

УДК 519.688

UDC 519.688

05.13.10 - Управление в социальных и экономических системах (технические науки)

05.13.10 - Management in social and economic systems (technical sciences)

**ОСНОВНЫЕ ПРИНЦИПЫ ИСПОЛЬЗОВАНИЯ
ОФИСНЫХ ПРИЛОЖЕНИЙ ДЛЯ
ПОДГОТОВКИ И ВЕРИФИКАЦИИ
ЭЛЕКТРОННЫХ ПРАКТИЧЕСКИХ ЗАДАНИЙ**

**BASIC PRINCIPLES FOR USING OFFICE
APPLICATIONS TO PREPARE AND VERIFY
ELECTRONIC PRACTICES**

Лаптев Владимир Николаевич

к.т.н., доцент

E-mail: mail@kubsau.ru

ФГБОУ «Кубанский государственный аграрный университет», 350044, Россия, г. Краснодар, ул.Калинина 13

Laptev Vladimir Nikolaevich

Cand.Tech.Sci., associate professor

E-mail: mail@kubsau.ru

Kuban State Agricultural university, 350044, Russia, Krasnodar, Kalinina, 13

Михайленко Евгений Владимирович

к.ф.-м.н., заместитель начальника кафедры

E-mail: post@krdu-mvd.ru

ФГКОУ ВПО «Краснодарский университет МВД РФ», 350005, Россия, г. Краснодар, ул. Ярославская, 128

Mikhaylenko Evgeny Vladimirovich

Cand.Phys.-Math.Sci., Deputy Head of a Department

E-mail: post@krdu-mvd.ru

Krasnodar University of Ministry of the Interior of Russian Federation, 350005 Russia, Krasnodar, Yaroslavskaya, 128

Статья посвящена проблемам создания программных модулей в офисных приложениях для генерирования и верификации практических заданий по математическим дисциплинам. В работе определяются принципы разработки программных продуктов, алгоритмы и технологии разрабатываемых макросов, описывается структура компьютерных программ, исследуются основные компоненты реализуемых макросов в среде разработки Visual Basic for Application, приводятся алгоритмы основных процедур. Особое место уделяется проблемам идентификации обучающихся и защите генерируемых данных. Даны рекомендации по использованию разрабатываемых программ в учебном процессе для профессорско-преподавательского состава, инженеров-программистов образовательных организаций

The article is devoted to the problems of creating software modules in office applications for generating and verifying practical tasks in mathematical disciplines. The work defines the principles of software development, algorithms and technologies of the developed macros, describes the structure of computer programs, examines the main components of implemented macros in the Visual Basic for Application development environment, provides algorithms for the main procedures. Particular attention is paid to the problems of identifying students and protecting the generated data. We have also given recommendations on the use of the developed programs in the educational process for the teaching staff and software engineers of educational organizations

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

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

<http://dx.doi.org/10.21515/1990-4665-173-007>

Настоящая статья описывает процедуры подготовки программных продуктов для генерирования практических заданий по математическим дисциплинам и проверки работ, выполненных обучающимися.

Описываемые методы и технологии рассчитаны на использование разработчиками среды офисного пакета Microsoft Office, который широко

<http://ej.kubagro.ru/2021/09/pdf/07.pdf>

распространен в образовательной среде, однако все рассуждения, касающиеся методов и технологий создания указанных программных продуктов, легко могут быть перенесены на другие информационно совместимые программные пакеты, такие как, например, OpenOffice или LibreOffice [1]. Кроме того, разработанные в среде Microsoft Office программные продукты в большинстве случаев без потерь будут открываться и работать в указанных системах.

Разработчиками представленного программного обеспечения могут быть, как профессиональные инженеры-программисты технических служб образовательных организаций, так и преподаватели кафедр, ведущие занятия по математическим и информационно-техническим дисциплинам, являющиеся продвинутыми пользователями в области компьютерных технологий.

Суть разрабатываемых программных комплексов заключается в использовании приложений интегрированных пакетов Microsoft Office, OpenOffice или LibreOffice, где на заранее подготовленную форму в виде документа, электронной таблицы или презентации автоматически по специальному алгоритму, реализованному в виде макросов, выводятся сгенерированные данные в виде комбинации чисел, символов, строк, таблиц или графических объектов [2].

На рисунке 1 представлено сгенерированное индивидуальное задание программного комплекса «Дискретная математика», предназначенное для использования на итоговом практическом занятии по одноименному разделу дисциплины прикладная математика. Подготовленная форма расположена на листе электронной таблицы и содержит одиннадцать типовых заданий по темам: «Комбинаторика», «Теория чисел», «Сравнения», «Булева алгебра», «Графы» и «Теория автоматов».



Прикладная математика
Итоговое практическое задание
Дискретная математика

Дата 19.10.2021
Время 05:21:43
Образец
(Фамилия И.О.)

Вариант 19614

1. Решить комбинаторную задачу.

В учебной группе 6 юношей и 4 девушек. Сколькими способами можно выбрать 7 курсантов, таким образом чтобы среди них было 3 девушек?

2. Выписать все простые числа из интервала [797; 840].

Образец выполнения задания
23, 29, 31, 37, ...

3. Найти наибольший общий делитель и наименьшее общее кратное чисел.

$(1898, 1035) =$ $[1898, 1035] =$

4. Найти каноническое разложение факториала.

$21! =$

Образец выполнения задания
 $2^{26} * 3^{14} * 5^5 * 7^3 * 11^2 * 13^2 \dots$

5. Из промежутка [-21; 112] выписать все целые числа, сравнимые

а) с числом 13 по модулю 14; б) с числом -20 по модулю 11.

Образец выполнения задания
-20, -8, 4, 16, 28, 40, 52, 64, 76, 88

6. Выписать приведенную систему вычетов.

а) по модулю 11; б) по модулю 18.

7. Решить сравнения.

$8x \equiv -17 \pmod{7};$ $20x \equiv 12 \pmod{6};$ $15x \equiv 30 \pmod{10}.$
 $x \equiv$ $x \equiv$ $x \equiv$

Образец выполнения задания
 $x \equiv$ 2, 5, 8, 11, 14 (mod 15)

8. Найти СДНФ дизъюнкции и конъюнкции булевых функций.

$f_1(A,B,C) = \bar{A}B(\bar{C}+A\bar{B}C) + \bar{C}$ $f_2(A,B,C) = \bar{A}\bar{B} + (\bar{A}B + \bar{B})B$

	A		
B	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
C	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

$f_1 + f_2 =$
 $f_1 f_2 =$

Образец записи СДНФ
(0, 1, 3, 6)

9. Найти СКНФ булевой функции.

а) $f_3(A,B,C) = ABC(\bar{A}\bar{B}C + \bar{A}BC) + \bar{A}\bar{C}$

СДНФ $f_3^- =$ СКНФ $f_3 =$

Образец записи СКНФ
[1, 2, 4, 7]

10. Для ориентированного графа G построить матрицу смежности M, матрицу сильной связности W и матрицу достижимости T.

$G = \{(g, h, i, j, k), \{(g, j), (g, k), (h, g), (h, j), (i, g), (i, j), (j, g), (j, j), (k, k), (k, k)\}\}$

M=	<input type="text"/>	W=	<input type="text"/>	T=	<input type="text"/>
----	----------------------	----	----------------------	----	----------------------

11. Опишите результат функционирования представленного автомата.

f/g	s0	s1*	s2	t	0	1	2	3	4	5	6	7	8	9	10
B	s1/2	s2/1	s1/1	x(t)											
П	s2/1	s0/0	s0/1	y(t)											
T	s0/1	s1/0	s2/1	s(t)											

$x = В Т П Т П В П В Т$

Критерий оценки:
21-23 баллов - "отлично"
18-20 баллов - "хорошо"
13-17 баллов - "удовлетворительно"
менее 13 баллов - "неудовлетворительно"

Количество набранных баллов: _____

Оценка за работу: _____

Рис. 1. Сгенерированное индивидуальное задание программного комплекса «Дискретная математика».

Для проведения практических занятий подготовленные задания можно разместить на доступном для обучаемых дисковом пространстве на сервере, для организации самостоятельной подготовки разослать заполненные формы по электронной почте или предоставить доступ к заданиям в «облачном» пространстве [3].

Получив обратно файлы с выполненными заданиями, с помощью макроса проверки преподаватель запускает макрос проверки, который по специальному алгоритму проверяет все заполненные обучаемыми поля, перечеркивает неверные ответы и согласно установленным критериям выставляет оценки.

Весь программный комплекс для подготовки и верификации индивидуальных электронных практических заданий может содержаться в одном файле любого офисного приложения. Наиболее удобным для разработчиков является формат электронных таблиц.

Файл комплекса программ, в котором расположен программный комплекс содержит всего два листа: «Генерация» и «Шаблон». На рисунке 2 представлен фрагмент листа «Генерация». Как мы видим, на листе расположены два графических объекта с надписями «Генерация» и «Проверить работу», поля, описывающие параметры заданий: «Имя листа», «Имя файла», «Метка», «Макрос проверки». Кроме того, на листе представлен номер учебной группы и список курсантов, в ней обучающихся.

	1	2	3	4
1			6119	
2	Генерация	1	Берест Елизавета	
3		2	Блягоз Артем	
4		3	Ванчугов Максим	
5		4	Гребнева Виолетта	
6		5	Григолия Дмитрий	
7	Имя листа	6	Демеденко Никита	
8	Дискретная математика	7	Егоров Егор	
9	Имя файла	8	Ибрагимов Айдемир	
10	ММ_Т7П32-	9	Кириленко Никита	
11		10	Ковган Максим	
12	Метка	11	Курочкина Диана	
13		12	Петраковская Ангелина	
14		13	Радченко Николай	
15		14	Смирнов Владислав	
16		15	Смирнов Юрий	
17		16	Суздалев Семен	
18		17	Урусбиев Руслан	
19		18	Хидриев Хайбула	
20		19	Шаваев Борис	
21		20	Шаршина Ольга	
22			Образец	
23	Проверить работу			
24				
25				

Рис. 2. Фрагмент листа «Генерация» комплекса «Дискретная математика».

На втором листе рабочей книги расположен незаполненный шаблон задания, копия которого во время подготовки индивидуального задания будет заполняться специальным образом сгенерированными данными. Наименование листа непосредственно связано с генерируемыми заданиями. На рисунке 3 представлена верхняя часть шаблона заданий, расположенного на листе «Дискретная_математика».

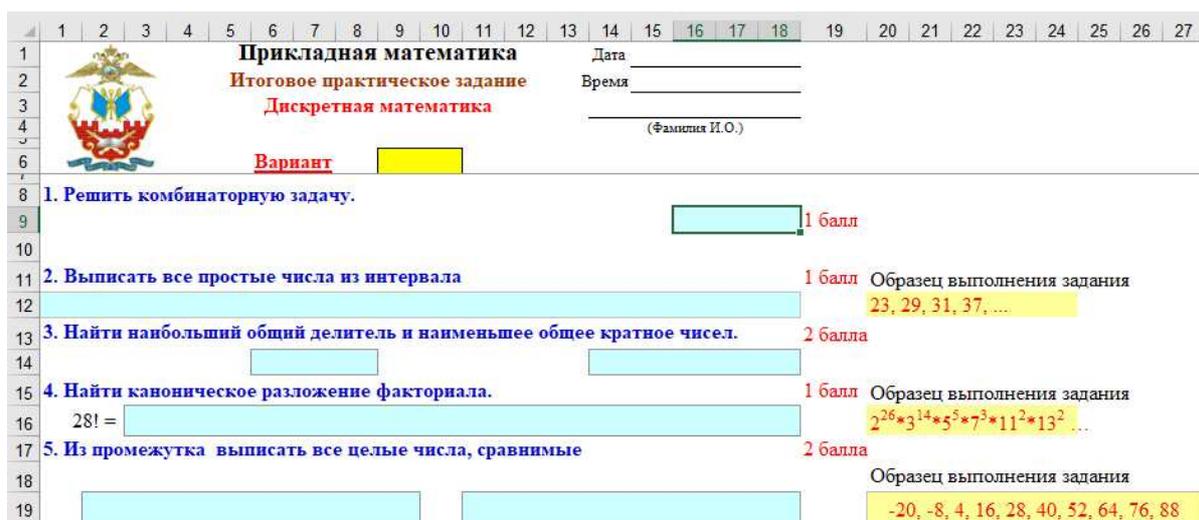


Рис. 3. Фрагмент листа «Дискретная_математика».

Кроме того, программный комплекс содержит четыре программных модуля для генерирования заданий и проверки выполненных работ.

1. Модуль процедур и функций разработчика, которые используются в макросах.
2. Модуль генерирования заданий.
3. Модуль проверки выполненных работ.
4. Модуль создания файлов с формой для размещения заданий.

Рассмотрим организацию работы программного комплекса.

Графический объект «Генерация» листа «Генерация» привязан к четвертому модулю и при нажатии на него выполняется макрос «Запуск».

```
Sub Запуск ()
'===== Заполнить обязательно! =====
list_name = Cells(7, 1)      'Имя листа
file_name = Cells(10, 1)    'Имя файла
metka = Cells(13, 1)       'метка
'=====

MyPath = ThisWorkbook.path
Dim Name_Kur(100, 3), m(3)
```

Рис. 4. Начальный фрагмент макроса «Запуск».

На рисунке 4 представлен фрагмент модуля «Запуск». В нем определяются основные параметры программы: имя листа, на котором расположена форма, имя создаваемого файла с заданиями, метка,

отображаемая в имени файла, путь расположения файла. Затем происходит чтение и формирование массива имен курсантов выбранной группы (рис. 5). На листе может располагаться несколько списков учебных групп и в зависимости от активной на листе ячейки будет сформирован тот или иной список. В случае, если активна ячейка первой строки, то автоматически будут сформированы файлы с индивидуальными заданиями для всей группы, а если активна ячейка с фамилией и именем курсанта, то создается только один файл.

```

i = 2                                'Заполнение массива Name_Kur
Do
  n = n + 1
  FullName = Cells(i, NoColumn)
  k = Len(FullName): j = 0
  For r = 1 To k
    If Mid(FullName, r, 1) = " " Then j = j + 1: m(j) = r
  Next r
  If j = 0 Then
    Name_Kur(n, 1) = FullName
    Name_Kur(n, 2) = ""
    Name_Kur(n, 3) = ""
  End If
  If j = 1 Then
    Name_Kur(n, 1) = Mid(FullName, 1, m(1) - 1)
    Name_Kur(n, 2) = Mid(FullName, m(1) + 1)
    Name_Kur(n, 3) = ""
  End If
  If j >= 2 Then
    Name_Kur(n, 1) = Mid(FullName, 1, m(1) - 1)
    Name_Kur(n, 2) = Mid(FullName, m(1) + 1, m(2) - m(1) - 1)
    Name_Kur(n, 3) = Mid(FullName, m(2) + 1)
  End If
  i = i + 1
Loop Until Cells(i, NoColumn) = ""

If NoRow = 1 Then
  a = 1: b = n                        'Задания для всей группы
Else
  a = NoRow - 1: b = NoRow - 1      'Задание для выбранного курсанта
End If

```

Рис. 5. Чтение списка имен курсантов.

Затем в цикле с параметром i для каждого задания копируется лист с заранее разработанной формой, вызывается макрос генерации задания, формирующий числовые, текстовые данные и свойства графических объектов для задач по заданной теме (рис. 6). Затем окончательно

составляется имя файла заданий, в него добавляется номер учебной группы, фамилия курсанта, первая буква его имени и метка. Сгенерированные файлы с индивидуальными заданиями сохраняются.

```

For i = a To b
  Sheets(list_name).Copy
  t = Name_Kur(i, 1)
  If Name_Kur(i, 2) <> "" Then t = t + " " + Mid(Name_Kur(i, 2), 1, 1) + "."
  If Name_Kur(i, 3) <> "" Then t = t + Mid(Name_Kur(i, 3), 1, 1) + "."
  Cells(3, 14) = t
  Call Prime_numbers_and_canonical_expansions 'вызов генерации задания
  t = Name_Kur(i, 1) + Mid(Name_Kur(i, 2), 1, 1) + Mid(Name_Kur(i, 3), 1, 1)
  ActiveWorkbook.SaveAs Filename:=MyPath + file_name + NoGroup + t + " " + _
    metka + ".xls", FileFormat:=xlExcel8, Password:="", WriteResPassword:="", _
    ReadOnlyRecommended:=False, CreateBackup:=False
  ActiveWindow.Close
Next i
    
```

Рис. 6. Копирование форм, вызов генерации и сохранение файла с заданием.

Подготовленные файлы с именными заданиями можно уже передавать обучаемым для их выполнения. Полученные файлы будут содержать только один лист со сгенерированными заданиями, без программных кодов. Все макросы, составляющие комплекс программ, остаются в исходном файле с шаблоном.

Как мы увидели ранее, вызов модуля генерации осуществляется автоматически из макроса «Запуск» при формировании именного файла с заданиями. Находящиеся на листе параметры: дата и время запуска генерации, а также введенный идентификатор обучаемого будут использоваться для определения номера варианта (рис. 7).

	Прикладная математика Итоговое практическое задание Дискретная математика	Дата <u>12.02.2021</u>
	Вариант	Время <u>08:23:34</u>
	30813	<u>Смирнов Ю.</u> (Фамилия И.О.)

1. Решить комбинаторную задачу.
 В учебной группе 7 юношей и 9 девушек. Сколькими способами можно выбрать 11 курсантов, таким образом чтобы среди них было 7 девушек?

1260

Рис. 7. Форма с заполненными параметрами и выведенным номером варианта.

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 [6].

Рассмотрим код процедуры вызова макросов генерации всех заданий программного комплекса «Прикладная математика» (рис. 8).

```
Sub Дискретная_математика ()
' Генерация заданий по темам: "Комбинаторика", "Теория чисел",
' "Сравнения", "Булева алгебра", "Графы" и "Теория автоматов".
PAROL_1 = "ПАРОЛЬ_ДЛЯ_ЗАЩИТЫ"
ActiveSheet.Protect Password:=PAROL_1, AllowFormattingCells:=True
ActiveSheet.EnableSelection = xlUnlockedCells
ActiveSheet.Unprotect Password:=PAROL_1

t = Timer
Cells(2, 15) = sTimer(t)
Cells(1, 15) = Date
Name = Cells(3, 14)
V = t Mod 31000 + codeName(Name)
Cells(6, 9) = V
Rnd (V * (-1))
Randomize (V)

Call Задание_1(V)
Call Задание_2(V)
Call Задание_3(V)
Call Задание_4(V)
Call Задание_5(V)
Call Задание_6(V)
Call Задание_7(V)
Call Задание_8_9(V)
Call Задание_10(V)
Call Задание_11(V)

ActiveSheet.Protect AllowFormattingCells:=True, Password:=PAROL_1
End Sub
```

Рис. 8. Код процедуры «Прикладная_математика».

В описываемом макросе все инструкции функционально можно разделить на три группы.

Первая группа инструкций предназначена для установления и снятия защиты листа с заданиями. Метод *ActiveSheet.Protect* используется для

защиты листа. В заполняемой форме, предлагаемой обучаемым для выполнения, все ячейки, кроме полей для ввода решений, должны быть помечены, как «защищаемые». Параметр *AllowFormattingCells* предоставляет пользователю возможность не только изменять содержимое заполняемых полей, но и форматировать ячейки, меняя их цвет, заливку, использовать надстрочные или подстрочные символы [5].

Во вторую группу инструкций макроса «Дискретная математика» включены инструкции, определяющие время и дату запуска макроса генерации, а также идентифицирующие обучаемого. На рисунке 9 представлен код процедуры преобразования идентификатора обучаемого в специальный код. Номер варианта формируется в диапазоне от 1 до 31999 и зависит только от времени запуска программы и кода введенной фамилии [6].

```
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. 9. Процедура codeName.

В завершении (третья группа инструкций) происходит вызов макросов генерации каждого из одиннадцати представленных заданий, также находящихся во втором модуле, и окончательная защита листа с подготовленными заданиями.

Рассмотрим основные принципы и особенности разработки макросов генерации и верификации математических заданий на примере процедуры

заполнения первого задания. На рисунке 7 уже было представлено сгенерированное первое задание комплекса. В задании предлагается типовая задача комбинаторного анализа с использованием формулы сочетаний без повторений.

На рисунке 10. представлен код макроса генерации этого задания.

```
Sub Задание_1(V)
Cells(8, 1) = "1. Решить комбинаторную задачу."
n1 = Int(Rnd(V) * 5) + 5
n2 = Int(Rnd(V) * 6) + 4
m1 = n1 - V Mod 2 - 2
M2 = n2 - V Mod 2 - 1
t = "В учебной группе "
t = t + Mid(Str(n1), 2)
t = t + " юношей и "
t = t + Mid(Str(n2), 2)
t = t + " девушек. Сколькими способами можно выбрать "
t = t + Mid(Str(m1 + M2), 2)
t = t + " курсантов, таким образом чтобы среди них было "
t = t + Mid(Str(M2), 2)
t = t + " девушек?"
Cells(9, 1) = t
End Sub
```

Рис. 10. Код генерации первого задания.

Используемыми параметрами задания являются переменные $n1$ и $n2$, которые содержат количество юношей и девушек в учебной группе, а также $m1$ и $m2$, определяющие количество выбранных курсантов. Значения указанных переменных генерируются с помощью функции *Rnd*, параметром которой является номер варианта v , и сколько бы раз не запускалась генерация упорядоченного набора заданий с этим вариантом, значения всех генерируемых параметров, рассчитываемых по одним и тем же алгоритмам, останутся неизменными. Этот факт в дальнейшем будет использоваться для проверки выполненных курсантами заданий. Определив номер варианта задания, выложенного для проверки, макрос проверки, используя последовательно те же алгоритмы расчета параметров, что и при генерации, рассчитает заново параметры заданий, найдет точное решение задания и сверит его с решением, внесенным на

проверяемый лист обучаемым. Если же изменить номер варианта, состав или порядок заданий, расчетные значения параметров также изменятся.

В приведенном примере, при генерации заданий, параметры $n1$ и $n2$ однозначно получают натуральные значения от 5 до 10 для юношей и от 4 до 10 для девушек, а сгенерированные значения $m1$ и $m2$ не будут превышать количество юношей и девушек. Во время автоматизированной проверки задания, выполненного обучаемым, с условием того, что алгоритмы вычисления параметров и порядок заданий не изменится, мы получим точно такие же значения вышеперечисленных переменных.

С учетом полученных значений параметров $n1$, $n2$, $m1$ и $m2$ в последующих строках кода процедуры генерации производится формирование значения переменной t , содержащей текст всего первого задания, после чего в заранее подготовленное поле выводится полученное строковое выражение.

Для проверки выполненных работ преподавателю достаточно нажать на графический объект на листе «Генерация» с надписью «Проверить работу». В результате запустится макрос «Найти_файл» (рис. 11), расположенный в четвертом модуле.

```
Private Sub Найти_файл()  
MyPath = ThisWorkbook.path  
  
fn = Application.GetOpenFilename _  
    ("Excel files (*.xls*),*.xls*,All files (*.*),*.*", 1, _  
    "Выберите Excel файлы для проверки", , False)  
  
'MsgBox "Выбран файл: '" & fn & "'", vbInformation, "Заголовок окна"  
Workbooks.Open Filename:=fn  
Call PR_Прикладная_математика(оценка)  
k = Len(fn)  
file_name = Mid(fn, 1, k - 4) & " " & оценка & ".xls"  
  
ActiveWorkbook.SaveAs Filename:=file_name, _  
    FileFormat:=xlExcel8, Password:="", WriteResPassword:="", _  
    ReadOnlyRecommended:=False, CreateBackup:=False  
Kill (fn)  
End Sub
```

Рис. 11. Макрос «Найти файл».

Данный макрос предназначен для поиска на дисковом пространстве файлов с выполненными работами. Для поиска файлов используется стандартный метод `GetOpenFilename`. Поиск начинается с текущего каталога, однако преподаватель в оконном режиме выбрать любую папку у себя на компьютере или в сетевом пространстве. Как правило, на занятии куранты используют определенные ресурсы файл-сервера, поэтому можно настроить начальный путь с нужной папки. После открытия нужного файла происходит вызов соответствующей процедуры для его проверки (рис. 12).

```
Sub PR_Прикладная_математика(оценка)
'
PAROL_1 = "GJPBND"
ActiveSheet.Protect Password:=PAROL_1, AllowFormattingCells:=True
ActiveSheet.EnableSelection = xlUnlockedCells
ActiveSheet.Unprotect Password:=PAROL_1

V = Cells(6, 9)
Name = Cells(3, 14)
stime = Cells(2, 15)

Name1000 = V - (Val(Mid(stime, 2, 2)) * 3600 + Val(Mid(stime, 5, 2)) * 60 _
+ Val(Mid(stime, 8, 2))) Mod 31000
If Name1000 <> codeName(Name) Then
    Cells(6, 12) = "Неверный код фамилии!!!"
    ActiveSheet.Protect Password:=PAROL_1
End
End If

Rnd (V * (-1))
Randomize (V)
mark = 0

Call PR_Задание_1(V, mark)
Call PR_Задание_2(V, mark)
Call PR_Задание_3(V, mark)|
Call PR_Задание_4(V, mark)
Call Pr_Задание_5(V, mark)
Call PR_Задание_6(V, mark)
Call PR_Задание_7(V, mark)
Call PR_Задание_8_9(V, mark)
Call PR_Задание_10(V, mark)
Call Pr_Задание_11(V, mark)
```

Рис. 12. Начальный фрагмент модуля верификации заданий.

В представленном макросе прослеживаются действия, похожие на те, что присутствовали и в модуле генерации. Сначала производится снятие

защиты с проверяемого листа, определяется номер варианта работы, имя фамилия и инициалы обучаемого, время запуска программы, затем осуществляется верификация соответствия трех элементов: варианта, идентификатора пользователя и указанного времени запуска программы. В случае несовпадения расчетного и указанного номеров вариантов выдается сообщение о нарушении идентификации и программа проверки прекращается.

В случае удачной проверки номера варианта таким же образом, как и во время генерации заданий в проверяемой работе, осуществляется инициализация генератора случайных чисел, объявление переменных, а затем выполнение проверки каждого выполненного задания.

Теперь рассмотрим код проверки первого задания программного комплекса «Дискретная математика» (рис. 13).

```
Sub PR_Задание_1(V, mark)
'1. Решить комбинаторные задачи.
n1 = Int(Rnd(V) * 5) + 5
n2 = Int(Rnd(V) * 6) + 4
m1 = n1 - V Mod 2 - 2
M2 = n2 - V Mod 2 - 1

' Проверка
r = soch(m1, n1) * soch(M2, n2)
If r = Cells(9, 16) Then mark = mark + 1 Else Call krest(9, 16)
End Sub
```

Рис. 13. Проверка задания «1. Решить комбинаторную задачу».

Как видим, процесс генерации целочисленных переменных $n1$, $n2$, $m1$ и $m2$ происходит так же, как и при формировании. Затем происходит расчет решения задачи. Так как мы одновременно выбираем $m1$ юношей из $n1$ имеющихся в учебной группе и $m2$ девушек из $n2$, то для нахождения решения r согласно «правилу умножения» нам достаточно перемножить количество сочетания $C_{n_1}^{m_1}$ и $C_{n_2}^{m_2}$.

Полученное числовое решение r сравниваем с результатом обучаемого, находящегося в ячейке Cells(9, 6). Если рассчитанное по описанному алгоритму решение совпадает с записью на листе, то

количество набранных за выполнение задания баллов *mark* увеличивается на единицу. В противном случае с помощью процедуры *krest* проверяемое поле перечеркивается красными диагональными линиями (рис. 14).

1. Решить комбинаторную задачу.

В учебной группе 7 юношей и 6 девушек. Сколькими способами можно выбрать 8 курсантов, таким образом чтобы среди них было 4 девушек?

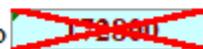


Рис. 14. Неверно выполненное задание после проверки.

Следует отметить, что трудоемкие задания могут оцениваться в два, три и более баллов.

Приведенный пример генерации и проверки задания использует только четыре целочисленных параметра, а линейный алгоритм расчета выводимых значений реализуется всего в одной элементарной инструкции, использующей обращения к функции *soch*. Но не всегда удается так лаконично построить вычисления, часто для формирования заданий необходимо использовать массивы числовых и текстовых данных, разветвляющиеся и циклические структуры, учитывать ряд ограничений и связей.

Полученные оценки в виде параметров могут передаваться в процедуру «Найти_файл» и автоматически приписываться к имени уже проверенного файла. Таким образом, проверенный файл с новым именем сохраняется, а файл-источник удаляется.

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. Булгаков, В.В. Структурно-методическая модель компьютерной программы контроля теоретических знаний курсантов /В.В. Булгаков // Открытое образование (Научный журнал). – М.: Российский экономический университет имени Г.В. Плеханова, 2018. – Том 22, №3. С. 4 – 13.

2. 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.

3. 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.

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. Лаптев, В.Н. О технологиях разработки программных приложений для генерирования и проверки практических заданий 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, 2020. – №01(155). С. 164 – 177. – IDA [article ID]: 1552001013. – Режим доступа: <http://ej.kubagro.ru/2020/01/pdf/13.pdf>, 0,875 у.п.л.

6. Лаптев, В.Н. О технологиях разработки программных приложений для генерирования и проверки практических заданий 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, 2020. – №02(156). С. 143 – 158. – IDA [article ID]: 1562002008. – Режим доступа: <http://ej.kubagro.ru/2020/02/pdf/08.pdf>, 1 у.п.л.

References

1. Bulgakov, V.V. Strukturno-metodicheskaya model` komp`yuternoj programmy` kontrolya teoreticheskix znaniy kursantov /V.V. Bulgakov // Otkry`toe obrazovanie (Nauchny`j zhurnal). – M.: Rossijskij e`konomicheskij universitet imeni G.V. Plekhanova, 2018. – Tom 22, №3. S. 4 – 13.

2. Laptev, V.N. Nekotory`e aspekty` primeneniya sredi` Visual Basic for Application dlya sozdaniya uchebny`x prilozhenij po matematicheskim disciplinam / V.N. Laptev, E.V. Mixajlenko // Politematicheskij setевой e`lektronny`j nauchny`j zhurnal Kubanskogo gosudarstvennogo agrarnogo universiteta (Nauchny`j zhurnal KubGAU) [E`lektronny`j resurs]. – 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.

3. Laptev, V.N. Metody` razrabotki testovy`x zadaniy v avtomatizirovannoj kontroliruyushhej sisteme «Kontrol`» / V.N. Laptev, E.V. Mixajlenko // Politematicheskij setевой e`lektronny`j nauchny`j zhurnal Kubanskogo gosudarstvennogo agrarnogo universiteta (Nauchny`j zhurnal KubGAU) [E`lektronny`j 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.

4. Laptev, V.N. Organizaciya testirovaniya v avtomatizirovannoj kontroliruyushhej sisteme «Kontrol`» / V.N. Laptev, E.V. Mixajlenko // Politematicheskij setевой e`lektronny`j nauchny`j zhurnal Kubanskogo gosudarstvennogo agrarnogo universiteta (Nauchny`j zhurnal KubGAU) [E`lektronny`j resurs]. – 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. Laptev, V.N. O texnologiyax razrabotki programmny`x prilozhenij dlya generirovaniya i proverki prakticheskix zadaniy po matematicheskim disciplinam / V.N. Laptev, E.V. Mixajlenko // Politematicheskij setевой e`lektronny`j nauchny`j zhurnal Kubanskogo gosudarstvennogo agrarnogo universiteta (Nauchny`j zhurnal KubGAU) [E`lektronny`j resurs]. – Krasnodar: KubGAU, 2020. – №01(155). S. 164 – 177. – IDA [article ID]: 1552001013. – Rezhim dostupa: <http://ej.kubagro.ru/2020/01/pdf/13.pdf>, 0,875 u.p.l.

6. Laptev, V.N. O texnologiyax razrabotki programmny`x prilozhenij dlya generirovaniya i proverki prakticheskix zadaniy po matematicheskim disciplinam / V.N. Laptev, E.V. Mixajlenko // Politematicheskij setевой e`lektronny`j nauchny`j zhurnal Kubanskogo gosudarstvennogo agrarnogo universiteta (Nauchny`j zhurnal KubGAU) [E`lektronny`j resurs]. – Krasnodar: KubGAU, 2020. – №02(156). S. 143 – 158. – IDA [article ID]: 1562002008. – Rezhim dostupa: <http://ej.kubagro.ru/2020/02/pdf/08.pdf>, 1 u.p.l.