

УДК 519.688

НЕКОТОРЫЕ АСПЕКТЫ ПРИМЕНЕНИЯ СРЕДЫ VISUAL BASIC FOR APPLICATION ДЛЯ СОЗДАНИЯ УЧЕБНЫХ ПРИЛОЖЕНИЙ ПО МАТЕМАТИЧЕСКИМ ДИСЦИПЛИНАМ

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

В статье описываются методы организации автоматизированной подготовки заданий по математическим и естественнонаучным дисциплинам с помощью макросов в среде Visual Basic for Application. Предлагаются алгоритмы генерации данных для задач, методы проверки выполненных работ

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

UDC 519.688

SOME ASPECTS OF USING VISUAL BASIC FOR APPLICATION FOR CREATION EDUCATIONAL APPLICATIONS FOR MATHEMATICAL DISCIPLINES

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

This article describes the methods of organizing an automated job preparation in mathematics and natural science disciplines with macros among Visual Basic for Application. Showing data generation algorithms for the problems, methods of testing the work performed

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

При подготовке специалистов информационно-технических и экономических направлений для формирования общекультурных, профессиональных и профессионально-специализированных компетенций достаточно большое место уделяется изучению различных математических дисциплин. Особенность изучения таких дисциплин состоит в усвоении обучающимися теоретического материала, получаемого на лекциях, а также в приобретении устойчивых навыков решения типовых задач по каждой из изученных тем.

В статьях [1], [5], [7], [8] уже подняты вопросы оптимизации образовательного процесса с использованием информационных технологий. Настоящая статья является попыткой описать некоторые методические приемы и технологии использования возможностей среды Visual Basic for Application для создания учебных программ.

Прежде всего, поясним, что выбор среды программирования не случаен. Нас как разработчиков привлек мощный математический аппарат VBA, доступность широких возможностей использования различных шрифтов, графических форм и объектов, простота и логичность синтаксиса и семантики языка. Не менее важным является доступность пакета программ Microsoft Office как для преподавателей, создающих исполняемые модули, так и для обучаемых. При написании программы использовались методы и технологии VBA, описанные в работах [2], [3], [4].

Основная идея описываемых технологий заключается в использовании приложения MS Office, где на заранее подготовленную форму автоматически по специальному алгоритму выводятся сгенерированные данные в виде комбинации чисел, символов или графических объектов. Приложением может быть, например, редактор MS Office Word, табличный процессор MS Excel или MS PowerPoint. Заполненную таким образом форму можно теперь довести до обучающегося: поместить файл с задачами на портале учебного заведения во время аудиторных занятий, отправить его по электронной почте или предоставить доступ к заданиям в облаке, организовав самостоятельную работу. Решая представленные задачи, обучаемый заполняет ответами предоставленный документ, сохраняет его, после чего отправляет документ для проверки. Затем работу открывает преподаватель. Используя макрос проверки, доступный только преподавателю, осуществляется компьютерная проверка выполненных заданий: автоматически зачеркиваются неверные ответы, подсчитываются набранные баллы, выставляется оценка за выполненную работу.

На рисунке 1 показан пустой шаблон без заданий на практическое занятие «Делимость чисел» по теме «Теория чисел» дисциплины математика.



Теория чисел
Практическое задание
Делимость чисел

Дата _____
 Время _____
 (Фамилия И.О.)

Вариант

1. Найти 6 делителей числа.

делители:

2. Найти 6 кратных числа.

кратные:

3. Представить целое число a через число b в виде $a = bq + r$ ($0 \leq r < b$).

a) б)

4. Применяя метод Евклида, найти НОД чисел и выписать 5 первых остатков.

a) $r_1 =$ $r_2 =$ $r_3 =$ $r_4 =$ $r_5 =$

б) $r_1 =$ $r_2 =$ $r_3 =$ $r_4 =$ $r_5 =$

5. Выписать все простые числа из промежутка [;].

6. Записать каноническое представление чисел.

a) б)

7. Используя канонические разложения чисел найти их НОД и НОК.

a)

б)

8. Найти канонические разложения факториалов.

a) б)

9. Найти количество делителей τ и сумму делителей σ чисел.

<input style="width: 70px; height: 15px;" type="text"/>			
<input style="width: 70px; height: 15px;" type="text"/>			

Критерий оценки:
31-33 баллов - "отлично"
26-30 баллов - "хорошо"
22-25 баллов - "удовлетворительно"
менее 22 баллов - "неудовлетворительно"

Количество набранных баллов: _____

Оценка за работу: _____

Проверка

Рис.1 Лист с пустым шаблоном без заданий.

Для осуществления генерации заданий обучаемому необходимо заполнить поле «Фамилия и инициалы». Введенные им данные, а также дата и время запуска генерации, автоматически выводимые на лист, используются для расчета номера варианта. Ниже представлен алгоритм вычислений номера варианта, реализованный на VBA.

```
Function codeName(Name)
    k1 = 0
    codeName = 0
    For i = 1 To Len(Name)
        k = ((Asc(Mid(Name, i, 1)) - 192) 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
v = timer Mod 31000 + codeName(Name)
```

В дальнейшем, при автоматизированной проверке работы, производится верификация соответствия номера варианта указанным в работе времени и фамилии тестируемого. Данная процедура позволяет избежать подлогов выполненных работ, и при нестыковке всех трех компонентов: варианта, времени и записи, идентифицирующей обучаемого, проверка работы не проводится и на листе появляется запись

«Вариант не соответствует фамилии и инициалам!».

Генерация параметров каждого задания в предлагаемых работах осуществляется по разработанным алгоритмам, обязательно включающим номер варианта. Приведем пример реализации алгоритма для первого задания.

```
' 1. Найти 6 делителей числа a
Do
  a = 1
  For i = 1 To 4
    a = a * Rnd(v) * 9 + 3
  Next i
Loop Until a > 100 And a < 1000
t = " 1. Найти 6 делителей числа " + st(a) + "."
Cells(8, 1) = t
Cells(8, 1).Characters(13, 1).Font.ColorIndex = 3
Cells(8, 1).Characters(32, 3).Font.ColorIndex = 3
```

В показанном алгоритме в цикле *Do ... loop until* в диапазоне [100, 1000] определяется значение числа a . Как видно из программы, генерируемое значение задается псевдослучайным образом, используя в качестве параметра функции Rnd номер варианта v . Переменная a объявлена в начале программы, При использовании данного алгоритма количество целых делителей числа a всегда будет не менее шести. Затем в строковой переменной t составляется условие задания, и полученное выражение выводится в ячейку A8 рабочего листа. Две последних строки фрагмента программы определяют цвет шрифта выводимых числовых данных.

Генерация второго задания реализована линейным алгоритмом.

```
' 2. Найти 6 кратных числа a
a = Rnd(v) * 70 + 20
t = " 2. Найти 6 кратных числа " + st(a) + "."
Cells(13, 1) = t
```

```
Cells(13, 1).Characters(13, 1).Font.ColorIndex = 3
```

```
Cells(13, 1).Characters(29, 2).Font.ColorIndex = 3
```

В данном фрагменте значение переменной a определяется в диапазоне [20, 90].

При определении условий заданий 3.а и 3.б «Представить число a через натуральное число b в виде $a = bq + r$ » генерируются одновременно параметры a , b , q и r . Покажем алгоритм данной генерации.

```
Do
```

```
  b = Rnd(v) * 40 + 10
```

```
  q = Rnd(v) * 20 + 10
```

```
  r = Rnd(v) * (b - 2) + 1
```

```
  a = b * q + r
```

```
Loop Until a < 1000
```

```
t = "a = " + st(a) + "; b = " + st(b): Cells(20, 2) = t
```

Необходимо обратить внимание на то, что в данной структуре переменная r зависит от значения b , а для расчета числа a используются значения всех заданные ранее значений.

Четвертое задание «4. Применяя метод Евклида, найти НОД чисел и выписать 5 первых остатков» посвящено использованию алгоритма Евклида для нахождения наибольшего общего делителя двух чисел.

```
Do
```

```
  nod = Rnd(v) * 20 + 15: a = nod: r = 0
```

```
For i = 1 To 6
```

```
  b = a
```

```
  If i = 1 Then q = Rnd(v) * 3 + 2 Else q = Rnd(v) * 4 + 1
```

```
  a = b * q + r
```

```
  r = b
```

```
Next i
```

```
Loop Until a > 1000 And b < 1000
```

$t = "(" + st(b) + ", " + st(a) + ") = "$: Cells(25, 4) = t

Следует заметить, что фрагмент программы выполняет преобразования обратные алгоритму Евклида [5]. Представленный алгоритм реализован в виде двух вложенных циклов. Сначала определяется наименьший общий делитель nod , затем в цикле For ... Next на каждом из шести шагов итерационного процесса переменной b присваивается значение a , генерируются частные q , по формуле $a = bq + r$ увеличивается коэффициент a , а остаток от деления r становится равным b . Циклический процесс продолжается до тех пор, пока не выполнится конъюнкция условий $a > 1000$ And $b < 1000$. Представленный алгоритм гарантирует ненулевые остатки при делении a на b . Кроме того, в процессе формирования чисел мы получим и все промежуточные значения r , которые могут быть использованы при проверке заполненной обучающимся формы.

При генерации задания «5. Выписать все простые числа из промежутка .. » определение значений границ $[a, b]$ осуществляется достаточно просто и приводить полностью программный код для этого задания мы не будем. Однако отметим, что необходимо провести проверку на то, чтобы такие простые числа нашлись.

```
n=0
For i = a To b
  m = 0
  For j = 2 To Sqr(i)
    If i Mod j = 0 Then m = m + 1
  Next j
  If m = 0 Then n=n+1
Next i
```

Данный фрагмент программы подсчитывает количество простых чисел. Поэтому целесообразно повторять процесс генерации границ интерва-

ла до тех пор, пока в результате очередной итерации количество простых чисел n не будет находиться в разумных пределах.

Расчет параметров шестого задания «Записать каноническое представление чисел» основан на определении количества простых множителей, входящих в канонические представления получаемых чисел. Для того, чтобы отображаемые значения находились в разумных пределах, можно брать только множители 2, 3, 5, 7, 11, 13, 17 с соответствующими степенями $s_2, s_3, s_5, s_7, s_{11}, s_{13}, s_{17}$. Однако в отображенном ниже алгоритме процесс подбора степеней повторяется пока не выполнится условие $a \in [2000, 20000]$.

Do

$$s_2 = \text{Rnd}(v) * 5$$

$$s_3 = \text{Rnd}(v) * 4$$

$$s_5 = \text{Rnd}(v) * 3$$

$$s_7 = \text{Rnd}(v) * 2$$

$$s_{11} = \text{Rnd}(v) * 2$$

$$s_{13} = \text{Rnd}(v) * 2$$

$$s_{17} = \text{Rnd}(v) * 2$$

$$a = 2^{s_2} * 3^{s_3} * 5^{s_5} * 7^{s_7} * 11^{s_{11}} * 13^{s_{13}} * 17^{s_{17}}$$

Loop Until $a < 20000$ And $a > 20000$

В задании «7. Используя канонические разложения чисел, найти НОК и НОД» процесс нахождения a и b начинается с определения наибольшего общего делителя nod этих двух чисел. Алгоритмическая структура основана на обратном ходе метода Евклида. Она будет повторяться неоднократно до тех пор, пока в результате очередной итерации одновременно не выполнятся условия $a > 1000$ и $b < 1000$.

Do

$$nod = \text{Rnd}(v) * 20 + 15$$

$$a = nod$$

```
r = 0
For i = 1 To 6
  b = a
  If i = 1 Then q = Rnd(v) * 3 + 2 Else q = Rnd(v) * 4 + 1
  a = b * q + r
  r = b
Next i
Loop Until a > 1000 And b < 1000
```

Генерация заданий «8. Найти канонические разложения факториалов» и «9. Найти количество делителей τ и сумму делителей σ чисел» не представляет особой сложности. Выводимые числа определяются в заданных промежутках, и особых ограничений для их расчета нет. Например, для генерации числа $n \in [a, b]$ можно использовать запись

$$n = \text{Rnd}(v) * (b - a) + a$$

Процесс заполнения формы у обучаемых обычно не вызывает трудностей. Ответы вводятся в заранее отведенные поля, доступные только для изменений. Правила ввода ответов решенных задач заранее оговариваются и зависят от сложности алгоритмов обработки данных при проверке заданий. После выполнения всей работы заполненный файл необходимо сохранить.

На рисунке 2 отображен лист со сгенерированными задачами. После вывода на лист всех заданий лист защищается. Каждая работа, сгенерированная подобным образом, является уникальной, так как число вариантов работ достигает 32000. При необходимости можно изменить алгоритм обработки времени и идентификационных записей и увеличить количество вариантов.



Теория чисел
Практическое задание
Делимость чисел

Дата 11.11.2014
 Время 15:09:20
Иванов А.В.
 (Фамилия И.О.)

Вариант **23765**

1. Найти 6 делителей числа 399.

делители:

2. Найти 6 кратных числа 22.

кратные:

3. Представить целое число a через число b в виде $a = bq + r$ ($0 \leq r < b$).

а) $a = 512; b = 35$ б) $a = 566; b = 25$

4. Применяя метод Евклида, найти НОД чисел и выписать 5 первых остатков.

а) $(864, 2394) =$ $r_1 =$ $r_2 =$ $r_3 =$ $r_4 =$ $r_5 =$

б) $(1312, 944) =$ $r_1 =$ $r_2 =$ $r_3 =$ $r_4 =$ $r_5 =$

5. Выписать все простые числа из промежутка [167; 201].

6. Записать каноническое представление чисел.

а) $8415 =$ б) $10472 =$

7. Используя канонические разложения чисел найти их НОД и НОК.

а) $(736, 4304) =$ $[736, 4304] =$

б) $(810, 2727) =$ $[810, 2727] =$

8. Найти каноническое разложения факториалов.

а) $22! =$ б) $34! =$

9. Найти количество делителей τ и сумму делителей σ чисел.

$\tau(29) =$ $\tau(184) =$ $\tau(589) =$ $\tau(1073) =$

$\sigma(29) =$ $\sigma(184) =$ $\sigma(589) =$ $\sigma(1073) =$

<p>Критерий оценки: 31-33 баллов - "отлично" 26-30 баллов - "хорошо" 22-25 баллов - "удовлетворительно" менее 22 баллов - "неудовлетворительно"</p>
--

Количество набранных баллов: _____

Оценка за работу: _____

Проверка

Рис.2 Лист после вывода в шаблон сгенерированных заданий.

Проверка выполненных работ осуществляется преподавателем либо самим курсантом. Для проверки создается специальный макрос, анализирующий заполненные обучающимся поля. Для проверки работ самими курсантами файл с проверкой временно помещается в специальную защищенную от просмотра и копирования область на сервере кафедры и автоматически удаляется после окончания каждой пары. Определяя вариант работы, программа заново генерирует все исходные данные, на основе их рассчитывает решение задачи. В ряде случаев, таких как приведенный пример, решение некоторых задач уже известно. Значения каждого поля сравниваются решением. Если заполненные поля удовлетворяют запросам программы, за каждое верное решение начисляются баллы, в противном случае поля зачеркиваются. Оценивание работы происходит согласно критерию, описанного в нижней левой части листа.

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. Лаптев В.Н. К алгоритму усвоения знаний, умений и навыков на базе информационных технологий / В.Н. Лаптев, И.А. Василенко. Научный журнал КубГАУ. [Электронный ресурс]. – Краснодар: КубГАУ, 2009. №43(09). – 16с. <http://ej.kubagro/2009/09/pdf/07.pdf>.
2. Кузьменко В.Г. ВВА. – М.: ООО «Бином-Пресс», 2012. Михайленко Е.В. Информатика: учебное пособие / Е.В. Михайленко, И.Н. Старостенко, Ю.Н. Сопильняк. – Краснодар: Краснодарский университет МВД России, 2010.
3. Михайленко Е.В. Математика и информатика: курс лекций / Е.В. Михайленко, И.Н. Старостенко. – Краснодар: Краснодарский университет МВД России, 2009.
4. Михайленко Е.В. Математические модели рейтинговых методик. / Вестник Краснодарской академии МВД России. – Краснодар: Краснодарская академия МВД России, 2005, №2, С. 7-10.
5. Михайленко Е.В. Прикладная математика: курс лекций / Е.В. Михайленко, А.А. Хромых. – Краснодар: Краснодарский университет МВД России, 2014.
6. Михайленко Е.В. Создание автоматизированной системы для ведения мониторинга работы профессорско-преподавательского состава. / Математические методы и информационно-технические средства: труды VI Всерос. науч.-практ. конф. – Краснодар: Краснодарский университет МВД России, 2008, С. 147-149.
7. Старостенко И.Н. О некоторых аспектах оптимизации образовательного процесса с использованием информационных технологий. / Математические методы и информационно-технические средства: труды VI Всерос. науч.-практ. конф. – Краснодар: Краснодарский университет МВД России, 2010, С. 183-189.

References

1. Laptev V.N. K algoritmu usvoenija znaniy, umeniy i navykov na baze informacionnyh tehnologij / V.N. Laptev, I.A. Vasilenko. Nauchnyj zhurnal KubGAU. [Jelektronnyj resurs]. – Krasnodar: KubGAU, 2009. №43(09). – 16s. <http://ej.kubagro/2009/09/pdf/07.pdf>.
2. Kuz'menko V.G. VBA. – M.: ООО «Binom-Press», 2012. Mihajlenko E.V. Informatika: uchebnoe posobie / E.V. Mihajlenko, I.N. Starostenko, Ju.N. Sopil'njak. – Krasnodar: Krasnodarskij universitet MVD Rossii, 2010.
3. Mihajlenko E.V. Matematika i informatika: kurs lekcij / E.V. Mihajlenko, I.N. Starostenko. – Krasnodar: Krasnodarskij universitet MVD Rossii, 2009.
4. Mihajlenko E.V. Matematicheskie modeli rejtingovyh metodik. / Vestnik Krasnodarskoj akademii MVD Rossii. – Krasnodar: Krasnodarskaja akademija MVD Rossii, 2005, №2, S. 7-10.
5. Mihajlenko E.V. Prikladnaja matematika: kurs lekcij / E.V. Mihajlenko, A.A. Hromyh. – Krasnodar: Krasnodarskij universitet MVD Rossii, 2014.
6. Mihajlenko E.V. Sozdanie avtomatizirovannoj sistemy dlja vedenija monitoringa raboty professorsko-prepodavatel'skogo sostava. / Matematicheskie metody i informacionno-tehnicheskie sredstva: trudy VI Vseros. nauch.-prakt. konf. – Krasnodar: Krasnodarskij universitet MVD Rossii, 2008, S. 147-149.
7. Starostenko I.N. O nekotoryh aspektah optimizacii obrazovatel'nogo processa s ispol'zovaniem informacionnyh tehnologij. / Matematicheskie metody i informacionno-tehnicheskie sredstva: trudy VI Vseros. nauch.-prakt. konf. – Krasnodar: Krasnodarskij universitet MVD Rossii, 2010, S. 183-189.