

УДК 519.688

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

О РАЗРАБОТКЕ ПРОГРАММНОГО КОМПЛЕКСА «ГРАФЫ. ФОРМЫ И ОПЕРАЦИИ»

Лаптев Владимир Николаевич
к.т.н., доцент
ФГБОУ «Кубанский государственный аграрный университет», 350044, Россия, г. Краснодар, ул. Калинина 13

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

Прокопенко Алексей Николаевич
к.т.н., доцент, начальник кафедры
E-mail: belui@mvd.ru
ФГКОУ ВПО «Белгородский юридический институт МВД России имени И.Д. Путилина», 308024, Россия, г. Белгород, ул. Горького, 71

Статья посвящена основным принципам разработки компьютерных приложений для генерирования практических заданий по разделу «Графы», особенностям использования некоторых методических приемов и технологий при разработке компьютерных программ для подготовки практических заданий и анализа их выполнения обучаемыми по математическим дисциплинам. В работе определяются принципы разработки программных продуктов, алгоритмы и технологии разрабатываемых макросов, описывается структура компьютерных программ, исследуются основные компоненты реализуемых программ в среде разработки Visual Basic for Application, приводятся алгоритмы основных процедур. Для профессорско-преподавательского состава, инженеров-программистов образовательных организаций

Ключевые слова: АЛГОРИТМ, ВЕРИФИКАЦИЯ, ГЕНЕРАЦИЯ, ГРАФ, КОМПЬЮТЕРНАЯ ПРОГРАММА, МАКРОС

<http://dx.doi.org/10.21515/1990-4665-175-010>

<http://ej.kubagro.ru/2022/01/pdf/10.pdf>

UDC 519.688

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

DEVELOPMENT OF THE “GRAPHS. FORMS AND OPERATIONS” SOFTWARE COMPLEX

Laptev Vladimir Nikolaevich
Cand.Tech.Sci., associate professor
Kuban State Agricultural university, 350044, Russia, Krasnodar, Kalinina, 13

Mikhaylenko Evgeny Vladimirovich
Cand.Phys.-Math.Sci., associate professor, 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

Prokopenko Alexey Nikolaevich
Cand.Tech.Sci., Associate Professor, Head of the Department
E-mail: belui@mvd.ru
Federal State Educational Institution of Higher Professional Education “Belgorod Law Institute of the Ministry of Internal Affairs of the Russian Federation named after I.D. Putilin”, 308024, Russia, Belgorod, Gorkogo, 71

The article is devoted to the basic principles of developing computer applications for generating practical tasks for the section "Graphs", the peculiarities of using some methodological techniques and technologies in developing computer programs for preparing practical tasks and analyzing their implementation by students 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 programs implemented in the Visual Basic for Application development environment, provides algorithms for the main procedures. It may be recommended for the teaching staff and software engineers of educational organizations

Keywords: ALGORITHM, VERIFICATION, GENERATION, GRAPH, COMPUTER PROGRAM, MACROS

Настоящая статья описывает процедуры подготовки программных продуктов для генерирования и верификации практических заданий по математическим дисциплинам. Описываемые методы рассчитаны на использование разработчиками среды офисного пакета Microsoft Office, в настоящее время являющегося наиболее распространенным в России, 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, 7]. Кроме того, разработанные в среде Microsoft Office программные продукты в большинстве случаев без потерь будут открываться и работать в указанных системах [5].

Статья предназначена для разработчиков программного обеспечения в области математического образования. Ими могут быть, как профессиональные инженеры-программисты технических служб образовательных организаций МВД России, так и преподаватели кафедр, ведущие занятия по математическим и информационно-техническим дисциплинам, являющиеся продвинутыми пользователями в области компьютерных технологий [12].

Рассмотрим программный комплекс компьютерных программ «Графы. Формы и операции», разработанный на кафедре информатики и математики Краснодарского университета МВД России для изучения дисциплин математика и прикладная математика.

На рисунке 1 представлено первое задание комплекса «Графы. Формы и операции»

1. Представить графы G_1 и G_2 в алгебраической форме, указать степени указанных вершин, количество ребер (дуг) и петель графов.

$G_1 =$ $\deg(c)$ Количество петель G_1
 $G_2 =$ $\deg^-(f)$ Полная степень G_1
 $G_3 =$ $\deg^+(d)$ Количество дуг G_2
 Вершины, смежные d Все изолированные вершины Висячие вершины

Рис. 1. Первое задание комплекса «Графы. Формы и операции»

На листе представлены три графа G_1 , G_2 и G_3 в графической форме: первый граф является неориентированным, второй – ориентированным, а граф G_3 – смешанный [3]. Обучаемым необходимо представить описанные графы в алгебраической форме, указать степени трех вершин, полную степень графа G_1 , количество ребер, дуг и петель графов.

При разработке кода для генерации этого задания пришлось столкнуться с проблемой определения графов и вывода этого графа на лист [8]. У графа G_1 всего три фиксированных вершины: a , b и c . В коде программы наш граф G_1 будет представлен в виде четырехмерного одноименного массива $G1$.

Сперва определим обозначения всех возможных ребер нашего графа (рис. 2). Этими обозначения мы впоследствии мы будем пользоваться для проверки вывода графа в алгебраической форме. В программе для ребер используются два обозначения: первое (слева) указывает на имя Shape-объекта листа заданий, соответствующего ребру, а второе (справа) – обозначение ребра в алгебраической форме. Как видим из фрагмента программы граф может иметь не более 15 ребер.

```

G1(1, 1, 1, 0) = "aa1": G1(1, 1, 1, 10) = "(a,a)"
G1(1, 1, 2, 0) = "aa2": G1(1, 1, 2, 10) = "(a,a)"
G1(2, 2, 1, 0) = "bb1": G1(2, 2, 1, 10) = "(b,b)"
G1(2, 2, 2, 0) = "bb2": G1(2, 2, 2, 10) = "(b,b)"
G1(3, 3, 1, 0) = "cc1": G1(3, 3, 1, 10) = "(c,c)"
G1(3, 3, 2, 0) = "cc2": G1(3, 3, 2, 10) = "(c,c)"
G1(1, 2, 1, 0) = "ab1": G1(1, 2, 1, 10) = "(a,b)"
G1(1, 2, 2, 0) = "ab2": G1(1, 2, 2, 10) = "(a,b)"
G1(1, 2, 3, 0) = "ab3": G1(1, 2, 3, 10) = "(a,b)"
G1(1, 3, 1, 0) = "ac1": G1(1, 3, 1, 10) = "(a,c)"
G1(1, 3, 2, 0) = "ac2": G1(1, 3, 2, 10) = "(a,c)"
G1(1, 3, 3, 0) = "ac3": G1(1, 3, 3, 10) = "(a,c)"
G1(2, 3, 1, 0) = "bc1": G1(2, 3, 1, 10) = "(b,c)"
G1(2, 3, 2, 0) = "bc2": G1(2, 3, 2, 10) = "(b,c)"
G1(2, 3, 3, 0) = "bc3": G1(2, 3, 3, 10) = "(b,c)"

```

Рис. 2. Установление обозначений ребер графа G_1 .

Определение ребер графа осуществляется в цикле с постусловием Do .. Loop Until. Перебором параметров i, j, k случайным образом осуществляется привязка соответствующего ребра к графу [4]. Если $G1(i, j, k, 1) = 1$, соответствующее ребро будет принадлежать графу G_1 , а в случае значения 0 ребро считается несуществующим. Тело цикла будет повторяться до тех пор, пока графу G_1 не будут инцидентны от шести до восьми ребер. На рисунке 3 представлен код задания графа G_1 .

```

Do
r = 0
  For i = 1 To 3
    For j = i To 3
      For k = 1 To 3
        If Not (i = j And k = 3) Then
          If Rnd(V) > 0.5 Then G1(i, j, k, 1) = 1: r = r + 1
          Else: G1(i, j, k, 1) = 0
        End If
      Next k, j, i
    Loop Until r >= 6 And r <= 8

```

Рис. 3. Определение ребер графа G_1 .

Граф G_1 уже выведен на лист задания. Однако изначально все его пятнадцать ребер представлены красными линиями. Зная структуру графа G_1 , оставим принадлежащие ему ребра видимыми, а не принадлежащие графу сделаем невидимыми. На рисунке 4. представлен код определения видимости ребер, инцидентных графу G_1 .

```

For i = 1 To 3                                'Определение видимости
  For j = i To 3
    For k = 1 To 3
      If Not (i = j And k = 3) Then
        With ActiveSheet.Shapes(G1(i, j, k, 0))
          If G1(i, j, k, 1) = 1 Then .Visible = 1 Else .Visible = 0
        End With
      End If
    Next k, j, i
  Next k, j, i

```

Рис 4. Осуществление видимости ребер графа G₁.

На рисунке 5 представлены примеры различных вариантов формирования графа G₁. Общее количество различных вариантов формирования графа равно $C_{15}^6 + C_{15}^7 + C_{15}^8 = 5005 + 6435 + 6435 = 17875$.

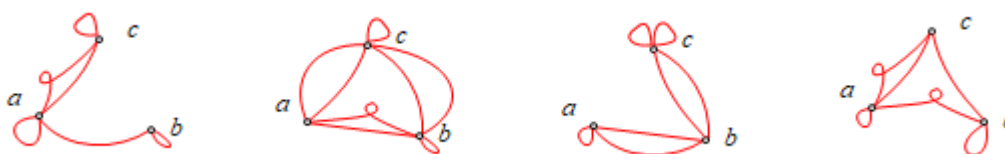


Рис. 5. Некоторые варианты формирования графа G₁.

Для проверки результата выполнения задания обучаемыми определим алгебраическую форму графа G₁ (рис. 6). Зная ребра инцидентные графу, в список t1 добавим имена ребер G1(i, j, k, 10), заданных ранее для алгебраической формы. Значение t1 во время проверки можно будет сравнить с результатом выполнения задания курсантом.

```

t1 = "[{a,b,c},{                                'определение алгебраической формы G1
For i = 1 To 3
  For j = i To 3
    For k = 1 To 3
      If Not (i = j And k = 3) Then
        If G1(i, j, k, 1) = 1 Then t1 = t1 + G1(i, j, k, 10) + ","
      End If
    Next k, j, i
  Next k, j, i
  t1 = Mid(t1, 1, Len(t1) - 1) + "]"
'Cells(15, 2) = t1

```

Рис. 6. Определение алгебраической формы графа G₁.

Рассчитаем степени всех вершин (рис. 7). Для этого, перебирая значения массива G1(i, j, k, 1), находим те, в которых присутствуют соответственно значения «a», «b» или «c». Рассчитанные степени вершин можно использовать при проверке результата выполнения сформированной формы обучаемыми.

```

dega = 0: degb = 0: degc = 0:   'Степени вершин a, b и c
For i = 1 To 3
  For j = i To 3
    For k = 1 To 3
      If Not (i = j And k = 3) And G1(i, j, k, 1) = 1 Then
        If Mid(G1(i, j, k, 0), 1, 1) = "a" Or _
          Mid(G1(i, j, k, 0), 2, 1) = "a" Then dega = dega + 1
        If Mid(G1(i, j, k, 0), 1, 1) = "b" Or _
          Mid(G1(i, j, k, 0), 2, 1) = "b" Then degb = degb + 1
        If Mid(G1(i, j, k, 0), 1, 1) = "c" Or _
          Mid(G1(i, j, k, 0), 2, 1) = "c" Then degc = degc + 1
      End If
    Next k, j, i
  If V Mod 3 = 0 Then Cells(15, 11) = "deg(a)": deg = dega
  If V Mod 3 = 1 Then Cells(15, 11) = "deg(b)": deg = degb
  If V Mod 3 = 2 Then Cells(15, 11) = "deg(c)": deg = degc

```

Рис. 7. Код расчета степеней вершин графа G_1 .

Алгоритм формирования ребер графа G_1 приводит к тому, что в графе с большой вероятностью могут появиться петли [11]. Определим количество петель графа G_1 (рис. 8). Для этого с использованием вложенных циклов с параметрами i, j, k подсчитаем количество сочетаний «aa», «bb», «cc» в массиве $G1$.

```

kp = 0                               'Количество петель G1
For i = 1 To 3
  For j = i To 3
    For k = 1 To 3
      If Not (i = j And k = 3) And G1(i, j, k, 1) = 1 Then
        If Mid(G1(i, j, k, 0), 1, 2) = "aa" Then kp = kp + 1
        If Mid(G1(i, j, k, 0), 1, 2) = "bb" Then kp = kp + 1
        If Mid(G1(i, j, k, 0), 1, 2) = "cc" Then kp = kp + 1
      End If
    Next k, j, i

```

Рис. 8. Расчет петель графа G_1 .

Граф G_2 является ориентированным, но алгоритм работы с таким графом схож с неориентированным. Для определения дуг ориентированного графа следует различать дуги с различными направлениями. Представим код формирования ориентированного графа G_2 (рис. 9). Здесь инцидентность дуги устанавливается с вероятностью 0,6.

```

Do
  r = 0
  For i = 1 To 5
    deg_(i) = 0: degP(i) = 0
  Next i

  For i = 1 To 5
    For j = 1 To 5
      For k = 1 To 1
        If Not (i = j) Then
          If Rnd(V) > 0.6 Then
            G2(i, j, k, 1) = 1
            r = r + 1
            deg_(i) = deg_(i) + 1
            degP(j) = degP(j) + 1
          Else
            G2(i, j, k, 1) = 0
          End If
        End If
      End If
    Next k, j, i
    'k2i - изолированные, k2v - висячие
  k2i = 0: k2v = 0
  izo_ver = "": vis_ver = ""
  For i = 1 To 5
    If deg_(i) + degP(i) = 0 Then k2i = k2i + 1
    If deg_(i) + degP(i) = 1 Then k2v = k2v + 1
  Next i
Loop Until r >= 6 And r <= 8 And k2i > 0 And k2v > 0

```

Рис. 9. Формирование ориентированного графа G_2 .

Как видим из кода фрагмента программы, в цикле Do .. Loop Until не только определяются инцидентные вершины, но и рассчитываются степени полузахода и полуисхода каждой вершины графа. Граф G_2 задается с условиями, что количество дуг должно быть от шести до восьми, и в графе обязательно должны присутствовать изолированные и висячие вершины. Для проверки задания формируются списки изолированных и висячих вершин графа G_2 (рис. 10).

```

t = "defgh"
'формируем изолированные и висячие вершины
For i = 1 To 5
  If deg_(i) + degP(i) = 0 Then izo_ver = izo_ver + Mid(t, i, 1)
  If deg_(i) + degP(i) = 1 Then vis_ver = vis_ver + Mid(t, i, 1)
Next i

```

Рис. 10. Создание списков изолированных и висячих вершин графа G_2 .

Особый интерес вызывает изображение ориентированного графа на лист. В ориентированном графе дуги изображаются прямыми или кривыми

линиями со стрелками, указывающими конец дуги. Мы не будем проверять, в каком состоянии находятся линии, представляющие дуги графа. Сначала мы делаем все линии невидимыми и убираем все стрелки (рис. 11), а затем, проверяя наличие дуг с учетом их направлений, делаем нужные линии видимыми и устанавливаем стрелки в начале или конце линии. Вариативность для графа G_2 гораздо выше, чем для графа G_1 , так граф содержит с учетом направлений двадцать дуг [5]. Общее количество различных вариантов формирования графа равно $C_{20}^6 + C_{20}^7 + C_{20}^8 = 38760 + 77520 + 125970 = 242250$.

```

For i = 1 To 5      'Делаем все линии невидимыми, убираем все стрелки
  For j = 1 To 5
    For k = 1 To 1
      If Not (i = j) Then
        With ActiveSheet.Shapes(G2(i, j, k, 0))
          .Visible = 0
          .Line.EndArrowheadStyle = 1
          .Line.BeginArrowheadStyle = 1
        End With
      End If
    Next k, j, i

For i = 1 To 5      'Делаем линии видимыми, рисуем стрелки
  For j = 1 To 5
    For k = 1 To 1
      If Not (i = j) Then
        With ActiveSheet.Shapes(G2(i, j, k, 0))
          If G2(i, j, 1, 1) = 1 Then
            .Visible = 1
            If i < j Then .Line.EndArrowheadStyle = 2
            If i > j Then .Line.BeginArrowheadStyle = 2
          End If
        End With
      End If
    Next k, j, i

```

Рис. 11. Определение видимости и направлений дуг графа G_2 .

Для смешанного графа G_3 алгоритм построения графа и вывода его на лист аналогичны ориентированному графу G_2 . Здесь шести вершинам графа G_3 инцидентны тридцать шесть различных дуг. Так как в графе G_3 должны присутствовать и ребра и дуги, то отсутствие стрелок на линии означает наличие двух дуг, направленных в разные стороны. С учетом

этого код программы обработки линий для смешанного графа G_3 существенно изменен (рис. 12).

```

For i = 1 To 6           'Делаем линии видимыми, рисуем стрелки
  For j = 1 To 6
    For k = 1 To 1
      If Not (i = j) Then
        With ActiveSheet.Shapes(G3(i, j, k, 0))
          If G3(i, j, 1, 1) = 1 Then
            .Visible = 1
            If i < j Then .Line.EndArrowheadStyle = 2
            If i > j Then .Line.BeginArrowheadStyle = 2
            If G3(j, i, 1, 1) = 1 Then
              .Line.EndArrowheadStyle = 1
              .Line.BeginArrowheadStyle = 1
            End If
          End If
        End With
      End If
    Next k, j, i
  
```

Рис. 12. Определение видимости и направлений дуг графа G_2 .

Во втором задании обучаемому предлагается заполнить матрицы смежности и матрицы инцидентности для графов, выведенных в графической форме (рис. 13).

2. Заполнить матрицы смежности и матрицы инцидентности (Ребра нумеруются в порядке представления в алгебраической форме).

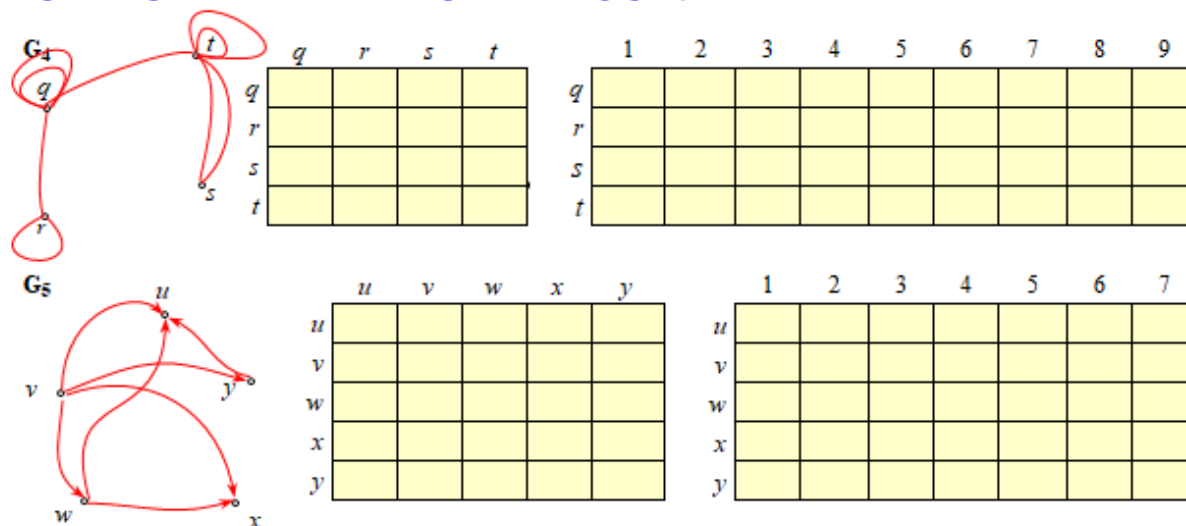


Рис. 13. Вид задания «2. Заполнить матрицы смежности и инцидентности.

Алгоритмы определения и вывода на лист неориентированного графа G_4 и ориентированного графа G_5 такие же, как и для ранее приведенных графов G_1 и G_2 . Однако количество ребер у графа G_4 ,

согласно заполняемым матрицам смежности, должно быть равно девяти, а число дуг графа G_5 составляет семь [6].

Для проверки задания для каждого графа составляются матрицы смежности и инцидентности. На рисунке 14 приведен фрагмент кода программы для формирования и проверки заполненной обучаемым матрицы смежности графа G_4 .

```
Erase MS4                                'Формирование матрицы смежности для G4
For i = 1 To 4
  For j = i To 4
    For k = 1 To 3
      If G4(i, j, k, 1) = 1 Then
        MS4(i, j) = MS4(i, j) + 1
        MS4(j, i) = MS4(i, j)
      End If
    Next k, j, i

ml = 0                                    'Проверка матрицы смежности для G4
For i = 1 To 4
  For j = 1 To 4
    If Cells(25 + i, 4 + j) = MS4(i, j) And Cells(25 + i, 4 + j) <> "" _
      Then ml = ml + 1 Else Call krest(25 + i, 4 + j)
    Next j, i
  If ml = 16 Then mark = mark + 2
```

Рис. 14. Формирование и проверка матрицы смежности графа G_4 .

В представленном коде для построения матрицы смежности для каждой пары вершин с номерами i и j проверяются все возможные варианты ребер, инцидентных данным вершинам, а затем построенная таким образом матрица смежности сравнивается с таблицей, заполненной курсантом. В случае идентичности всех шестнадцати значений матрицы смежности обучаемый получает два балла. Похожим образом осуществляются формирование, вывод и проверка остальных матриц.

Третье задание комплекса программ «Графы. Формы и операции» заключается в выполнении операций над графами, заданных в матричных формах. Обучаемому предлагается выполнить объединение и пересечение для неориентированных графов G_6 и G_7 , а также разность и кольцевую сумму ориентированных графов G_8 и G_9 . На рисунке 15 Представлена форма со сгенерированным третьим заданием. В задании обучаемый

должен заполнить не только внутреннюю часть матриц в табличной форме – количество инцидентных ребер или дуг, но и вывести идентификаторы вершин.

3. Выполнить операции над графами, заданными в матричных формах.

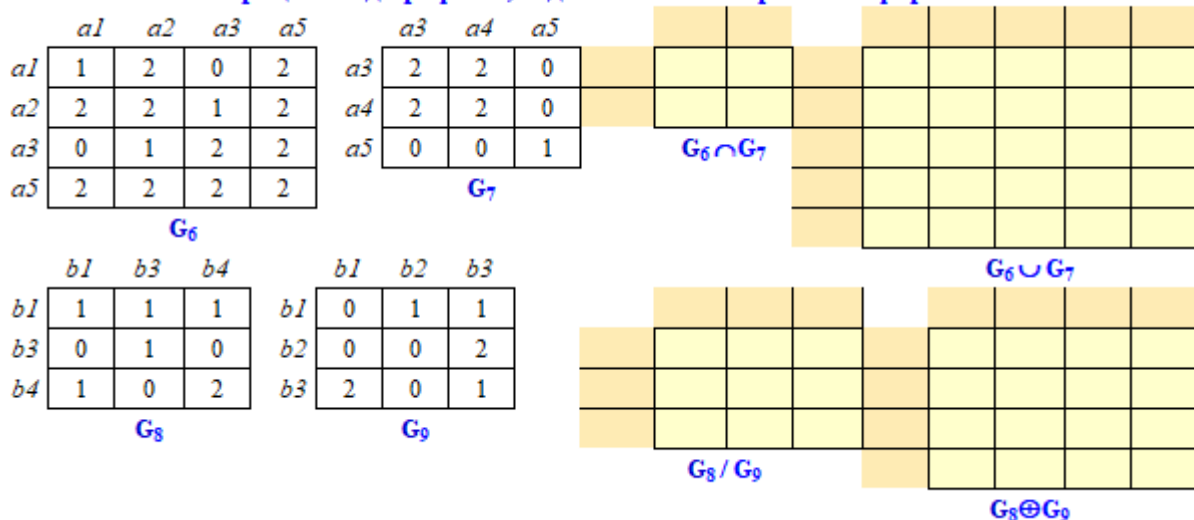


Рис. 15. Задание 3. Выполнить операции над матрицами.

Нам не нужно выводить графы на лист в графической форме, поэтому достаточно сформировать только матрицы смежности для каждого графа (рис. 15). Однако, для того, чтобы участвующие в бинарной операции два графа имели общие вершины и их пересечение не было пустым определяем фиксированное количество вершин и случайным образом для каждого из двух графов исключаем некоторое количество неинцидентных вершин [2].

Например, для графов G_6 и G_7 мы определим всего пять вершин (рис. 16). Для графа G_6 с помощью генератора случайных чисел исключим вершину vx_1 , а для графа G_7 исключим две вершины vx_2 и vx_3 . Таким образом, граф G_6 будет содержать четыре вершины, а у графа G_7 останется только пять. Согласно представленному алгоритму все исключенные вершины vx_1 , vx_2 и vx_3 будут различными.

```

Do                                     'Находим отсутствующие вершины из пяти
  vx1 = Int(Rnd(V) * 100) Mod 5 + 1    'для графа G6
  vx2 = Int(Rnd(V) * 100) Mod 5 + 1    'для графа G7
  vx3 = Int(Rnd(V) * 100) Mod 5 + 1    'для графа G7
Loop Until vx1 <> vx2 And vx2 < vx3 And vx3 <> vx1

```

Рис. 16. Исключение вершин для графов G_6 и G_7 .

С учетом выбранных вершин алгоритм формирования матриц смежности для графов G_6 и G_7 будет следующим (рис. 17).

```

Do                                     'Генерируем матрицу смежности графа G6
  r = 0
  For i = 1 To 5
    For j = i To 5
      If i <> vx1 And j <> vx1 Then
        If Rnd(V) < 0.3 Then
          p = Int(Rnd(V) * 2.5)
          GM_6(i, j) = p
          If i <> j Then GM_6(j, i) = p
          r = r + p
        End If
      End If
    Next j, i
  Loop Until r >= 8 And r <= 12

Do                                     'Генерируем матрицу смежности графа G7
  r = 0
  For i = 1 To 5
    For j = i To 5
      If i <> vx2 And j <> vx2 And i <> vx3 And j <> vx3 Then
        If Rnd(V) < 0.3 Then
          p = Int(Rnd(V) * 2.5)
          GM_7(i, j) = p
          If i <> j Then GM_7(j, i) = p
          r = r + p
        End If
      End If
    Next j, i
  Loop Until r >= 5 And r <= 10

```

Рис. 17. Формирование матриц смежности графов G_6 и G_7 .

Из представленного кода программы мы видим, что у графа G_6 определяются четыре вершины и от восьми до двенадцати ребер, часть из которых будет кратными, а для графа G_7 от пяти до десяти ребер, инцидентных трем вершинам.

Операции объединения и пересечения графов G_6 и G_7 также будем выполнять в матричном виде.

```
Erase MX                                     'формируем матрицу объединения графов G6 и G7
For i = 1 To 5
    MX(i, 0) = "a" + st(i)
    MX(0, i) = "a" + st(i)
Next i

For i = 1 To 5
    For j = 1 To 5
        If GM_6(i, j) > GM_7(i, j) Then MX(i, j) = GM_6(i, j) Else MX(i, j) = GM_7(i, j)
    Next j, i
```

Рис. 18. Объединение графов G_6 и G_7 .

Алгоритм объединения графов G_6 и G_7 следующий (рис. 18). Сначала мы определяем наименования строк и столбцов матрицы смежности объединения графов, которые тоже будут проверяться в выполненном задании. а затем заполняем каждый элемент матрицы смежности объединения графов, вставляя туда наибольшее из значений соответствующих элементов матриц смежности графов G_6 и G_7 .

Некоторую трудность представляет проверка матрицы пересечения графов G_6 и G_7 (рис. 19).

```

For i = 1 To 5                                'формируем матрицу пересечения графов G6 и G7
  For j = 1 To 5
    If GM_6(i, j) < GM_7(i, j) Then MX(i, j) = GM_6(i, j) Else MX(i, j) = GM_7(i, j)
  Next j, i

vx4 = 0                                       'Находим вершины пересечения графов
Do
  vx4 = vx4 + 1
Loop Until vx4 <> vx1 And vx4 <> vx2 And vx4 <> vx3
vx5 = vx4
Do
  vx5 = vx5 + 1
Loop Until vx5 <> vx1 And vx5 <> vx2 And vx5 <> vx3

PX(1, 0) = MX(vx4, 0)
PX(2, 0) = MX(vx5, 0)

PX(0, 1) = MX(0, vx4)
PX(1, 1) = MX(vx4, vx4)
PX(2, 1) = MX(vx5, vx4)

PX(0, 2) = MX(0, vx5)
PX(1, 2) = MX(vx4, vx5)
PX(2, 2) = MX(vx5, vx5)

m2 = 0                                       'Проверяем матрицу пересечения графов G6 и G7
For i = 0 To 2
  For j = 0 To 2
    If Not (i = 0 And j = 0) Then
      If Cells(39 + i, 10 + j) = PX(i, j) And Cells(39 + i, 10 + j) <> "" _
        Then m2 = m2 + 1 Else Call krest(39 + i, 10 + j)
    End If
  Next j, i
If m2 = 8 Then mark = mark + 2

```

Рис. 19. Реализация пересечения графов G_6 и G_7 .

Матрицу смежности пересечения графов G_6 и G_7 формируем, выбирая наименьшие значения двумерных массивов GM_6 и GM_7 . Однако в результирующей матрице три столбца и три строки оказываются нулевыми, следовательно, необходимо перед проверкой работы привести матрицу ко второму порядку. Операции с графами G_8 и G_9 реализованы аналогично.

В представленной статье мы рассмотрели общие принципы разработки компьютерных приложений для генерирования и верификации практических заданий по изучаемым курсантами математическим дисциплинам [9], а также специфические методы и технологии для создания обучающих компьютерных программ, содержащих сложные математические построения.

Использование методов и технологий создания приложений для генерирования и проверки практических заданий по математическим дисциплинам с использованием офисных программных продуктов является перспективным направлением в работе по созданию и организации электронной образовательной среды [10].

Представленный в статье программный продукт успешно используются в учебном процессе Краснодарского университета МВД России при изучении дисциплин математика, прикладная математика курсантами и слушателями, обучающимися по специальностям 38.05.01 Экономическая безопасность, 10.05.05 Безопасность информационных технологий в правоохранительной сфере и 40.05.03 Судебная экспертиза.

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. 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.

2. Лаптев, В.Н. Основные принципы использования офисных приложений для подготовки и верификации электронных практических заданий / В.Н. Лаптев, Е.В. Михайленко // Политематический сетевой электронный научный журнал Кубанского государственного аграрного университета (Научный журнал КубГАУ) [Электронный ресурс]. – Краснодар: КубГАУ, 2021. – №09(173). С. 76 – 92. – IDA [article ID]: 1732109007. – Режим доступа: <http://ej.kubagro.ru/2021/09/pdf/07.pdf>, 1,062 у.п.л.

3. Михайленко, Е.В. Автоматизация обработки табличных данных для сбора и анализа социально-экономических показателей сотрудниками подразделений органов внутренних дел / Е.В. Михайленко, А.А. Хромых – Краснодар: Краснодарский университет МВД России, 2021. 148 с.

4. Михайленко, Е.В. Введение в программирование: учеб. пособие / Е.В. Михайленко – Краснодар: Краснодарский университет МВД России, 2020. 109 с.

5. Михайленко, Е.В. О разработке компьютерных программ для автоматизации практических занятий по разделу «Теория вероятностей и математическая статистика» дисциплины математика / Е.В. Михайленко // Проблемы информационного обеспечения деятельности правоохранительных органов: сборник статей 5-й Всероссийской научно-практической конференции. – Белгород: Белгородский юридический институт МВД России имени И.Д. Путилина, 2019. – С. 27 – 45.

6. Михайленко, Е.В. О разработке программных приложений для выполнения практических заданий по разделу «Методы оптимизации» дисциплины прикладная

математика / Е.В. Михайленко // Проблемы информационного обеспечения деятельности правоохранительных органов: сборник статей 4-й Всероссийской научно-практической конференции. – Белгород: Белгородский юридический институт МВД России имени И.Д. Путилина, 2018. – С. 210 – 222.

7. Михайленко, Е.В. Общие принципы автоматизации подбора заданий и проверки выполненных курсантами работ по математическим дисциплинам / Е.В. Михайленко // Математические методы и информационно-технические средства: материалы XV Всерос. науч.-практ. конф. – Краснодар: Краснодарский университет МВД России, 2019. С. 122 – 127.

8. Михайленко, Е.В. Особенности разработки компьютерных приложений для генерации и верификации практических заданий, включающих обработку матриц / Е.В. Михайленко // Проблемы информационного обеспечения деятельности правоохранительных органов: сборник статей 7-й Всероссийской научно-практической конференции. – Белгород: Белгородский юридический институт МВД России имени И.Д. Путилина, 2020. – С. 3 – 20.

9. Михайленко, Е.В. Принципы автоматизации подбора и проверки практических заданий, включающих обработку полиномов / Е.В. Михайленко // Математические методы и информационно-технические средства: материалы XVI Всерос. науч.-практ. конф. – Краснодар: Краснодарский университет МВД России, 2020. С. 65 – 70.

10. Михайленко, Е.В. Технологии подготовки компьютерных программ для генерации и проверки практических заданий по теме «Сравнения» / Е.В. Михайленко // Математические методы и информационно-технические средства: материалы XVII Всерос. науч.-практ. конф. – Краснодар: Краснодарский университет МВД России, 2021. С. 101 – 111.

11. Михайленко, Е.В. Технологии программирования: учеб. пособие / Е.В. Михайленко, А.К. Назаров – Краснодар: Краснодарский университет МВД России, 2019. 510 с.

12. Старостенко И.Н. К вопросу об использовании информационных технологий при организации самостоятельной работы обучающихся / И.Н. Старостенко, Е.В. Михайленко // Проблемы современного педагогического образования. – Ялта: РИО ГПА, 2019. № 65-1. С. 257 - 262.

References

1. Laptev, V.N. O tehnologijah razrabotki programnyh prilozhenij dlja generirovaniya i proverki prakticheskikh zadaniy po matematicheskim disciplinam / V.N. Laptev, E.V. Mihajlenko // Politematicheskij setevoj jelektronnyj nauchnyj zhurnal Kubanskogo gosudarstvennogo agrarnogo universiteta (Nauchnyj zhurnal KubGAU) [Jelektronnyj 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.

2. Laptev, V.N. Osnovnye principy ispol'zovaniya ofisnyh prilozhenij dlja podgotovki i verifikacii jelektronnyh prakticheskikh zadaniy / V.N. Laptev, E.V. Mihajlenko // Politematicheskij setevoj jelektronnyj nauchnyj zhurnal Kubanskogo gosudarstvennogo agrarnogo universiteta (Nauchnyj zhurnal KubGAU) [Jelektronnyj resurs]. – Krasnodar: KubGAU, 2021. – №09(173). S. 76 – 92. – IDA [article ID]: 1732109007. – Rezhim dostupa: <http://ej.kubagro.ru/2021/09/pdf/07.pdf>, 1,062 u.p.l.

3. Mihajlenko, E.V. Avtomatizacija obrabotki tablichnyh dannyh dlja sbora i analiza social'no-jekonomicheskikh pokazatelej sotrudnikami podrazdelenij organov vnutrennih del / E.V. Mihajlenko, A.A. Hromyh – Krasnodar: Krasnodarskij universitet MVD Rossii, 2021. 148 s.

4. Mihajlenko, E.V. Vvedenie v programmirovanie: ucheb. posobie / E.V. Mihajlenko – Krasnodar: Krasnodarskij universitet MVD Rossii, 2020. 109 s.

5. Mihajlenko, E.V. O razrabotke komp'juternyh programm dlja avtomatizacii praktičeskix zanjatij po razdelu «Teorija verojatnostej i matematičeskaja statistika» discipliny matematika / E.V. Mihajlenko // Problemy informacionnogo obespečenija dejatel'nosti pravoohranitel'nyh organov: sbornik statej 5-j Vserossijskoj nauchno-praktičeskoj konferencii. – Belgorod: Belgorodskij juridičeskij institut MVD Rossii imeni I.D. Putilina, 2019. – S. 27 – 45.

6. Mihajlenko, E.V. O razrabotke programmnyh prilozhenij dlja vypolnenija praktičeskix zadanij po razdelu «Metody optimizacii» discipliny prikladnaja matematika / E.V. Mihajlenko // Problemy informacionnogo obespečenija dejatel'nosti pravoohranitel'nyh organov: sbornik statej 4-j Vserossijskoj nauchno-praktičeskoj konferencii. – Belgorod: Belgorodskij juridičeskij institut MVD Rossii imeni I.D. Putilina, 2018. – S. 210 – 222.

7. Mihajlenko, E.V. Obshhie principy avtomatizacii podbora zadanij i proverki vypolnennyh kursantami rabot po matematičeskim disciplinam / E.V. Mihajlenko // Matematičeskie metody i informacionno-tehnicheskie sredstva: materialy XV Vseros. nauch.-prakt. konf. – Krasnodar: Krasnodarskij universitet MVD Rossii, 2019. C. 122 – 127.

8. Mihajlenko, E.V. Osobennosti razrabotki komp'juternyh prilozhenij dlja generacii i verifikacii praktičeskix zadanij, vključajushhix obrabotku matric / E.V. Mihajlenko // Problemy informacionnogo obespečenija dejatel'nosti pravoohranitel'nyh organov: sbornik statej 7-j Vserossijskoj nauchno-praktičeskoj konferencii. – Belgorod: Belgorodskij juridičeskij institut MVD Rossii imeni I.D. Putilina, 2020. – S. 3 – 20.

9. Mihajlenko, E.V. Principy avtomatizacii podbora i proverki praktičeskix zadanij, vključajushhix obrabotku polinomov / E.V. Mihajlenko // Matematičeskie metody i informacionno-tehnicheskie sredstva: materialy XVI Vseros. nauch.-prakt. konf. – Krasnodar: Krasnodarskij universitet MVD Rossii, 2020. C. 65 – 70.

10. Mihajlenko, E.V. Tehnologii podgotovki komp'juternyh programm dlja generacii i proverki praktičeskix zadanij po teme «Sravnenija» / E.V. Mihajlenko // Matematičeskie metody i informacionno-tehnicheskie sredstva: materialy XVII Vseros. nauch.-prakt. konf. – Krasnodar: Krasnodarskij universitet MVD Rossii, 2021. C. 101 – 111.

11. Mihajlenko, E.V. Tehnologii programmirovanija: ucheb. posobie / E.V. Mihajlenko, A.K. Nazarov – Krasnodar: Krasnodarskij universitet MVD Rossii, 2019. 510 s.

12. Starostenko I.N. K voprosu ob ispol'zovanii informacionnyh tehnologij pri organizacii samostojatel'noj raboty obučajushhixsja / I.N. Starostenko, E.V. Mihajlenko // Problemy sovremennogo pedagogičeskogo obrazovanija. – Jalta: RIO GPA, 2019. № 65-1. S. 257 - 262.