



РАЗРАБОТКА МЕТОДИЧЕСКИХ МАТЕРИАЛОВ В СРЕДЕ СИСТЕМЫ MATHEMATICA

Сосновский Николай Николаевич

Аннотация

Преподавателю необходимо иметь своего рода генератор задач, цель которых проверить понимание излагаемого материала и помочь студентам приобрести навыки решения типовых задач. В статье рассматривается генератор задач дискретной математики, разработанный в среде системы Математика. Сгенерированные файлы заданий доводятся до студентов по электронной почте. Генератор обеспечивает формирование индивидуального задания каждому студенту группы. Студент выбирает вариант по своему номеру в списке группы, который также доводится до студентов по электронной почте.

Ключевые слова: генератор задач, *Mathematica*, индивидуальные практические задания, дискретная математика.

Важным элементом процесса преподавания математики является разработка методических материалов для студентов. Автор исходит из того, что процесс освоения математики должен быть активным. Студент должен выполнить обязательный минимум упражнений, для того чтобы достигнуть определенного уровня понимания материала и закрепить его в навыках. Предлагаемые упражнения бывают двух типов. Первый тип упражнений — это достаточно простые упражнения (задачи), цель которых проверить понимание излагаемого материала и помочь студентам приобрести необходимые навыки решения типовых задач. Второй тип упражнений — это упражнения, требующие искусства для их выполнения. Такие упражнения предназначены для углубленного понимания математики и могут быть осилены не всеми студентами. Разработка таких упражнений не может быть осуществлена стандартными способами и требует творческого подхода от преподавателя. Что же касается упражнений первого типа, то, напротив, разработка их может и должна быть автоматизирована, так как здесь требуется приготовить большое количество задач одного типа, чтобы обеспечить выдачу каждому студенту индивидуального задания, по результатам проверки которого можно сделать вывод о степени усвоения материала. Так как это индивидуальные практические задания, то их много, и, следовательно, необходимо обеспечить наличие ответов для упрощения проверки и оценки результатов. Сказанное говорит о том, что преподавателю необходимо иметь своего рода генератор задач для обеспечения формирования индивидуальных практических заданий по излагаемому материалу.

В СПбГЭТУ (ЛЭТИ) имеется опыт разработки таких генераторов [1, 2]. В данной статье рассматривается генератор задач дискретной математики, разработанный в среде системы **Mathematica**. Целью автора является показать, что создание такого генерато-

ра вполне посылно математику-непрограммисту, освоившему **Mathematica** на уровне пользовательского интерфейса.

На рис. 1 приводится пример вариантов заданий и ответов по дискретной математике, формируемых таким генератором.

<p>Вар.№01(3111)</p> <p>1) Решить диофантово уравнение $4876x + 1909y = -115$. Ответ записать в виде: $x=x_0+k \cdot t, y=y_0+h \cdot t$; x_0-наименьшее положительное число.</p> <p>2) Представить $\sqrt{140}$ в виде периодической цепной дроби.</p> <p>3) Найти представление рационального числа $\frac{734}{327}$ непрерывной дробью.</p> <p>4) Вычислить $\frac{29}{26}$ в кольце вычетов по модулю 57.</p> <p>5) Решить уравнение, записанное в 5-ичной системе счисления: $1200 + 24x = 4021$. Решение записать в десятичной и 5 - ичной системах.</p> <p>6) Найти наименьшее натуральное число, удовлетворяющее условиям $x \equiv 4 \pmod{23}, x \equiv 3 \pmod{17}, x \equiv 7 \pmod{41}$.</p> <p>7) Найти остаток от деления $77^{13^{74}}$ на 90.</p>	<p>Вар.№02(3111)</p> <p>1) Решить диофантово уравнение $1749x + 1177y = 77$. Ответ записать в виде: $x=x_0+k \cdot t, y=y_0+h \cdot t$; x_0-наименьшее положительное число.</p> <p>2) Представить $\sqrt{176}$ в виде периодической цепной дроби.</p> <p>3) Найти представление рационального числа $\frac{965}{434}$ непрерывной дробью.</p> <p>4) Вычислить $\frac{45}{11}$ в кольце вычетов по модулю 94.</p> <p>5) Решить уравнение, записанное в 8-ичной системе счисления: $371 + 16x = 1073$. Решение записать в десятичной и 8 - ичной системах.</p> <p>6) Найти наименьшее натуральное число, удовлетворяющее условиям $x \equiv 22 \pmod{11}, x \equiv 26 \pmod{23}, x \equiv 11 \pmod{17}$.</p> <p>7) Найти остаток от деления $5^{17^{78}}$ на 86.</p>
<p>Вар.№01(3111)</p> <p>1) $x = 45 - 83t$ $y = -115 + 212t$</p> <p>2) {11, {1, 4, 1, 22}}</p> <p>3) {2, 4, 11, 2, 3}</p> <p>4) 34</p> <p>5) {{24, 10}, {44, 5}}</p> <p>6) 2672</p> <p>7) 77</p>	<p>Вар.№02(3111)</p> <p>1) $x = 31 - 107t$ $y = -46 + 159t$</p> <p>2) {13, {3, 1, 3, 26}}</p> <p>3) {2, 4, 2, 9, 5}</p> <p>4) 81</p> <p>5) {{23, 10}, {27, 8}}</p> <p>6) 946</p> <p>7) 5</p>

Рис. 1

Рассмотрим генератор по частям. Первая часть представлена на рис. 2.

Сначала мы видим две процедуры: `intstr[m_]` и `sp[n_]`.

Первая процедура предназначена для формирования чисел определенной длины в текстовом формате. В данном случае это необходимо, чтобы получать номера билетов в виде текстовой строки длины 2 для чисел от 1 до 99, то есть третий билет будет иметь номер 03.

Вторая процедура формирует строку пробелов заданной длины, в данном случае 35.

Далее идут операторы, создающие таблицы данных, используемые генератором в процессе работы. Так, `doma1`, `domb1` и `domd`, — это три непересекающихся диапазона простых чисел.

Таблица `domQ` содержит целые числа, разложения которых в цепную дробь имеют подходящую длину, в данном случае от 4 до 6.

В таблицу `domR` отобраны несократимые рациональные дроби, обладающие подобным свойством.

```
(*Процедура формирования m порядковых целых чисел в текстовом формате*)
intstr[m_] := (sn = Table[0, {m}]; Do[sn[[i]] = IntegerString[i]; If[i < 10, sn[[i]] = StringJoin["0", sn[[i]]],
    {i, 1, m}]);
intstr[100];
(*Процедура формирования строки из n пробелов*)
sp[n_] := (spn := ""; Do[spn = StringJoin[spn, " "], {i, 1, n}]);
sp[35];
doma1 = Table[Prime[n], {n, 13, 20}];
domb1 = Table[Prime[n], {n, 21, 30}];
domd = Table[Prime[n], {n, 5, 9}];
domQ = {119, 125, 128, 140, 155, 167, 173, 176, 185, 194, 215, 218, 248, 254, 266, 287, 293, 299, 305,
    320, 335, 338, 359, 365, 368, 377, 392, 395, 398};
domR = {
    219/76, 652/329, 993/338, 634/433, 476/169, 821/328, 250/137, 745/318, 102/53, 683/324, 928/455, 443/193, 931/370, 135/74, 220/83, 799/412, 90/53, 638/415,
    109/53, 719/324, 817/328, 327/226, 179/73, 751/306, 659/309, 355/218, 436/171, 945/326, 401/192, 359/216, 323/154, 216/133, 629/302, 668/411, 127/73, 695/332,
    191/87, 166/67, 650/347, 79/29, 37/17, 945/317, 974/451, 215/134, 706/309, 606/337, 178/77, 410/187, 962/441, 841/314, 99/43, 727/332, 83/32, 189/74, 761/313,
    734/327, 949/334, 466/209, 827/399, 848/433, 958/411, 741/332, 97/57, 228/83, 73/45, 907/317, 759/359, 965/434, 305/106, 363/218, 219/131, 165/73};
```

Рис. 2

Отбор был сделан предварительно с помощью операторов цикла, использующих в качестве условия указанное ограничение на длину.

Заметим, в качестве примера, что формирование диапазонов чисел происходит с помощью команды `Table[Prime[n], n, 13, 20]`, которая в данном случае создает диапазон простых чисел с номерами от 13 до 20.

Следующая часть генератора представляет собой процедуру, которая состоит из последовательности программ, генерирующих условия и ответы к задачам соответствующего типа, входящих в билет.

Рассмотрим эти программы последовательно с тем, чтобы представить используемые операторы и оценить сложность программ.

Программа для первой задачи представлена на рис. 3.

```
(*задача 1*)
a1 = RandomChoice[doma1];
b1 = RandomChoice[domb1];
d = RandomChoice[domd];
a = a1 d RandomInteger[{1, 5}];
b = b1 d RandomChoice[{1, -1}];
c = d RandomInteger[{3, 10}] RandomChoice[{1, -1}];
Clear[x, y];
usl1 = StringForm["`", a x + b y == c];
{{x0, y0}} = {x, y} /. FindInstance[{a x + b y == c, 0 <= x <= Abs[b/d]}, {x, y}, Integers];
ans1 = StringForm["x = `1`
y = `2`", x0 - b/d t, y0 + a/d t];
```

Рис. 3

Первые строки программы — это генерация с помощью операторов `RandomChoice[]` и `RandomInteger[]` коэффициентов уравнения. Затем с помощью команд `StringForm[]`

FindInstance[] происходит формирование текста условия и ответа к задаче. Программа для второй задачи представлена на рис. 4.

```
(*задача 2*)
q = RandomChoice[domQ];
us12 = StringForm["√q", q];
ans2 = ContinuedFraction[√q];
```

Рис. 4

Первый оператор — это случайный выбор числа. Второй и третий оператор формируют условие и ответ к задаче на разложение корня квадратного из целого числа в непрерывную дробь. Как видим, используются два оператора: StringForm [] и ContinuedFraction [].

Программа для третьей задачи представлена на рис. 5.

```
(*задача 3*)
us13 = RandomChoice[domR];
ans3 = ContinuedFraction[us13];
```

Рис. 5

Здесь осуществляется выбор из диапазона рациональных дробей и формирование ответа.

Программа для четвертой задачи представлена на рис. 6.

```
(*задача 4*)
m = RandomInteger[{51, 99}];
lm = EulerPhi[m];
t1m = Table[0, {lm}];
s1 = 0; Do[If[GCD[k1, m] == 1, (s1 = s1 + 1; t1m[[s1]] = k1)], {k1, 1, m - 1}];
k = RandomChoice[t1m];
If[k == 1, k = t1m[[2]]];
sol = FindInstance[k x + m y == 1 && 0 < x < m, {x, y}, Integers];
{a} = x /. sol;
b = RandomChoice[t1m];
While[GCD[b, k] != 1 ∨ Mod[b/k, 1] == 0, b = RandomChoice[t1m]];
us14 = b/k;
ans4 = Mod[a b, m];
m4 = m;
```

Рис. 6

Алгоритм следующий:

1. Случайно выбираем целое число m , у которого достаточно большая система приведенных вычетов.
2. Далее формируем эту таблицу приведенных вычетов $t1m$ и выбираем из нее случайное число k , не равное 1.

3. Затем решаем сравнение $kx \equiv 1 \pmod{m}$. Запоминаем решение — число a .
4. Далее выбираем число b так, чтобы числа k и b были взаимно простыми, и для которого b/k не является целым.
5. В условии пишем рациональную дробь b/k , в ответ пишем остаток от деления числа ab на m .

Программа для пятой задачи представлена на рис. 7.

```
(*задача 5*)
a = RandomInteger[{5, 19}];
b = RandomInteger[{101, 299}];
u = RandomInteger[{11, 29}];
c = a u + b;
q = RandomChoice[{2, 3, 4, 5, 6, 7, 8, 9}];
aq = IntegerString[a, q];
bq = IntegerString[b, q];
uq = IntegerString[u, q];
cq = IntegerString[c, q];
us15 = StringForm["`", aq x + bq == cq];
ans5 = {{u, 10}, {uq, q}};
```

Рис. 7

Алгоритм

1. Выбираем случайно целые числа a , b , u и вычисляем число c : $c = au + b$.
2. Наугад выбираем основание системы счисления q из списка $\{2, 3, 4, 5, 6, 7, 8\}$.
3. Переводим затем числа a , b , u и c в q -ичную систему счисления и формируем условие и ответ к задаче.

Программа для шестой задачи представлена на рис. 8.

```
(*задача 6*)
c = RandomSample[Range[0, 40], 3];
domr = {11, 13, 17, 19, 23, 29, 31, 37, 41, 43};
domr1 = {10, 14, 15, 21, 25, 35, 49};
e = RandomChoice[domr1];
AppendTo[domr, e];
m = RandomSample[domr, 3];
us16 = StringForm["x ≡ `` mod `` , x ≡ `` mod `` , x ≡ `` mod ``", c[[1]], m[[1]], c[[2]], m[[2]],
  c[[3]], m[[3]]];
ans6 = ChineseRemainder[c, m];
```

Рис. 8

Алгоритм

1. Выбираем 3 целых числа без повторения из диапазона $\{0, 1, \dots, 40\}$.
2. Формируем диапазон для выбора трех попарно простых чисел — модулей сравнений m_1, m_2, m_3 . Делаем случайную выборку без повторения 3 чисел.
3. Формируем условие и ответ к задаче.

Программа для седьмой задачи представлена на рис. 9.

```
(*задача 7*)
m = RandomInteger[{31, 99}];
lm = EulerPhi[m];
t1m = Table[0, {k, 1, lm}];
s1 = 0; Do[If[GCD[k1, m] == 1, (s1 = s1 + 1; t1m[[s1]] = k1)], {k1, 1, m - 1}];
na = RandomInteger[{2, lm}];
a = t1m[[na]];
lm1 = EulerPhi[lm];
t1m1 = Table[0, {k, 1, lm1}];
s1 = 0; Do[If[GCD[k1, lm] == 1, (s1 = s1 + 1; t1m1[[s1]] = k1)], {k1, 1, lm - 1}];
nb = RandomInteger[{2, lm1}];
b = t1m1[[nb]];
t1mc = Table[If[GCD[k, a] == 1 && GCD[k, b] == 1, k, 0], {k, 1, 125}];
t1mc = Select[t1mc, # > 0 &];
lc = Length[t1mc];
nc = RandomInteger[{2, lc}];
c = t1mc[[nc]];
b1 = PowerMod[b, c, lm];
xm = PowerMod[a, b1, m];
us17 = StringForm["`1`^`2`^`3`", a, b, c];
ans7 = xm;
```

Рис. 9

Алгоритм

1. Выбираем случайное целое число m из диапазона $\{39, \dots, 99\}$.
2. Строим приведенную систему вычетов $t1m$ для этого числа.
3. Выбираем случайное целое число a из $t1m$, не равное 1.
4. Строим приведенную систему вычетов $t1m1$ для этого числа.
5. Выбираем случайное целое число b из $t1m1$, не равное 1.
6. Строим таблицу $t1mc$ чисел, попарно простых с a и b .
7. С помощью операторов $\text{PowerMod}[b, c, lm]$ и $\text{PowerMod}[a, b1, m]$ вычисляем значение искомой степени.
8. Формируем условие и ответ к задаче.

Далее происходит формирование текста билета к заданному варианту и текста ответов к нему.

Формат текста условия представлен на рис. 10.

Формат текста ответов представлен на рис. 11.

Рассмотрим теперь, как работает генератор. На рис. 12 представлена программа формирования файлов билетов и ответов к ним.

Программа оформлена в виде процедуры с двумя параметрами: число вариантов и номер группы.

Число вариантов округляется до ближайшего сверху целого числа.

```

(*формирование билета и ответа к нему*)
us1 = StringForm[
  "Вар.№`1``2``3`
1) Решить диофантово уравнение
`4`.
Ответ записать в виде:  $x=x_0+k \cdot t, y=y_0+h \cdot t$ ;
 $x_0$ -наименьшее положительное число.
2) Представить `5` в виде периодической цепной
дроби.
3) Найти представление рационального числа `6`
непрерывной дробью.
4) Вычислить `7` в кольце вычетов по модулю `8`.
5) Решить уравнение, записанное в `9`-ичной системе
счисления: `10`.
Решение записать в десятичной и `9`-ичной
системах.
6) Найти наименьшее натуральное число,
удовлетворяющее условиям
`11`.
7) Найти остаток от деления `12` на `13`.", sn[[vr]], ngr, spn, us11, us12, us13, us14,
m4, q, us15, us16, us17, m];

```

Рис. 10

```

ans = StringForm[
  "Вар.№`1``2``3`
1) `4`
2) `5`
3) `6`
4) `7`
5) `8`
6) `9`
7) `10`", sn[[vr]], ngr, spn, ans1, ans2, ans3, ans4, ans5, ans6, ans7
];

```

Рис. 11

```

prn[nvr_, ngr_] :=
(If[Mod[nvr, 2] == 1, nv = nvr + 1, nv = nvr];
cnd = Table[0, {str, 1, nv/2}, {nzd, 1, 2}, {tip, 1, 2}];
Do[{k = Mod[2 s - 1, 99, 1]; przd[k, ngr];
cnd[[s, 1, 1]] = us1; cnd[[s, 1, 2]] = ans;
k = Mod[2 s, 99, 1]; przd[k, ngr];
cnd[[s, 2, 1]] = us1; cnd[[s, 2, 2]] = ans;}, {s, 1, nv/2}];
Do[Print[Grid[{{cnd[[s, 1, 1]], cnd[[s, 2, 1]]}}, Frame -> All, Spacings -> {1, 1}], {s, 1, nv/2}];
Do[Print[Grid[{{cnd[[s, 1, 2]], cnd[[s, 2, 2]]}}, Frame -> All, Spacings -> {1, 1}], {s, 1, nv/2}];)

```

Рис. 12

Строится трехмерная таблица `snd`, первоначально пустая. Каждый элемент этой таблицы соответствует паре задач, печатаемых на листе рядом. Он состоит из двух элементов — для первой и второй задачи. Элемент каждой задачи, в свою очередь, делится на два элемента. В первый из последних элементов записывается текст условия, а во второй — текст ответа для первой или второй задачи данной строки таблицы.

Заполнение таблицы `snd` происходит в цикле по числу строк файла, то есть по числу пар задач, печатаемых на листе рядом. Для каждой строки последовательно два раза запускается процедура `przd[vr_, ngr_]`, параметрами которой являются номер варианта и номер группы. Каждый вариант состоит из одних и тех же задач, но так как процедура каждой задачи включает генерацию случайных чисел, то варианты с разными номерами состоят из задач с различающимися исходными данными.

Две последние строки программы это циклы формирования (печати) билетов и ответов к ним.

Список литературы

1. Степанов А.В. Система компьютерной генерации задач по математике // Компьютерные инструменты в образовании, 2000. № 3-4. С. 28–31.
2. Посов И.А. Автоматическая генерация задач // Компьютерные инструменты в образовании, 2007. № 1. С. 54–62

DEVELOPMENT OF METHODOICAL MATERIALS IN THE ENVIRONMENT OF "MATHEMATICA"

Sosnovski N. N.

Abstract

The teacher must have a kind of generator of tasks aimed to check understanding of the material and to help students to acquire the skills of solving typical problems. The article discusses the task generator of discrete mathematics, developed in the environment of the system of Mathematics. The generated files of the tasks are communicated to students via e-mail. The generator ensures the formation of individual assignments to each student of group. The student selects an option by its number in the list group, which is also communicated to students via e-mail.

Keywords: *generator of tasks, Mathematica, individual practical tasks, discrete mathematics..*

© Наши авторы, 2015.
Our authors, 2015.

Сосновский Николай Николаевич,
кандидат технических наук, старший
научный сотрудник СПбГЭТУ (ЛЭТИ), доцент
кафедры высшей математики ВМ-2,
sosnovskiy-78@yandex.ru