

УДК 519.688

UDC 519.688

05.13.18 Математическое моделирование, численные методы и комплексы программ (технические науки)

05.13.18 Mathematical modeling, numerical methods and software packages (technical sciences)

О ТЕХНОЛОГИЯХ РАЗРАБОТКИ ПРОГРАММНЫХ ПРИЛОЖЕНИЙ ДЛЯ ГЕНЕРИРОВАНИЯ И ПРОВЕРКИ ПРАКТИЧЕСКИХ ЗАДАНИЙ ПО МАТЕМАТИЧЕСКИМ ДИСЦИПЛИНАМ

ABOUT TECHNOLOGIES FOR DEVELOPING SOFTWARE APPLICATIONS FOR THE GENERATION AND VERIFICATION OF PRACTICAL TASKS IN MATHEMATICAL DISCIPLINES

Лаптев Владимир Николаевич
к.т.н., доцент
ФГБОУ «Кубанский государственный аграрный университет», 350044, Россия, г. Краснодар, ул. Калинина 13, E-mail: mail@kubsau.ru

Laptev Vladimir Nikolaevich
Cand.Tech.Sci., associate professor
Kuban State Agricultural university, 350044, Russia, Krasnodar, Kalinin st., 13, E-mail: mail@kubsau.ru

Михайленко Евгений Владимирович
к.ф.-м.н., заместитель начальника кафедры
ФГКОУ ВПО «Краснодарский университет МВД РФ», 350005, Россия, г. Краснодар, ул. Ярославская, 128, . E-mail: post@krdu-mvd.ru

Mikhaylenko Evgeny Vladimirovich
Cand.Phys.-Math.Sci., Deputy Head of a Department
Krasnodar University of Ministry of the Interior of Russian Federation, 350005 Russia, Krasnodar, Yaroslavl st. 128. E-mail: post@krdu-mvd.ru

В статье рассматриваются общие принципы разработки программных приложений для генерирования и проверки практических заданий по математическим дисциплинам в среде разработки Visual Basic for Application. Описывается структура компьютерных программ, исследуются основные компоненты реализуемых макросов, приводятся алгоритмы основных процедур. Особое место уделяется проблемам идентификации обучающихся и защите генерируемых данных. Даны рекомендации по использованию разрабатываемых программ в учебном процессе

This article discusses the general principles of developing software applications for generating and verifying practical tasks in mathematical disciplines in the Visual Basic for Application development environment. The article describes the structure of computer programs, the main components of the implemented macros are also investigated, the algorithms of the basic procedures are given. A special place is given to the problems of identifying students and protecting the generated data. We have also given recommendations on the use of developed programs in the educational process

Ключевые слова: АВТОМАТИЗАЦИЯ, МАТЕМАТИКА, АЛГОРИТМ, ГЕНЕРАЦИЯ, ВЕРИФИКАЦИЯ, МАКРОС

Keywords: AUTOMATION, MATHEMATICS, ALGORITHM, GENERATION, VERIFICATION, MACRO

DOI: <http://dx.doi.org/10.21515/1990-4665-156-008>

В настоящее время широко внедряются методы и технологии оптимизации образовательного процесса с использованием различного рода программных продуктов. Прежде всего, это использование текстовых редакторов и электронных таблиц для подготовки методического материала, компьютерных приложений для создания и показа презентаций, применение специализированных пакетов программ математических и инженерных вычислений, а также систем автоматизированного проектирования. Большое место в используемом программном

обеспечении занимают контролирующие системы для оценивания знаний обучаемых [1].


Использование вышеперечисленных программных продуктов эффективно влияет на скорость и качество подготовки учебного материала, активизирует процессы восприятия и усвоения представляемой информации обучаемыми, формирует устойчивые навыки расчета требуемых значений в инженерных и экономических исследованиях, позволяет производить качественную оценку знаний обучаемых с использованием основных стандартных форм тестовых заданий [4].

Важное место в процессе изучения математических и естественно-научных дисциплин обучающимися занимает подготовка заданий на практические занятия и самостоятельную работу [2]. Традиционно для этих целей преподавателями подбираются задачи из общеизвестных либо самостоятельно написанных учебников, учебных пособий, сборников задач и других учебно-методических изданий. Однако эффективность использования такого рода материалов крайне мала. Необходимо отметить существенную ограниченность количества типовых задач в изданиях. Преподаватель вынужден выдавать на самостоятельную подготовку общий список задач для всех обучаемых или в лучшем случае задания по нескольким вариантам [5].

В связи с этим обучающиеся теряют мотивацию самостоятельно выполнять общие «для всех» задания. Часто происходит простое копирование выполненных работ у добросовестных однокурсников или даже переписывание решений прошлых лет. Оценить качество работ, выполненных студентами во внеаудиторное время, в лучшем случае совместными усилиями всей учебной группы, просто невозможно [7].

Настоящая статья посвящена основным принципам разработки компьютерных приложений для генерирования практических заданий по изучаемым дисциплинам, особенностям использования некоторых методических приемов и технологий при разработке компьютерных программ для подготовки практических заданий и анализа их выполнения обучаемыми.

Суть разрабатываемых проектов заключается в использовании приложений интегрированного пакета Microsoft Office, где на заранее подготовленную форму автоматически по специальному алгоритму выводятся сгенерированные данные в виде комбинации чисел, символов или графических объектов (рис. 1).



Теория игр и принятия решений
Матричные игры
Практическое задание
Решение матричных игр
Вариант

Дата _____
Время _____

(Фамилия И.О.)

1. Для матричных игр 2x2, заданных платежными матрицами, определить смешанные стратегии игроков S_A и S_B , а также цену игры v . по 1 баллу за пункт задания

а) $\left(\begin{array}{c} \\ \end{array} \right)$ $v =$

$S_A =$

$S_B =$

б) $\left(\begin{array}{c} \\ \end{array} \right)$ $v =$

$S_A =$

$S_B =$

в) $\left(\begin{array}{c} \\ \end{array} \right)$ $v =$

$S_A =$

$S_B =$

г) $\left(\begin{array}{c} \\ \end{array} \right)$ $v =$

$S_A =$

$S_B =$

2. Упростить платежные матрицы игр и поместить полученный результат в левый верхний угол представленной формы. по 1 баллу за пункт задания

а) $\left(\begin{array}{c} \\ \end{array} \right)$

б) $\left(\begin{array}{c} \\ \end{array} \right)$

в) $\left(\begin{array}{c} \\ \end{array} \right)$

г) $\left(\begin{array}{c} \\ \end{array} \right)$

3. Определить смешанные стратегии игроков S_A и S_B , а также цену игры v . по 2 балла за пункт задания

а) $\left(\begin{array}{c} \\ \end{array} \right)$ $v =$

$S_A =$

$S_B =$

б) $\left(\begin{array}{c} \\ \end{array} \right)$ $v =$

$S_A =$

$S_B =$

в) $\left(\begin{array}{c} \\ \end{array} \right)$ $v =$

$S_A =$

$S_B =$

Критерий оценки:
13-14 баллов - "отлично"
11-12 баллов - "хорошо"
9-10 баллов - "удовлетворительно"
менее 9 баллов - "неудовлетворительно"

Количество набранных баллов: _____

Оценка за работу: _____

Подпись преподавателя _____

Проверка

Рис. 1. Форма практического задания «Решение матричных игр» по теме «Теория игр и принятия решений» дисциплины Прикладная математика.

Выбор программного продукта для генерации заданий не случаен. При детальном исследовании технических, финансовых, правовых возможностей разработчиков и пользователей выбор программного обеспечения для реализации идеи пал на общедоступный пакет программ Microsoft Office. Нас, как разработчиков, привлек мощный математический аппарат среды программирования Visual Basic for Application (VBA), доступность широких возможностей использования различных шрифтов, графических форм и объектов, простота и логичность синтаксиса и семантики языка программирования [3, 6].

Не менее важным является доступность пакета программ Microsoft Office. Удобство использования программного продукта заключается также и в том, что перечень заданий, выведенный в виде документа Word или Excel, не требует разработки специальных печатных форм [10]. В соответствии с выбором среды все программные продукты, реализующие генерацию и проверку выполненных заданий, написаны в виде макросов среды VBA, являющейся основным инструментарием пакета Microsoft Office.

Заполненную таким образом форму можно донести до обучающегося несколькими способами: для проведения аудиторных занятий поместить файл с заданиями на доступное для курсантов дисковое пространство, например, в виртуальный учебно-методический кабинет кафедры информатики и математики, размещенный на сервере компьютерного центра университета, организовав самостоятельную подготовку отправить шаблон с макросами для генерации курсантам адресно по электронной почте или предоставить доступ к заданиям в «облачном» пространстве.

Получив одним из вышеперечисленных способов файл с шаблоном, обучаемый с помощью находящихся в документе макросов генерирует набор типовых заданий, решает предложенные задачи, заполняет ответами выделенные поля, сохраняет выполненную работу в своей папке на сервере или отправляет документ преподавателю для проверки [9].

Преподаватель, используя макрос проверки осуществляет компьютерную проверку выполненных заданий. Во время проверки автоматически проверяются все заполненные курсантом поля, зачеркиваются неверные ответы, подсчитываются набранные баллы и согласно определенным критериям выставляется оценка. Следует отметить, что для организации аудиторных занятий или самостоятельной подготовки макрос проверки лучше размещать в отдельном файле, в недоступном для обучаемых месте. В противном случае обучаемые могут воспользоваться проверкой неоднократно, исправляя предварительно сохраненное выполненное задание.

Для осуществления генерации заданий курсанту достаточно заполнить поле «Фамилия и инициалы» (рис. 2) и запустить макрос генерации, нажав на графический объект, для которого назначен макрос запуска. В приведенном примере макрос генерации привязан к знаку университета, расположенном в левом верхнем углу шаблона.

Комбинаторика
Практическое задание
Конфигурации с повторениями

Дата _____
Время _____

(Фамилия И.О.)

Вариант

1. Вычислить

	—	=		—	=	
	—	=		—	=	

Рис. 2. Верхняя часть шаблона задания для генерации.

Введенные пользователем данные, а также дата и время запуска генерации, автоматически выводимые на лист, используются для расчета номера варианта. В дальнейшем, при автоматизированной проверке работы, производится верификация соответствия номера варианта указанным в работе времени и фамилии курсанта. Данная процедура позволяет избежать подлогов выполненных работ, и при нестыковке всех трех компонентов: варианта, времени и записи, идентифицирующей обучаемого, проверка работы не проводится.

На рисунке 3 представлен начальный фрагмент кода типового макроса «Делимость чисел», предлагаемого обучаемым для генерации заданий на практическом задании по теме «Теория чисел» дисциплины математика. Рассмотрим подробно представленный набор инструкций.

```
Private Sub deviz_num()  
  
    'Теория чисел  
    'Делимость чисел  
  
    PAROL_1 = "GJPBNBD"  
    ActiveSheet.Protect Password:=PAROL_1, _  
        UserInterfaceOnly:=True, AllowFormattingCells:=True  
    'ActiveSheet.EnableSelection = xlUnlockedCells  
    'ActiveSheet.Unprotect Password:=PAROL_1  
  
    If Cells(6, 9) <> "" Then  
        Cells(4, 4) = "ЗАДАНИЕ ЗАПУЩЕНО ПОВТОРНО!"  
        ActiveSheet.Protect Password:=PAROL_1  
    End  
End If  
  
Name = Cells(3, 14)  
If codeName(Name) = 0 Then  
    MsgBox ("Фамилия вводится только русскими буквами")  
    ActiveSheet.Protect Password:=PAROL_1  
End  
End If  
Range("A1:Z7").Locked = True  
  
t = Timer  
Cells(2, 15) = sTimer(t)  
s = Cells(2, 15)  
Cells(1, 15) = Date  
  
v = t Mod 31000 + codeName(Name)  
Cells(6, 9) = v  
Rnd (v * (-1))  
Randomize (v)
```

Рис. 3. Начальный фрагмент типового макроса для генерации заданий.

После объявления имени макроса *deviz_num*, к которому привязан графический объект на листе MS Excel для запуска основной подпрограммы, и комментариев, отображающих тему и наименование практического задания, мы видим шесть блоков инструкций, разделенных пустыми строками.

Первый блок необходим для установления и снятия защиты листа с заданиями. Здесь переменной *PAROL_1* присваивается соответствующее значение ключевого слова "GJPBNBD". Инструкция *ActiveSheet.Protect* во

второй строке рассматриваемого блока программы используется преподавателем в процессе разработки модуля для защиты высылаемой формы до выполнения генерации заданий курсантами. Она также необходима во избежание конфликтов при снятии защиты во время генерации заданий обучаемым, если лист с формой по какой-то причине не был ранее защищен.

Процедура *ActiveSheet.Protect* будет автоматически запускаться при открытии файла и устанавливать защиту на заданные листы книги. Дополнительно введенный параметр *UserInterfaceOnly*, указывает, что защита распространяется только на операции пользователя и не будет препятствовать макросу выполнять предписанные алгоритмом действия, а параметр *AllowFormattingCells* предоставляет пользователю возможность не только вносить данные в заблокированные ячейки, но и форматировать их, меняя, например, цвет ячейки или текста, или использовать надстрочные или подстрочные символы. Если на защищенном листе обучаемому нужно пользоваться группировкой, то необходимо включить в код программы метод *EnableOutlining*.

Для активного листа, с которым работает пользователь, устанавливается защита, и все ячейки листа, помеченные преподавателем, как «защищаемая ячейка» будут недоступны для редактирования обучаемыми. На форме, предлагаемой курсанту для генерации, все ячейки, кроме поля «Фамилия И.О.» должны быть защищены, однако, после генерации заданий необходимо защитить это поле, предоставив доступ обучающегося только к полям для ввода готовых решений.

Для того, чтобы выполнялась защита листа, но не осуществилась генерация заданий, разработчику достаточно запустить макрос с установленной «точкой останова» на первой строке второго блока, а затем прекратить выполнение макроса.

Закомментированные инструкции третьей и четвертой строки рассматриваемого блока программы предназначены для временного снятия

ранее установленной защиты листа разработчиком. Для редактирования формы разработчику часто бывает необходимо вносить символьные и числовые данные в ячейки листа заданий, изменять шрифт, размер, начертание, цвет и подчеркивание текста, определять верхние или нижние индексы в редактируемых записях, устанавливать выравнивание, границы, заливку и защиту обрабатываемых ячеек, выводить графические объекты и изменять некоторые параметры рабочей книги. После редактирования листа следует снова защитить лист с использованием инструкции *ActiveSheet.Protect* с параметрами *UserInterfaceOnly* и *AllowFormattingCells*.

Следует отметить, что используемый метод *Protect* объекта *ActiveSheet* аналогичен защите листа вручную (вкладка «Рецензирование», группа «Изменения», кнопка «Защитить лист»). Однако, даже зная пароль, снять защиту с листа без использования макросов невозможно.

Кроме защиты листа разработчику программного продукта также необходимо ограничить доступ к самим исполняемым макросам. Стандартные методы защиты проекта VBA не являются эффективными. Для продвинутых недобросовестных пользователей, особенно если работа с заданиями ведется на самоподготовке без участия преподавателей, найти в интернете средство для «взлома» не представляет труда. Однако, смысла в доступе к коду программы нет, так как в нем не только не прописаны ответы или алгоритмы решения заданий, но и нет никаких подсказок по их решению. Все же для надежности защиты можно использовать сторонние общедоступные приложения.

Второй блок фрагмента макроса используется для предотвращения повторного запуска программы, например, для выбора обучаемыми нужного варианта. Здесь проверяется наличие номера генерируемого варианта: если поле, предназначенное для вывода варианта пусто, то на лист выводится запись «ЗАДАНИЕ ЗАПУЩЕНО ПОВТОРНО!», производится за-

щита листа и выполнение программы прекращается. В противном случае продолжается выполнение программы.

В третьем блоке рассматриваемого фрагмента считывается значение поля «Фамилия И.О.». Затем это значение обрабатывается функцией codeName. На рисунке 4 представлен код данной процедуры.

```
Function codeName(Name)
k1 = 0
codeName = 0
For i = 1 To Len(Name)
    k = ((Asc(Mid(Name, i, 1)) - 191) Mod 32) * i
    If k > 0 Then
        codeName = (codeName + k) Mod 1000
    Else
        Select Case Mid(Name, i, 1)
            Case ".", " ", "-" : k1 = k1
            Case Else: k1 = 1
        End Select
    End If
Next i
If k1 = 1 Then codeName = 0
End Function
```

Рис. 4. Процедура codeName.

В зависимости от введенной фамилии и инициалов представленный алгоритм формирует числовое значение в диапазоне от 0 до 999, которое в дальнейшем будет использовано в расчетах номера варианта. В случае, если по ошибке обучаемый ввел в поле хотя бы один символ отличный от заглавных или строчных букв, русского алфавита, точки, пробела или дефиса, то расчетное значение функции codeName будет равно нулю, на экране появится информационное сообщение о некорректном вводе (рис. 5), выполнение программы прекращается и после исправления ошибки пользователь сможет запустить макрос генерации заданий еще раз.

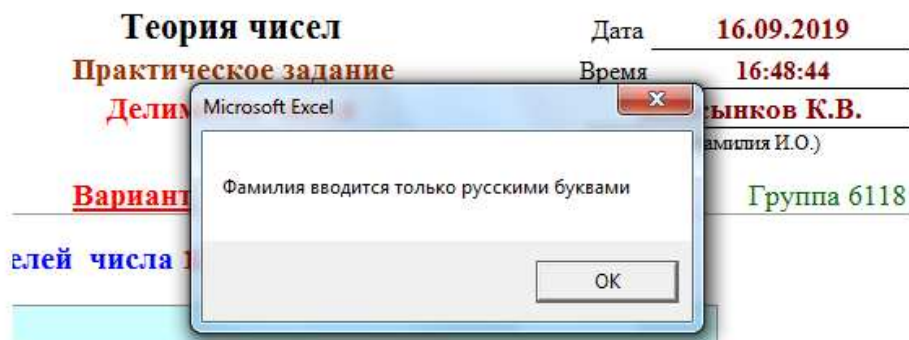


Рис. 5. Сообщение о некорректном вводе идентификатора пользователя.

В случае успешного ввода идентификатора пользователя устанавливается защита на все ячейки первых семи строк заполняемого шаблона и выполняются инструкции четвертого блока рассматриваемого фрагмента программы, в котором с использованием встроенных функций *Timer* и *Date* определяются и наносятся на лист время и дата запуска макроса генерации. Полученное значение переменной t представляет собой количество сотых долей секунд, прошедших с начала текущих суток до момента вызова функции *Timer*. С кодом функции *sTimer* можно ознакомиться в приложении к методическим рекомендациям.

В пятом блоке программы рассчитывается и записывается номер варианта. Как видно из листинга макроса, номер варианта формируется в диапазоне от 1 до 32000 и зависит только от времени запуска программы и кода введенной фамилии. Здесь же с участием номера варианта происходит инициализация генератора случайных чисел *Rnd*, определяется начальное значение, которое будет использоваться для расчета параметров генерируемых заданий. Завершающий блок начального фрагмента типового макроса используется только для объявления переменных, используемых в коде генерации.

Генерация параметров каждого задания в предлагаемых работах осуществляется по разработанным алгоритмам, учитывающим номер варианта и начального значения генератора случайных чисел.

Для верификации, выложенной на проверку работы, преподавателю необходимо открыть ее с помощью соответствующего приложения Microsoft Office и нажать на кнопку «Проверка», находящуюся в правом нижнем углу выполненной работы (рис. 6). Данная кнопка привязана к макросу, предназначение которого заключается в открытии файла проверки, запуска соответствующего модуля проверки и закрытии файла проверки.

<p>Критерий оценки:</p> <p>31-33 баллов - "отлично"</p> <p>26-30 баллов - "хорошо"</p> <p>22-25 баллов - "удовлетворительно"</p> <p>менее 22 баллов - "неудовлетворительно"</p>	<p>Количество набранных баллов: _____</p> <p>Оценка за работу: _____</p>
	<div style="border: 1px solid black; padding: 5px; display: inline-block; background-color: #f0f0f0;"> <p>Проверка</p> </div>

Рис. 6. Кнопка запуска макроса проверки.

На рисунке 7 размещен программный код привязанного макроса запуска проверки.

```

Private Sub proverka(path, file_name, sub_name)
Ful_name = path + "\" + file_name
If Dir(Ful_name) <> "" Then
    Application.Run "" + Ful_name + "!" + sub_name
    Workbooks(file_name).Close
End
End If
End Sub

```

```

Private Sub pz_pr()
'Определяем папку, файл-проверки, программу
dir_name = "Комбинаторика"
file_name = "kombinatorika_s_povt_2_pr.xls"
sub_name = "kombinatorika2_pr"
'Программа завершается, если работа уже проверена
If Cells(44, 15) <> "" Then
    MsgBox "Работа уже проверена!", Buttons:=vbCritical
End
End If

'Проверяем из R:\ghosts\teachers\Proverka
path = "R:\ghosts\teachers\Proverka\" + dir_name
Call proverka(path, file_name, sub_name)
MsgBox "У вас нет прав для осуществления проверки!", Buttons:=vbCritical
End Sub

```

Рис. 7. Программный код процедуры вызова модуля проверки.

Из листинга видно, что данная процедура определяет полную спецификацию файла с модулем проверки, исследует представленное для проверки задание на наличие уже поставленной оценки, вызывает модуль проверки, а в случае его недоступности, если запуск осуществлен обучаемым, выдает сообщение «У вас нет прав для осуществления проверки!».

Модуль проверки выполненных работ следует размещать в отдельном файле, в месте недоступном для обучаемых. Это может быть жесткий диск на персональном компьютере преподавателя, USB-флеш-накопитель, специально выделенная, с правами доступа «на чтение» только для преподавателей, область сетевого диска или облачное пространство.

В программе проверки достаточно определить только вариант выполненной работы. Для идентификации обучаемого происходит проверка на соответствие варианта работы дате, времени и фамилии, указанных на листе. Затем по тем же алгоритмам, что и в генерации работы обучаемого, производится определение параметров каждого задания. Таким образом, нет необходимости сохранять уникальные данные условий заданий для каждого варианта, они снова генерируются каждый раз при проверке работы [8].

Проверка всех представленных заданий заканчивается автоматическим выставлением оценки за выполненную работу. Рассматривая инструкции кода макроса (рис. 8), отметим, что сначала программа выводит на проверяемый лист количество набранных баллов, в зависимости от значения переменной *mark* рассчитывается и отображается оценка за выполненную работу, происходит защита листа с параметрами, установленными ранее.

```
Cells(56, 15) = mark
Select Case mark
  Case 31 To 33
    osenka = "отлично"
  Case 26 To 30
    osenka = "хорошо"
  Case 22 To 25
    osenka = "удовлетворительно"
  Case Else
    osenka = "неудовлетворительно"
End Select

If Cells(57, 13) = "" Then
  Cells(57, 13) = osenka
Else
  Cells(57, 13) = "Повторная проверка!"
End If

ActiveSheet.Protect Password:=PAROL_1
End Sub
```

Рис. 8. Фрагмент программного кода выставления оценки.

Таким образом, мы рассмотрели общие принципы разработки компьютерных приложений для генерирования и верификации практических заданий по изучаемым математическим и естественнонаучным дисциплинам. В работе дано описание структуры разрабатываемых программных продуктов, общие рекомендации по использованию обучающих программ в учебном процессе.

Программные продукты, представленные в данной статье, успешно используются в учебном процессе Кубанского государственного университета имени И.Т. Трубилина и Краснодарского университета МВД России при изучении дисциплин математика и информатика специалистами и курсантами, обучающимися по специальности 38.05.01 Экономическая безопасность, также курсантами, обучающимися по специальностям 10.05.05 Безопасность информационных технологий в правоохранительной сфере и 40.05.03 Судебная экспертиза.

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. Денисенко Л.Г. Тестовые материалы в условиях реализации ФГОС для учреждений профессионального образования (из опыта разработки). Часть 1. – Новосибирск: Новосибирский институт мониторинга и развития образования, 2014. – 110 с.

2. Лаптев, В.Н. Методы разработки тестовых заданий в автоматизированной контролирующей системе «Контроль» / В.Н. Лаптев, Е.В. Михайленко // Политематический сетевой электронный научный журнал Кубанского государственного аграрного университета (Научный журнал КубГАУ) [Электронный ресурс]. – Краснодар: КубГАУ, 2015. – №09(113). – С. 826 – 840. – IDA [article ID]: 1131509061. – Режим доступа: <http://ej.kubagro.ru/2015/09/pdf/61.pdf>, 0,938 у.п.л.

3. Лаптев, В.Н. Некоторые аспекты применения среды Visual Basic for Application для создания учебных приложений по математическим дисциплинам / В.Н. Лаптев, Е.В. Михайленко // Политематический сетевой электронный научный журнал Кубанского государственного аграрного университета (Научный журнал КубГАУ) [Электронный ресурс]. – Краснодар: КубГАУ, 2014. – №09(103). – С. 222 – 233. – IDA [article ID]: 1031409014. – Режим доступа: <http://ej.kubagro.ru/2014/09/pdf/14.pdf>, 0,75 у.п.л.

4. Лаптев, В.Н. Организация тестирования в автоматизированной контролирующей системе «Контроль» / В.Н. Лаптев, Е.В. Михайленко // Политематический сетевой электронный научный журнал Кубанского государственного аграрного университета (Научный журнал КубГАУ) [Электронный ресурс]. – Краснодар: КубГАУ, 2016. – №10(124). – С. 461 – 471. – IDA [article ID]: 1241610026. – Режим доступа: <http://ej.kubagro.ru/2016/10/pdf/26.pdf>, 0,688 у.п.л.

5. Михайленко, Е.В. Автоматизация подготовки заданий и проверки выполненных работ по математическим и естественнонаучным дисциплинам / Е.В. Михайленко // Проблемы информационного обеспечения деятельности правоохранительных органов: материалы международной научно-практической конференции. – Белгород: Белгородский юридический институт МВД России, 2015. – С.107 – 114.

6. Михайленко, Е.В. О разработке обучающих программ в среде Visual Basic for Application / Е.В. Михайленко // Математические методы и информационно-технические средства: материалы XII Всерос. Науч.-практ. Конф. 17 июня 2016 г. / редкол.: И.Н. Старостенко, Е.В. Михайленко, А.А. Хромых, М.В. Шарпан. – Краснодар: Краснодарский университет МВД России. 2016. – С. 179 – 184.

7. Михайленко, Е.В. Об автоматизации проведения контрольно-проверочных и итоговых занятий по определению уровня профессиональной подготовленности сотрудников органов внутренних дел к выполнению служебных задач / Е.В. Михайленко // Проблемы информационного обеспечения деятельности правоохранительных органов: сборник статей 3-й Международной научно-практической конференции (14 октября 2016 г.). – Белгород: Белгородский юридический институт МВД России имени И.Д. Путилина, 2017. – С. 185 – 193.

8. Михайленко, Е.В. Особенности разработки в VBA программных приложений для генерирования практических заданий и анализа их выполнения / Е.В. Михайленко // Математические методы и информационно-технические средства: материалы XI Всерос. науч.-практ. конф., 19 июня 2015 г. / редкол. И.Н. Старостенко (отв. ред.), Е.В. Михайленко, Ю.Н. Сопильняк, А.В. Еськов, М.В. Шарпан. – Краснодар: Краснодар. ун-т МВД России, 2015. – С. 159 – 162.

9. Михайленко, Е.В. Особенности разработки компьютерных программ для подготовки практических заданий и анализа их выполнения по разделам булевой алгебры / Е.В. Михайленко // Проблемы информационного обеспечения деятельности правоохранительных органов: материалы 2-й международной научно-практической конференции. – Белгород: Белгородский юридический институт МВД России имени И.Д. Путилина, 2016. – С. 268 – 274.

10. Михайленко, Е.В. Технологии разработки программных приложений для генерации и проверки выполнения практических заданий по математическим дисциплинам / Е.В. Михайленко // Математические методы и информационно-технические средства: материалы XIII Всерос. науч.-практ. конф. (16 июня 2017 г.) / редкол.: И.Н. Старостенко, Е.В. Михайленко; М.В. Шарпан, А.А. Хромых. – Краснодар: Краснодарский университет МВД России, 2017. – С. 192 – 199.

References

1. Denisenko L.G. Testovye materialy v usloviyah realizacii FGOS dlja uchrezhdenij professional'nogo obrazovaniya (iz opyta razrabotki). Chast' 1. – Novosibirsk: No-vosibirskij institut monitoringa i razvitija obrazovaniya, 2014. – 110 s.

2. Laptev, V.N. Metody razrabotki testovyh zadaniy v avtomatizirovannoj kontrolirujushhej sisteme «Kontrol'» / V.N. Laptev, E.V. Mihajlenko // Politematicheskij setevoj jelektronnyj nauchnyj zhurnal Kubanskogo gosudarstvennogo agrarnogo universi-teta (Nauchnyj zhurnal KubGAU) [Jelektronnyj resurs]. – Krasnodar: KubGAU, 2015. – №09(113). – S. 826 – 840. – IDA [article ID]: 1131509061. – Rezhim dostupa: <http://ej.kubagro.ru/2015/09/pdf/61.pdf>, 0,938 u.p.l.

3. Laptev, V.N. Nekotorye aspekty primeneniya sredy Visual Basic for Application dlja sozdaniya uchebnyh prilozhenij po matematicheskim disciplinam / V.N. Laptev, E.V. Mihajlenko // Politematicheskij setевой jelektronnyj nauchnyj zhurnal Kubanskogo gosudarstvennogo agrarnogo universiteta (Nauchnyj zhurnal KubGAU) [Jelektronnyj re-surs]. – Krasnodar: KubGAU, 2014. – №09(103). – S. 222 – 233. – IDA [article ID]: 1031409014. – Rezhim dostupa: <http://ej.kubagro.ru/2014/09/pdf/14.pdf>, 0,75 u.p.l.

4. Laptev, V.N. Organizacija testirovanija v avtomatizirovannoj kontrolirujushhej sisteme «Kontrol'» / V.N. Laptev, E.V. Mihajlenko // Politematicheskij setевой jelektronnyj nauchnyj zhurnal Kubanskogo gosudarstvennogo agrarnogo universiteta (Nauchnyj zhurnal KubGAU) [Jelektronnyj re-surs]. – Krasnodar: KubGAU, 2016. – №10(124). – S. 461 – 471. – IDA [article ID]: 1241610026. – Rezhim dostupa: <http://ej.kubagro.ru/2016/10/pdf/26.pdf>, 0,688 u.p.l.

5. Mihajlenko, E.V. Avtomatizacija podgotovki zadaniy i proverki vypolnennyh rabot po matematicheskim i estestvennonauchnym disciplinam / E.V. Mihajlenko // Problemy informacionnogo obespechenija dejatel'nosti pravoohranitel'nyh organov: materialy mezhdunarodnoj nauchno-prakticheskoj konferencii. – Belgorod: Belgorodskij juridicheskij institut MVD Rossii, 2015. – S.107 – 114.

6. Mihajlenko, E.V. O razrabotke obuchajushhih programm v srede Visual Basic for Application / E.V. Mihajlenko // Matematicheskie metody i informacionno-tehnicheskie sredstva: materialy XII Vseros. Nauch.-prakt. Konf. 17 ijunya 2016 g. / redkol.: I.N. Starostenko, E.V. Mihajlenko, A.A. Hromyh, M.V. Sharpan. – Krasnodar: Krasnodarskij universitet MVD Rossii. 2016. – S. 179 – 184.

7. Mihajlenko, E.V. Ob avtomatizacii provedeniya kontrol'no-proverochnyh i itogovyh zanjatij po opredeleniju urovnja professional'noj podgotovlennosti sotrudnikov organov vnutrennih del k vypolneniju sluzhebnyh zadach / E.V. Mihajlenko // Problemy informacionnogo obespechenija dejatel'nosti pravoohranitel'nyh organov: sbornik statej 3-j Mezhdunarodnoj nauchno-prakticheskoj konferencii (14 oktjabrja 2016 g.). – Belgorod: Belgorodskij juridicheskij institut MVD Rossii imeni I.D. Putilina, 2017. – S. 185 – 193.

8. Mihajlenko, E.V. Osobennosti razrabotki v VBA programmnyh prilozhenij dlja generirovanija prakticheskikh zadaniy i analiza ih vypolnenija / E.V. Mihajlenko // Matematicheskie metody i informacionno-tehnicheskie sredstva: materialy XI Vseros. nauch.-prakt. konf., 19 ijunya 2015 g. / redkol. I.N. Starostenko (otv. red.), E.V. Mihajlenko, Ju.N. Sopil'njak, A.V. Es'kov, M.V. Sharpan. – Krasnodar: Krasnodar. un-t MVD Rossii, 2015. – S. 159 – 162.

9. Mihajlenko, E.V. Osobennosti razrabotki komp'juternyh programm dlja podgotovki prakticheskikh zadaniy i analiza ih vypolnenija po razdelam bulevoj algebry / E.V. Mihajlenko // Problemy informacionnogo obespechenija dejatel'nosti pravoohranitel'nyh organov: materialy 2-j mezhdunarodnoj nauchno-prakticheskoj konferencii. – Belgorod: Belgorodskij juridicheskij institut MVD Rossii imeni I.D. Putilina, 2016. – S. 268 – 274.

10. Mihajlenko, E.V. Tehnologii razrabotki programmnyh prilozhenij dlja generacii i proverki vypolnenija prakticheskikh zadaniy po matematicheskim disciplinam / E.V. Mihajlenko // Matematicheskie metody i informacionno-tehnicheskie sredstva: materialy XIII Vseros. nauch.-prakt. konf. (16 ijunya 2017 g.) / redkol.: I.N. Starostenko, E.V. Mihajlenko; M.V. Sharpan, A.A. Hromyh. – Krasnodar: Krasnodarskij universitet MVD Rossii, 2017. – S. 192 – 199.