаемого параметра клавиатурного почерка. В области «имена пользователей» (поз. 3 на рисунке 6) отображается легенда – имена пользователей, чьи параметры клавиатурного почерка сейчас отображаются. Линейная диаграмма (поз. 4 на рисунке 6) служит для отображения графической информации о параметрах клавиатурного почерка.

Как видно из графика, статистика собранная программой по пользователям достаточно сильно различается.



1 – комбинированный список; 2 – кнопка подтверждения выбора  
Рис. 7. Вкладка «*Настройки*»

С помощью комбинированного списка (поз. 1 на рисунке 7) производится ввод нового имени пользователя или выбор уже существующего. Кнопка подтверждения выбора (поз. 2 на рисунке 7) служит для принятия выбранного пользователя в качестве текущего.

### ЗАКЛЮЧЕНИЕ

В процессе выполнения выпускной квалификационной работы было разработано и реализовано программное средство для определения параметров клавиатурного почерка пользователя персонального компьютера.

Также были решены следующие частные задачи:

изучены особенности и способы определения клавиатурного почерка пользователей, такие как статистический метод и методы на базе весовых коэффициентов и нейронных сетей;

разработана структура программы для определения параметров клавиатурного почерка пользователя

разработан формат хранения пользовательских данных – маршализованная информация в виде XML-файлов и алгоритм работы программы;

реализована программа для определения параметров клавиатурного почерка пользователя, позволяющая определять основные параметры, создавать эталонные шаблоны клавиатурного почерка пользователя, а также производить его идентификацию с использованием этих шаблонов.

#### СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Identity Theft, Computers and Behavioral Biometrics // Ben Gurion University (BGU) of the Negev [Электронный ресурс]. – Режим доступа: <http://www.ise.bgu.ac.il/faculty/liorr/idth.pdf>

2. Rick Joyce and Gopal Gupta – Identity Authentication Based on Keystroke Latencies // Carnegie Mellon's University [Электронный ресурс]. – Режим доступа: <http://www.cs.cmu.edu/~maxion/courses/JoyceGupta90.pdf>

3. Fabian Monrose – Keystroke dynamics as a biometric for authentication // Columbia University in the city of New York [Электронный ресурс]. – Режим доступа: <http://www.cs.columbia.edu/4180/hw/keystroke.pdf>

4. Яндиев И. Б. Исследование временных характеристик клавиатурного почерка для быстрой аутентификации личности // Молодой ученый. — 2017. — №14. — С. 154-158. — URL <https://moluch.ru/archive/148/41543/> (дата обращения: 25.04.2018).

5. Richard L. Halterman – Fundamentals of Programming C++ // Online programming books [Электронный ресурс]. – Режим доступа: <https://www.onlineprogrammingbooks.com/free-download-fundamentals-of-programming-c/>

6. Extensible Markup Language (XML) 1.0 (Fifth Edition) // W3C Organization [Электронный ресурс]. – Режим доступа: <https://www.w3.org/TR/REC-xml/>

7. JSR 222: Java™ Architecture for XML Binding (JAXB) 2.0 // Java community process [Электронный ресурс]. – Режим доступа: <https://jcp.org/en/jsr/detail?id=222>

8. Using JAXB Data Binding // Oracle documentation [Электронный ресурс]. – Режим доступа: [https://docs.oracle.com/cd/E12840\\_01/wls/docs103/webserv/data\\_types.html#wp223908](https://docs.oracle.com/cd/E12840_01/wls/docs103/webserv/data_types.html#wp223908)

9. Package java.nio // Oracle documentation [Электронный ресурс]. – Режим доступа: <https://docs.oracle.com/javase/7/docs/api/java/nio/package-summary.html>

# DEVELOPMENT AND IMPLEMENTATION OF A PROGRAM FOR THE BIOMETRICS OF A PERSONAL COMPUTER USER BASED ON THE DEFINITION OF KEYSTROKE DYNAMICS PARAMETERS

**A.A. Vyazigin, N.Y. Tupikina, E.V. Sypin**

*Biysk Technological Institute (branch) of the Altay State Technical University, Biysk, Russia*

**Abstract** – This article describes the development and implementation of a software tool for determining the parameters of the keystroke dynamics, the functional and technical requirements for the software tool for determining the parameters of the keystroke dynamics, the methods for determining the user's keystroke dynamics, and ways to identify the user using it. A program was developed for a personal computer to determine the keystroke dynamics signature and identify it.

*Index terms:* biometrics, identification, keystroke dynamics, programming, algorithmization.

## REFERENCES

1. Identity Theft, Computers and Behavioral Biometrics // Ben Gurion University (BGU) of the Negev. Available at: <http://www.ise.bgu.ac.il/faculty/liorr/idth.pdf>
2. Rick Joyce and Gopal Gupta – Identity Authentication Based on Keystroke Latencies // Carnegie Mellon's University. Available at: <http://www.cs.cmu.edu/~maxion/courses/JoyceGupta90.pdf>
3. Fabian Monrose – Keystroke dynamics as a biometric for authentication // Columbia University in the city of New York Available at: <http://www.cs.columbia.edu/4180/hw/keystroke.pdf>
4. Yandiev I.B. Investigation of keystroke latencies for fast authentication of personality // Young scientist. — 2017. — №14. — p. 154-158. — Available at: <https://moluch.ru/archive/148/41543/> (accessed 25 April 2018).
5. Richard L. Halterman – Fundamentals of Programming C++ // Online programming books. Available at: <https://www.onlineprogrammingbooks.com/free-download-fundamentals-of-programming-c/>
6. Extensible Markup Language (XML) 1.0 (Fifth Edition) // W3C Organization. Available at: <https://www.w3.org/TR/REC-xml/>
7. JSR 222: Java™ Architecture for XML Binding (JAXB) 2.0 // Java community process. Available at: <https://jcp.org/en/jsr/detail?id=222>
8. Using JAXB Data Binding // Oracle documentation. Available at: [https://docs.oracle.com/cd/E12840\\_01/wls/docs103/websevr/data\\_types.html#wp223908](https://docs.oracle.com/cd/E12840_01/wls/docs103/websevr/data_types.html#wp223908)
9. Package java.nio // Oracle documentation. Available at: <https://docs.oracle.com/javase/7/docs/api/java/nio/package-summary.html>

*Vyazigin Andrey Aleksandrovich – is a student of a magistracy of “Methods and means of measurement and automation” department on the branch of «Information systems and technologies», Biysk Technological Institute, e-mail: [andrey.vyazigin@gmail.com](mailto:andrey.vyazigin@gmail.com).*

*Tupikina Nadezhda Yurievna – is an applicant of department “Methods and means of measuring and automation”, Biysk Technological Institute, 89635266318, e-mail: [tnu@bti.secna.ru](mailto:tnu@bti.secna.ru).*

*Sypin Eugene Victorovich – is an associate professor at the Chair of Methods and Means of Measurement and Automation, Candidate of Technical Science, Biysk Technological Institute, 89039499975, e-mail: [sev@bti.secna.ru](mailto:sev@bti.secna.ru).*