

Генератор узоров «Каракули»

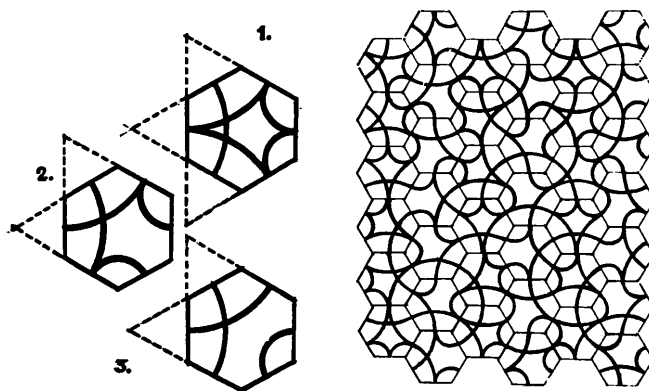
Для версий:
Microsoft Office 2016
LibreOffice 7.0

Лет эдак тридцать тому назад дома у моей бабушки был выпуск «Науки и жизни», который почему-то запомнился скромной по объёму публикацией, в коей рассказывалось про составление витиеватых узоров при помощи сложенных вместе картонных шестиугольников с нанесённым на каждый простым рисунком из пары-тройки линий. Тот номер журнала давным-давно куда-то пропал без следа, не оставив в памяти сведений ни о годе, ни тем более о месяце своего издания. Забегая немного вперёд, скажу, что до недавнего времени так оно и было.

В 2010 году лето выдалось на редкость жарким и засушливым. В ту пору на сайте Яндекса висела забавная безделица, оформленная в виде «однорукого бандита» и предлагавшая дёрганьем за виртуальную рукоятку сделать прогноз погоды по своему желанию – там частенько «выпадало» что-то наподобие тропического ливня в понедельник, снежной бури во вторник и пустынного зноя в среду. Именно такая «погодная рулетка» натолкнула на мысль попробовать создать компьютерную реализацию упомянутого выше составителя случайного узорчатого рисунка, однако в бытовой суматохе это, даже когда изредка вспоминалось, каждый раз откладывалось на неопределённый срок.

В конце прошлого года было решено всё-таки задумку исполнить, но прежде требовалось отыскать ту самую публикацию, дабы спокойно сослаться на неё как на первоисточник. Найти удалось далеко не сразу, ведь выудить в старых подшивках журнала нужное, имея лишь одно очень смутное воспоминание о том, какие картинки эту статейку иллюстрировали – та ещё задачка. Но нет худа без добра – по ходу поисков мне попались не менее интересные материалы про приёмы скорописи и о полимерной глине*, а также ряд публикаций на тему игры (клеточного автомата) «Жизнь»**.

Итак, на странице 91 мартовского выпуска журнала «Наука и жизнь» за 1973 г. в рубрике «Развлечения не без пользы» была опубликована заметка «Орнамент “Каракули”», автором которой указан В. Ракитин из г. Алма-Ата. Там приведены рекомендации об изготовлении самих шестиугольников, описаны подходы по работе с ними и упомянуты возможные области применения (прежде всего – в декоративных целях для, например, вышивания и разрисовки тканей). Ниже на иллюстрациях из заметки показаны варианты типовых рисунков на шестиугольниках (паттернов) и пример получающегося из них узора:



* См. заметку «Мини-обзор более ранних источников» в разделе «Соображалки» сайта (URL: <http://shurichimik.narod.ru/consideration/36early-source/36-early-source.htm>).

** См. заметку «Игра “Жизнь”» в текущем разделе сайта (URL: <http://shurichimik.narod.ru/comprecreative/e-tables-data/11-life-20220322.pdf>).

Ракитин В. не забыл также упомянуть, что для составления узоров годятся не только шестиугольники, но и квадраты, ромбы и треугольники.

Ниже предлагается описание создания нескольких типов генераторов узоров на базе табличных процессоров, заодно будут рассмотрены некоторые особенности этих приложений, про которые я в своих «электронно-табличных» публикациях ранее ещё ни разу не упоминал.

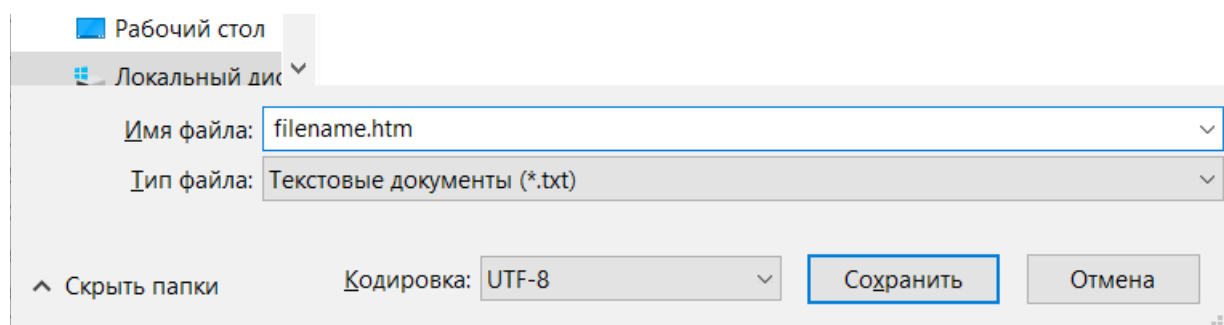
Вариант 1. Узор из квадратных паттернов с визуализацией на веб-странице

Реализация данного варианта пришла в голову первой как наиболее простая и заключалась в формировании в среде электронных таблиц набора тегов (команд языка HTML) для вставки в веб-страницу изображений.

Для тех, кто незнаком с языком гипертекстовой разметки HTML, проведу очень краткий ликбез. Этот искусственный язык используется для оформления веб-страниц (по-другому – html-файлов), которые по сути представляют собой текстовые файлы, содержащие отображающийся на странице сам текст и команды по его оформлению – теги (отделяются от основного содержимого при помощи символов "<" и ">"). Есть специальные программы – HTML-редакторы, использующиеся для разработки интернет-сайтов, ведь каждый отдельно взятый сайт в первом приближении можно считать хранящимся на каком-либо сервере набором связанных гиперссылками веб-страниц и сопутствующих вспомогательных файлов (например, файлов изображений). HTML является весьма простым языком, более того – для редактирования веб-страниц сгодится даже стандартное windows-приложение «Блокнот». Чтобы создать простейший html-файл при помощи Блокнота, запустите его и введите в окне текст:

```
<html>
<head><title>Hi!</title></head>
<body>Hello, world!</body>
</html>
```

После этого выберите в программном меню команду «Файл» → «Сохранить как...» и в открывшемся стандартном диалоговом окне укажите папку, где хотите расположить создаваемую веб-страницу. Самое главное при этом – явно указать не только собственно имя файла, но и его расширение – “htm” (можно “html”):



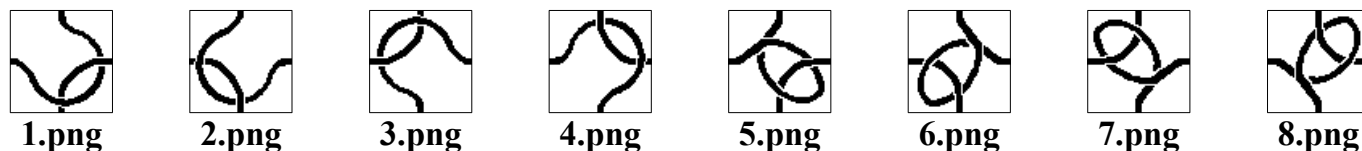
Если такой файл затем открыть при помощи любого браузера, то он считает его содержимое и отобразит в своём окне в соответствии с инструкциями (тегами) – в строке заголовка выведет “Hi!”, а в области страницы покажет фразу “Hello, world!”. Браузер позволяет просматривать веб-страницу, для редактирования её можно открыть при помощи того же «Блокнота».

Считаю необходимым отметить, что изучение основ HTML сейчас включено в школьный курс по информатике, поэтому прежде, чем переходить к дальнейшему описанию создания генератора узоров, могу заинтересовавшимся посоветовать издание, где этот материал изложен более подробно, нежели у меня здесь: Угринович Н.Д. Информатика и информационные технологии. Учебник для 10-11 классов. М.: БИНОМ. Лаборатория знаний, 2003. – 512 с. (Глава 13 «Основы языка гипертекстовой разметки документов», с. 467).

При помощи графического редактора GIMP я сначала создал две такие картинки в формате PNG размерами 64×64 пикселей:



Предполагалось формировать из них на веб-странице блок 10×10, причём каждый паттерн должен был не только выбираться случайно, но ещё и случайно ориентироваться одним из четырёх возможных способов. Таким образом требовалось восемь файлов, именам которых были присвоены числа от 1 до 8:



За основу я взял ранее созданную книгу (файл электронных таблиц), способную генерировать массив случайных чисел*, и прежде всего подправил (уменьшил) в ней размеры этого массива для данной ситуации, указав также требуемые настройки «генератора»:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	Интервал				1	2	3	4	5	6	7	8	9	10
2	Начало:	1	1	8	3	1	7	6	3	3	7	3	3	
3	Конец:	8	2	4	6	4	4	4	6	5	3	2	8	
4			3	1	3	3	7	4	3	2	3	2	4	
5	Округление:	0	4	6	5	3	4	4	8	2	3	5	3	
6			5	4	5	5	6	2	4	7	2	1	7	
7			6	8	8	7	5	7	7	5	5	7	2	
8			7	2	5	1	2	3	5	4	5	6	5	
9			8	6	1	4	2	6	5	7	2	5	6	
10			9	2	6	4	6	3	7	5	5	4	7	
11			10	4	6	4	8	5	3	2	1	8	7	
12														

Далее предстояло превратить полученный набор произвольных чисел в совокупность html-тегов – использование подобного приёма упоминалось ранее в качестве нестандартного применения программы для транслитерации текста**. В ячейку “P2” была введена такая формула:

=

Её откопировали маркером заполнения вправо, до “Y2” включительно, а в “AA2” была размещена формула, выполняющая объединение текста в ячейках диапазона “P2:Y2”:

=P2&Q2&R2&S2&T2&U2&V2&W2&X2&Y2&"
"

После этого осталось только размножить все формулы вниз, вплоть до 11-й строки, чтобы в диапазоне “AA2:AA11” получался готовый html-код, пригодный для копирования в буфер обмена:

* См. заметку «Генерация массива модельных данных» в текущем разделе сайта (URL: http://shurichimik.narod.ru/compcreative/e-tables-data/06-random_20201230.pdf).

** См. заметку «Транслитерация текста» в текущем разделе сайта (URL: http://shurichimik.narod.ru/compcreative/e-tables-data/08-transliteration_20210128.pdf).

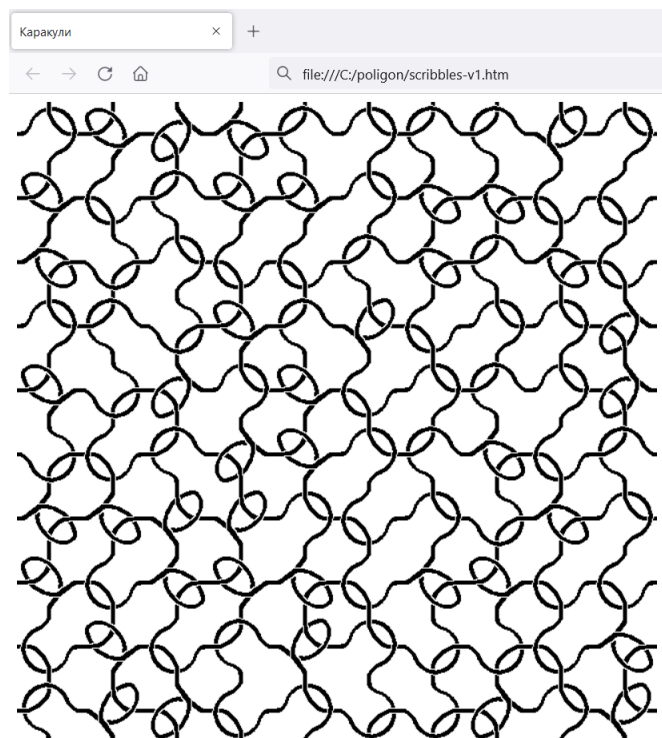
AA	AB	AC	AD
			
			
			
			
			
			
			
			
			
			

Итак, сам генератор кода готов, теперь же осталось создать веб-страницу, в который его можно будет вставить. Для этого в той же папке, где лежат PNG-картинки с паттернами, нужно разместить html-файл, содержащий такие инструкции:

```
<html>
<head>
<title>Каракули</title>
</head>
<body>
<!-- после этой строки вставить сгенерированный код -->

<!-- -->
</body>
</html>
```

После переноса html-команд из электронной таблицы в веб-страницу, последнюю нужно сохранить и открыть любым браузером (или обновить страницу, если она уже им открыта). Результат может выглядеть так:



Чуть позже я добавил ещё вот такие рисунки:



9.png



10.png



11.png



12.png



13.png



14.png



15.png

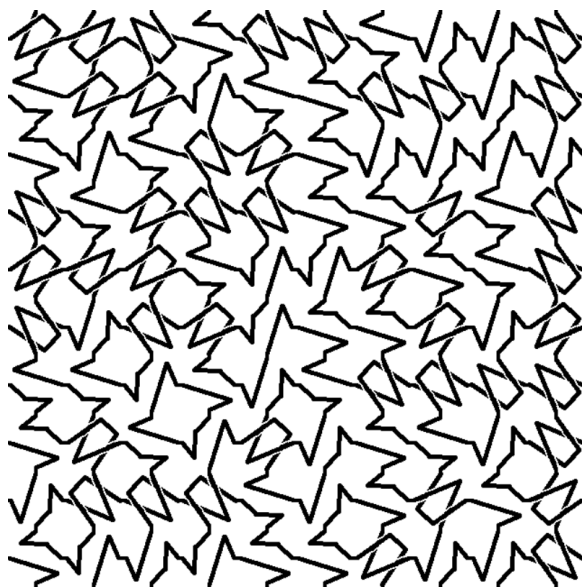


16.png

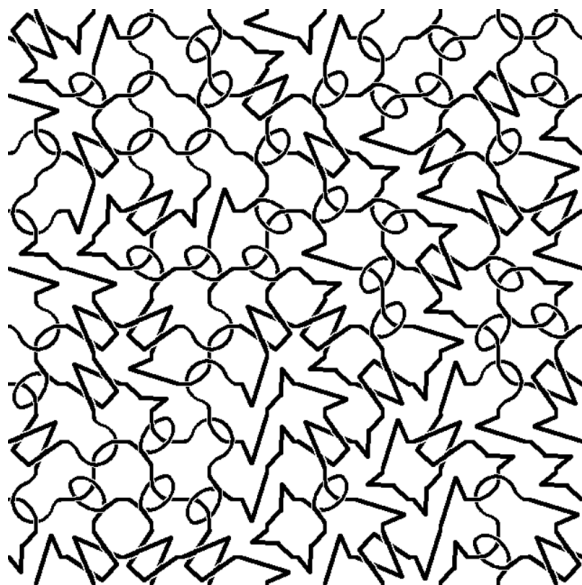
Изменяя значения начала и конца интервала, в котором производится генерирование случайных чисел, можно управлять наборами используемых для создания узора паттернов. Как нетрудно догадаться, при таких параметрах:

	A	B
1	Интервал	
2	Начало:	9
3	Конец:	16
4		
5	Округление:	0
6		

будут создаваться узоры из второго комплекта изображений, подобные показанному ниже:

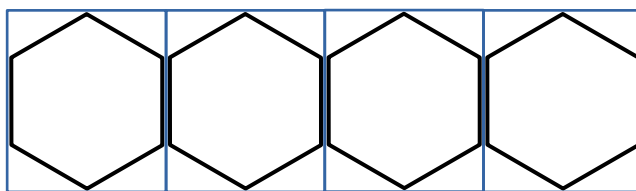


При желании можно задействовать весь имеющийся диапазон файлов – для этого достаточно указать значения начала и конца интервала 1 и 16 соответственно:

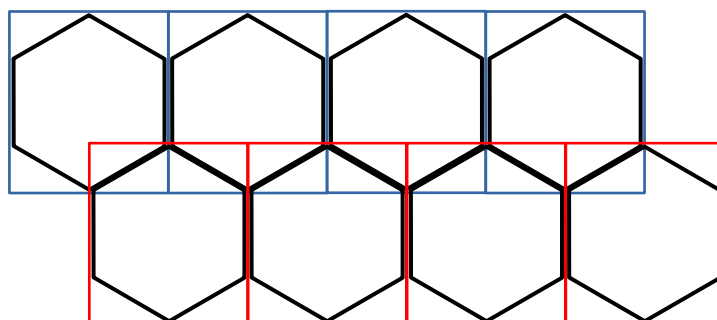


Вариант 2. Узор из шестиугольных паттернов с визуализацией на веб-странице

Цифровые изображения характеризуются шириной и высотой, поэтому сложить из одинаковых по размерам картинок мозаичный блок, вплотную состыковав рисунки друг с другом на веб-странице, не составляет большого труда. В случае шестиугольных паттернов, таких, как в обсуждавшейся выше заметке из «Науки и жизни», так сделать уже не получится – в самом файле всё равно будет по сути прямоугольное изображение и плотно соединить его с другими посредством стандартных средств HTML можно лишь в виде одного ряда:

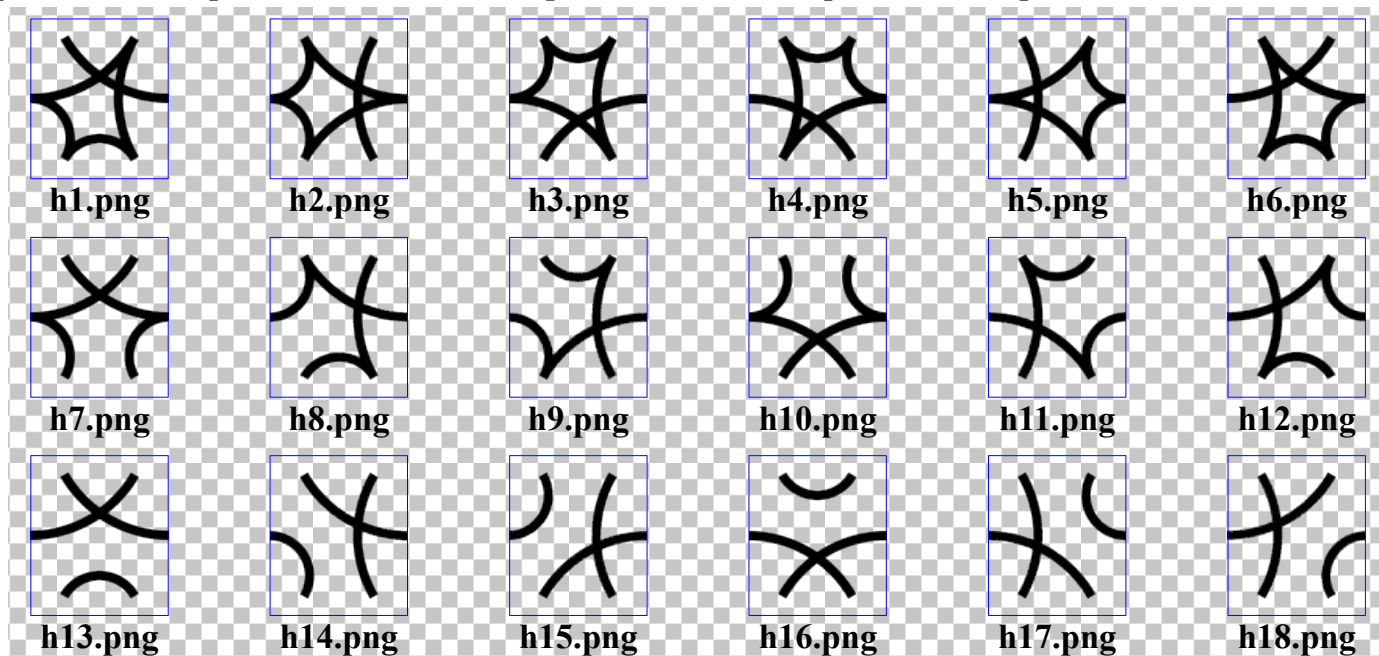


Чтобы уложить следующий ряд шестиугольников встык с первым, его нужно не только сместить вбок, но ещё и вверх, частично наложив рисунки друг на друга:

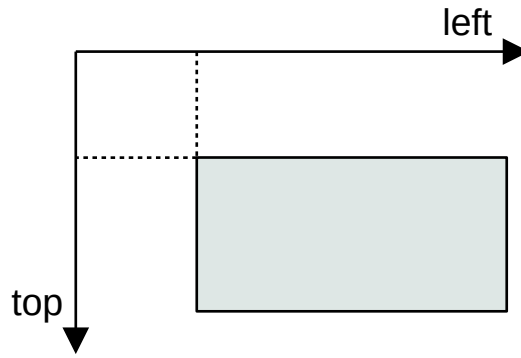


Отсюда дополнительно следует, что сами паттерны обязательно должны иметь участки с прозрачностью, чтобы при наложении одних рисунков поверх других не происходило закрытие изображений.

Для генератора «каракулей» Ракитина В. я создал 18 PNG-файлов (86×100 пикселей) с тремя типами предложенных им паттернов – по шесть вариантов поворота для каждого:



Как уже было сказано, стандартными тегами HTML обойтись не получится, поэтому для размещения рисунков в нужном порядке на веб-странице я воспользовался CSS – если кратко, то это средство, позволяющее задавать частям веб-страницы дополнительные параметры оформления. В окне браузера координаты размещаемого на странице объекта (текст, рисунок и т. п.) отсчитываются от верхнего левого угла:



В связи с этим задумка была следующая. Сначала нужно поместить ряд рисунков внутри контейнера <div>:

```
<div><img src=pic1.png><img src=pic2.png>...</div>
```

Затем полученный блок можно позиционировать на веб-странице как надо, указав у <div> параметр “style” с необходимыми инструкциями CSS:

```
<div style=' position: absolute; top: 42px; left: 24px; '>
```

Для создания требующегося в данном случае html-кода в “AA2” (см. Вариант 1) была размещена модифицированная формула, которую затем размножили вниз до 11-й строки включительно:

```
= "<div style=' position: absolute; top: "&1+(D2-1)*73&"px; left: "&1+21,5*(1+(-1)^D2)&"px; '>"&P2&Q2&R2&S2&T2&U2&V2&W2&X2&Y2&"</div>"
```

Обратите внимание: красным шрифтом выделена её часть, которая ориентируется на стоящие в столбце “D” числа:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	Интервал			1	2	3	4	5	6	7	8	9	10	
2	Начало:	1	1	8	3	1	7	6	3	3	7	3	3	
3	Конец:	8	2	4	6	4	4	4	6	5	3	2	8	
4			3	1	3	3	7	4	3	2	3	2	4	
5	Округление:	0	4	5	5	3	4	4	8	2	3	5	3	
6			5	4	5	5	6	2	4	7	2	1	7	
7			6	3	8	7	5	7	7	5	5	7	2	
8			7	2	5	1	2	3	5	4	5	6	5	
9			8	6	1	4	2	6	5	7	2	5	6	
10			9	2	6	4	6	3	7	5	5	4	7	
11			10	4	6	4	8	5	3	2	1	8	7	
12														

Указанный фрагмент отвечает за вычисление значения координаты y (CSS-инструкция “top”) в зависимости от целочисленного параметра n и задаёт смещение от верхней части окна браузера для блока паттернов с шагом 73 пикселя:

$$y = 1 + 73 \cdot (n - 1)$$

Аналогично, выделенная зелёным цветом часть формулы задаёт попеременный отступ x от левого края окна браузера (CSS-инструкция “left”) в 1 или в 44 пикселя также на основании значения n:

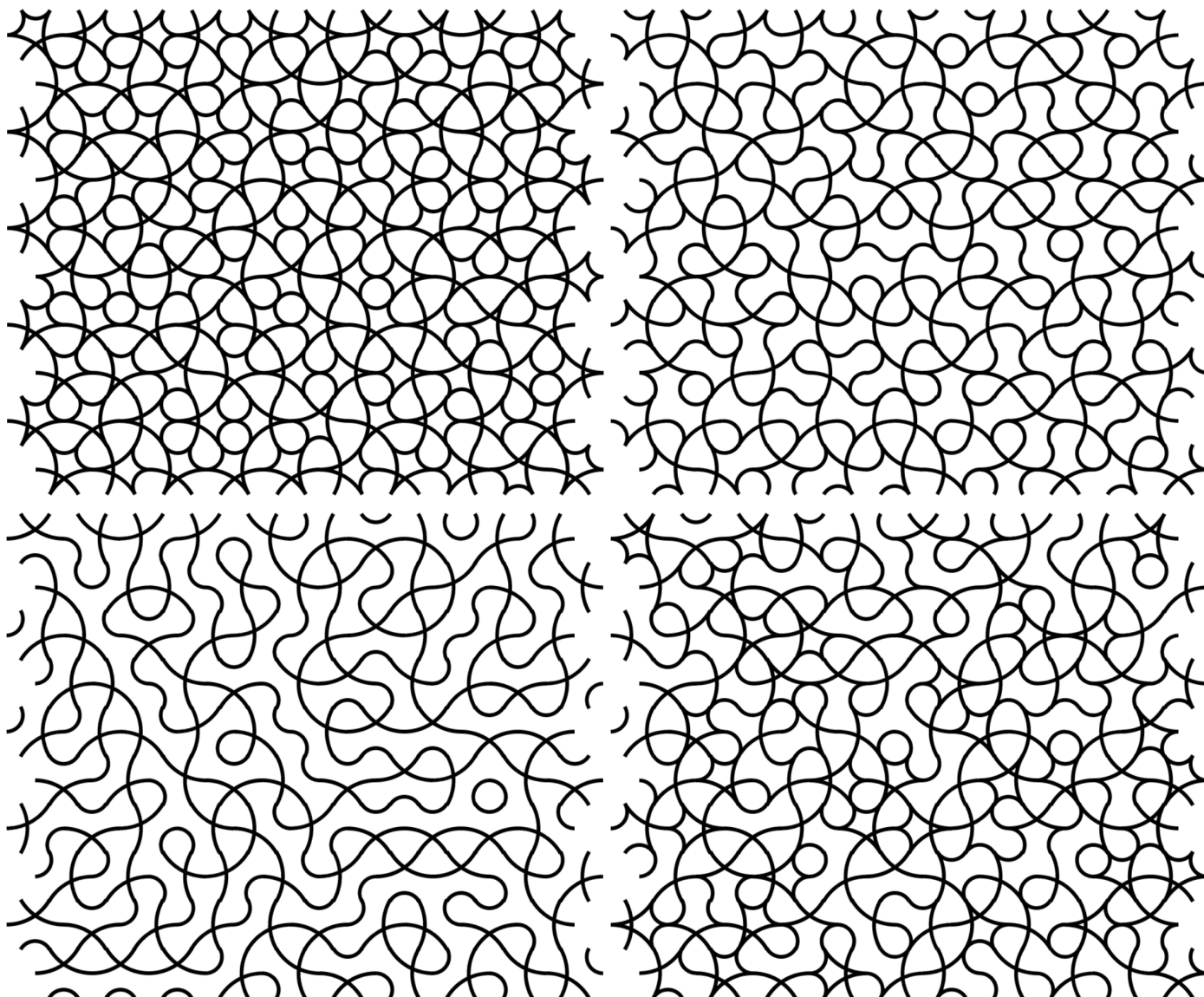
$$x = 1 + 21,5 \cdot (1 + (-1)^n)$$

Как легко догадаться, этим обеспечивается разница смещения вбок на $44 - 1 = 43$ пикселя между соседними контейнерами <div>, что как раз равно половине ширины изображения в файле каждого паттерна (86 пикселей).

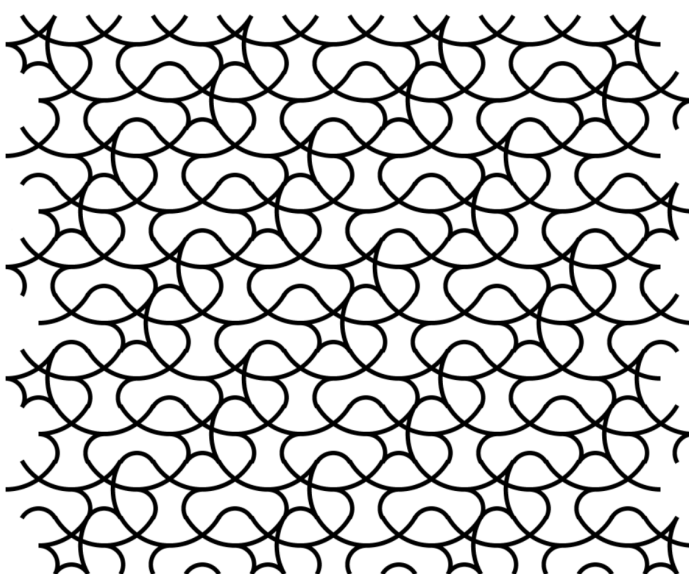
Формулы, создающие тег для вставки изображения (диапазон “P2:Y11”), чуток отличаются от указанных в Варианте 1, так как имена файлов с шестиугольными паттернами содержат букву “h” (показано на примере “P2”):

= ""

Получающийся в итоге код в ячейках диапазона "AA2:AA11" можно добавлять в html-файл. Ниже приведены примеры четырёх узоров, составленных из первого (вверху слева), второго (вверху справа), третьего (внизу слева) паттернов и из всех трёх сразу (внизу справа):



Строго говоря, в заметке Ракитина В. прежде всего велась речь о наличии какой-либо системы в чередовании типов паттернов и их поворотов, благодаря чему можно создавать именно орнаменты, а не хаотично составленные рисунки. Добиться регулярности несложно – справа для примера показан узор, который получается, если в диапазоне ячеек "E2:N11" вручную проставить как по горизонтали, так и по вертикали циклично повторяющиеся числа 1 – 7 – 13 вместо имеющихся там формул.



Вариант 3. Узор из квадратных паттернов с визуализацией средствами табличного процессора

Предложенные выше Вариант 1 и Вариант 2 несмотря на свою работоспособность обладают одним недостатком: чтобы создать и увидеть очередной узор, необходимо сначала сгенерировать новый массив случайных чисел (например нажатием клавиши [Delete] при выделенной какой-нибудь заведомо пустой ячейке), затем выделить и скопировать из ячеек таблицы в буфер обмена соответствующий html-код, после этого – переключиться на открытый в «Блокноте» html-файл и вставить из буфера информацию в него, не забыв сохранить изменения, а в завершение – переключиться на открытый в браузере этот же файл и обновить страницу.

Процесс визуализации каждого нового узора можно сделать быстрым, причём прямо в среде табличного процессора, хотя для этого потребуются более хлопотная подготовка. Предлагаемый способ заключается в построении диаграммы, где вместо стандартных маркеров точек графика (кружки, квадратики, треугольники и т. п.) будут использоваться изображения, загруженные из внешнего файла (подходят популярные графические форматы, в том числе – PNG). Но сначала, чтобы программа могла нормально воспринять и корректно отобразить сами точки, данные из “E2:N11” (см. Вариант 1) нужно рассортировать. Так как в указанном диапазоне ячеек каждому числу соответствует какой-либо определённый паттерн (тип рисунка и вариант его поворота), попробуем представить для начала все случаи выпадения числа 1 в “E2:N11” в виде точек с координатами, соответствующими номеру столбца (координата x) и строки (координата y) в этом рассматриваемом блоке (эти значения для наглядности проставлены в “E1:N1” и “D2:D11”). В столбцах “P” и “Q” нужно перебрать все возможные комбинации номеров столбцов и строк в блоке ячеек “E2:N11”:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
1	Интервал				1	2	3	4	5	6	7	8	9	10			Паттерн:		1	
2	Начало:	1		1	3	5	3	4	4	8	2	2	3	2		Столбец	Строка	x	y	
3	Конец:	8		2	6	6	5	4	3	8	3	8	5	4		1		1		
4				3	3	2	7	2	5	1	8	3	2	3		1		2		
5	Округление:	0		4	4	8	3	2	5	8	6	3	4	6		1		3		
6				5	4	1	4	2	2	6	7	2	4	4		1		4		
7				6	3	2	5	1	6	1	4	6	8	3		1		5		
8				7	7	6	5	4	2	7	7	6	4	6		1		6		
9				8	5	5	8	1	6	3	4	2	1	1		1		7		
10				9	7	6	2	4	3	4	5	4	6	7		1		8		
11				10	2	8	7	6	7	6	5	6	4	7		1		9		
12																1		10		
13																2		1		
14																2		2		
15																2		3		
16																2		4		
17																2		5		
18																2		6		
19																2		7		
20																2		8		
21																2		9		
22																2		10		
23																3		1		
24																3		2		
25																3		3		

Те их сочетания, при которых в “E2:N11” будет находиться ячейка с числом 1, нужно будет отобразить в “R” и “S”. Для этого поместим в “R3” такую формулу:

$$=ЕСЛИ(ИНДЕКС(E2:N11;$Q3;$P3)=R$1;$P3;-3)$$

$$=IF(INDEX(E2:N11;$Q3;$P3)=R$1;$P3;-3)$$

В ней использована функция

$$ИНДЕКС(; ;)$$

INDEX (; ;)

Результатом её работы является ссылка на определённую ячейку листа электронной таблицы, которая выбирается с учётом значений аргументов. Первый из них – это диапазон (блок) ячеек, внутри которого ищется нужная. Вторым и третьим аргументы – номера, соответственно, строки и столбца, отсчитываемые от верхнего левого угла блока. У рассматриваемой функции есть ещё четвёртый необязательный аргумент, который в данной ситуации не нужен, так как в качестве первого аргумента указан только один блок ячеек.

Таким образом, формула в “R3” проверяет, стоит ли в ячейке “E2” число 1 и если да, то возвращает значение координаты x , то есть соответствующий номер столбца внутри блока (диапазона) “E2:N11”, а в противном случае формула возвращает число -3 , чуть позже станет понятно, почему нужно именно оно.

В ячейке “S3” понадобится указать такую формулу:

=ЕСЛИ (R3=-3; 3; -\$Q3)

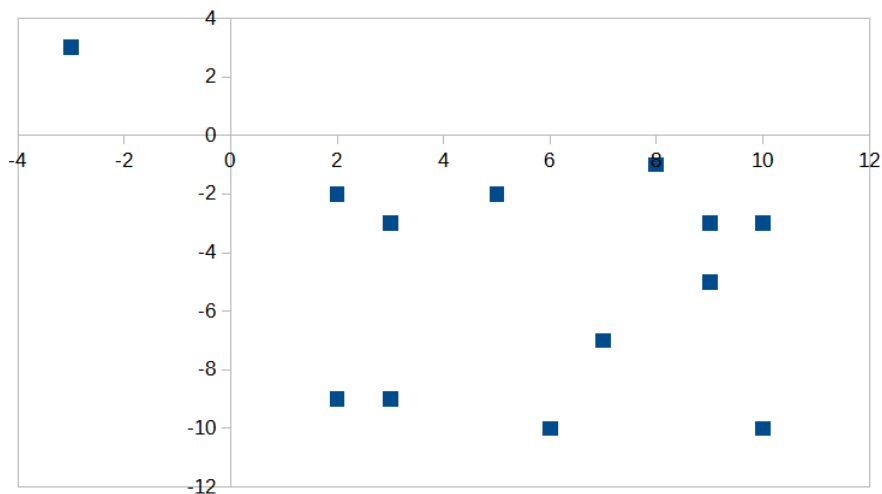
=IF (R3=-3; 3; -\$Q3)

Как легко видеть, она возвращает число 3, если в “R3” оказалось -3 , в остальных случаях возвращается номер строки (координату y), взятый с обратным знаком. Для чего это надо тоже станет ясно позже, а пока следующим действием должно быть размножение формул в “R3” и “S3” маркером заполнения вниз вплоть до 102-й строки:

	O	P	Q	R	S	T
90		9	8	-3	3	
91		9	9	-3	3	
92		9	10	-3	3	
93		10	1	-3	3	
94		10	2	-3	3	
95		10	3	10	-3	
96		10	4	-3	3	
97		10	5	-3	3	
98		10	6	-3	3	
99		10	7	-3	3	
100		10	8	-3	3	
101		10	9	-3	3	
102		10	10	10	-10	
103						

На основе полученного ряда пар чисел (координат) в “R3:S102” можно построить диаграмму в виде графика с изображением только точек (маркеров) без соединения их линиями (*Пособие*, с. 58). Выглядеть это будет примерно так:



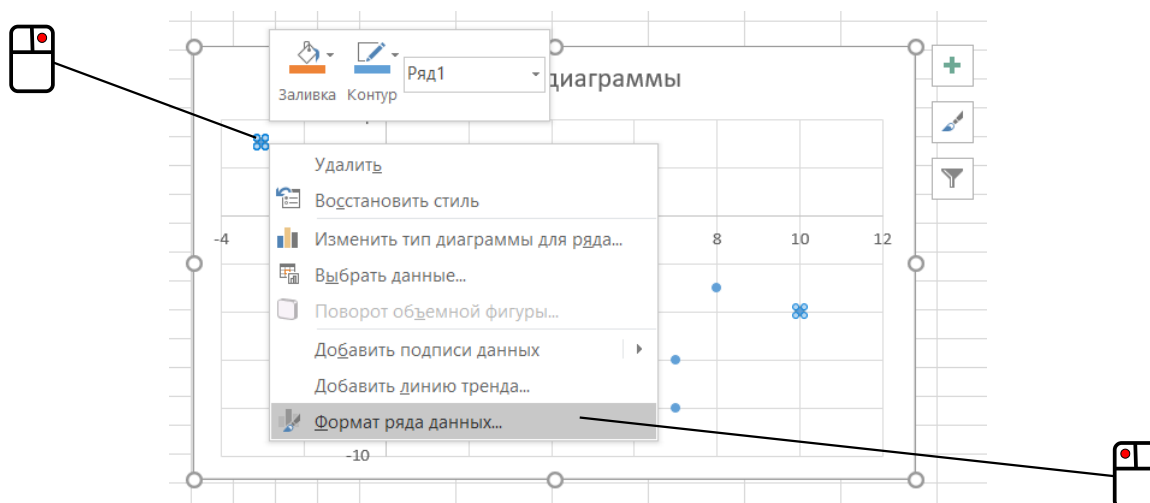


В процессе построения графика в окне «Мастера диаграмм» были сняты флажки «Подписи в первой строке», «Подписи в первом столбце», «Ось X» и «Ось Y» («Отобразить сетки»), а также «Показать легенду».

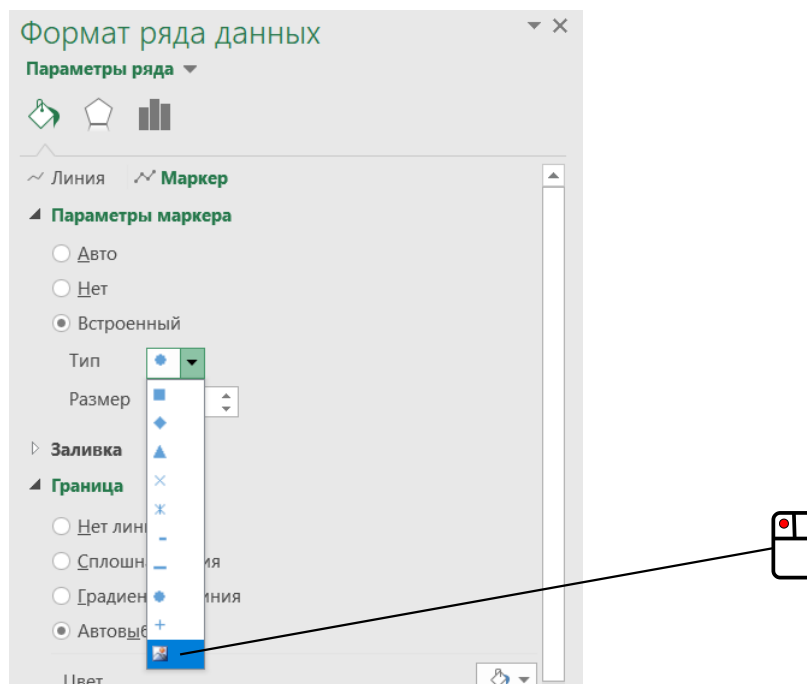
Легко заметить, что маркеры точек на графике располагаются в четвёртом квадранте соответственно тому, как расположены числа 1 в блоке “E2:N11”, а ячейки с другими числами отображаются как одна и та же точка с координатами $(-3; 3)$, то есть во втором квадранте – вот для чего были необходимы составленные именно таким образом формулы в столбцах “R” и “S”. Дело в том, что когда график на диаграмме строится по диапазону, в котором отсутствуют прогалы (пустые ячейки), программа изображает точки как надо, иначе она может повести себя неадекватно, отобразив данные в соответствии с ведомой только ей внутренней логикой.

Полученная диаграмма нуждается в дополнительной настройке и прежде всего – в изменении маркера.

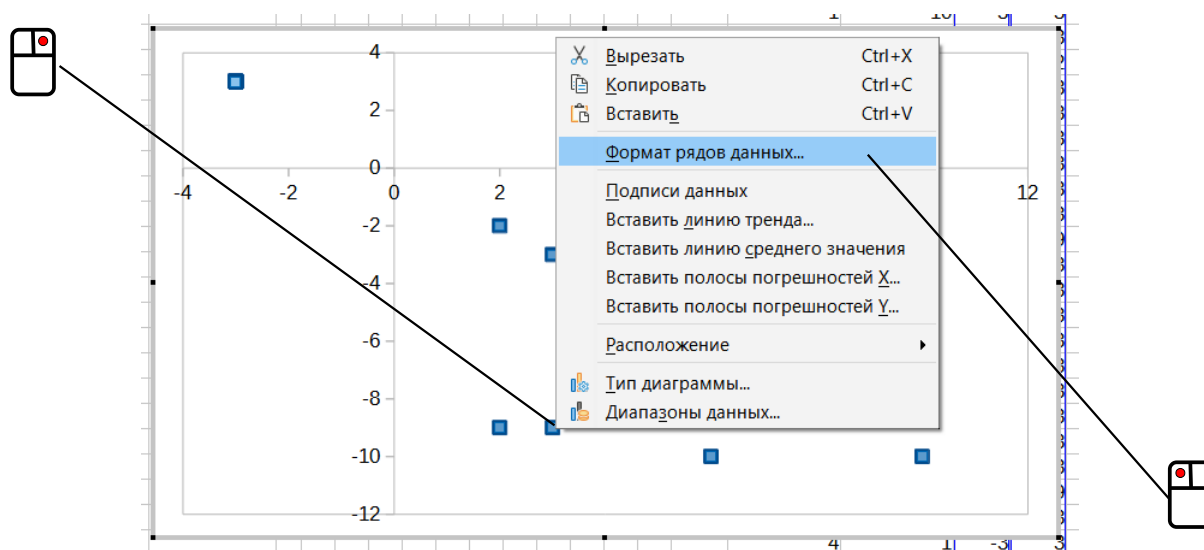
Вызовите настройки вида маркера:



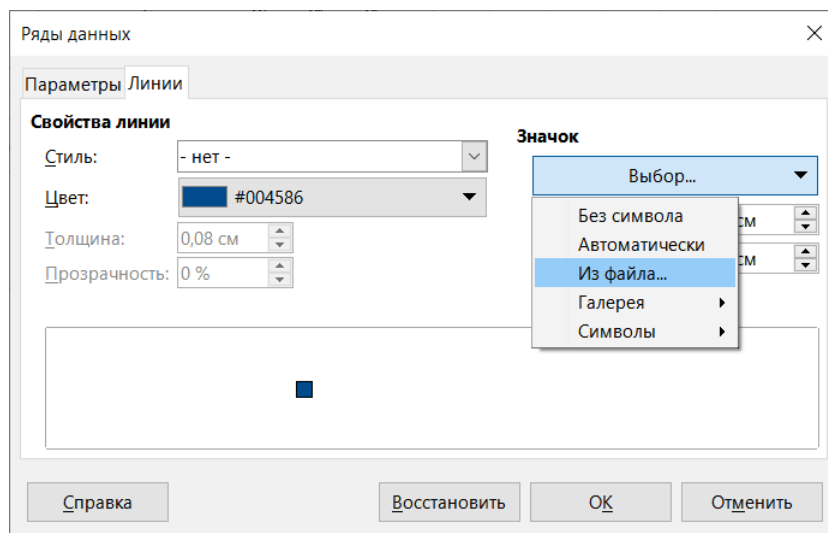
В выпадающем списке с типами маркеров выберите загрузку рисунка для него из внешнего файла:



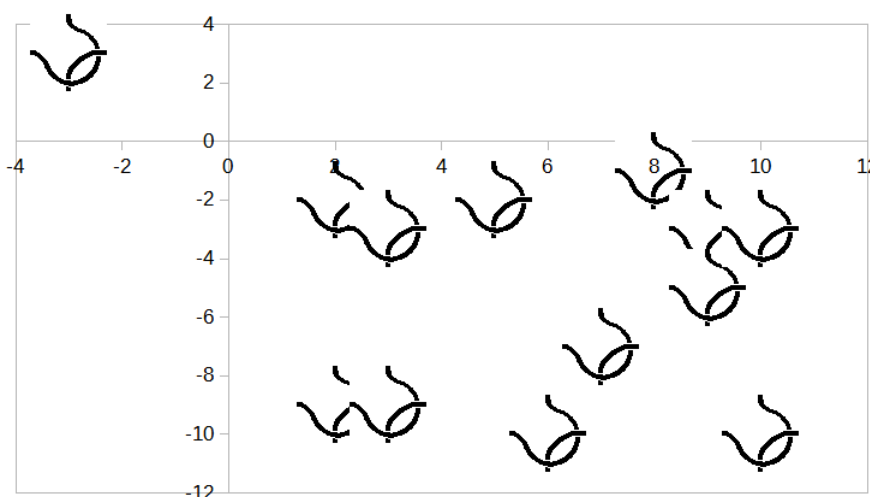
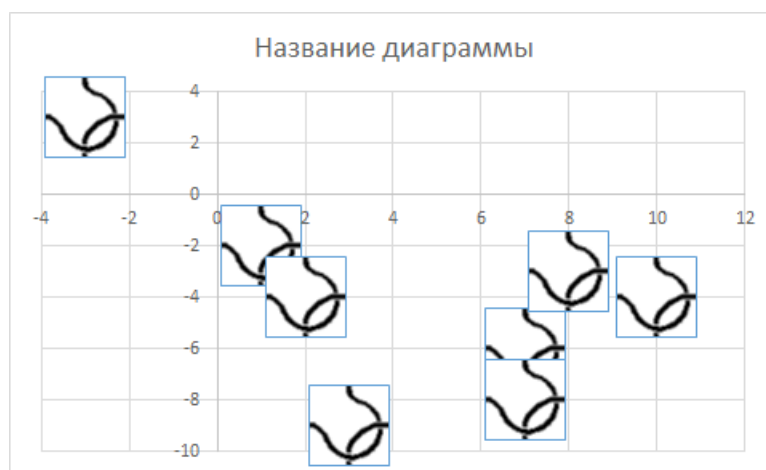
Переключитесь в режим правки диаграммы, выполнив двойной щелчок по ней и вызовите настройки вида маркера:



В появившемся диалоговом окне «Ряды данных» на вкладке «Линии» выберите в выпадающем списке «Значок» загрузку рисунка маркера из внешнего файла:



Откроется стандартное диалоговое окно, в котором нужно будет указать файл "1.png". В итоге диаграмма приобретёт следующий вид:

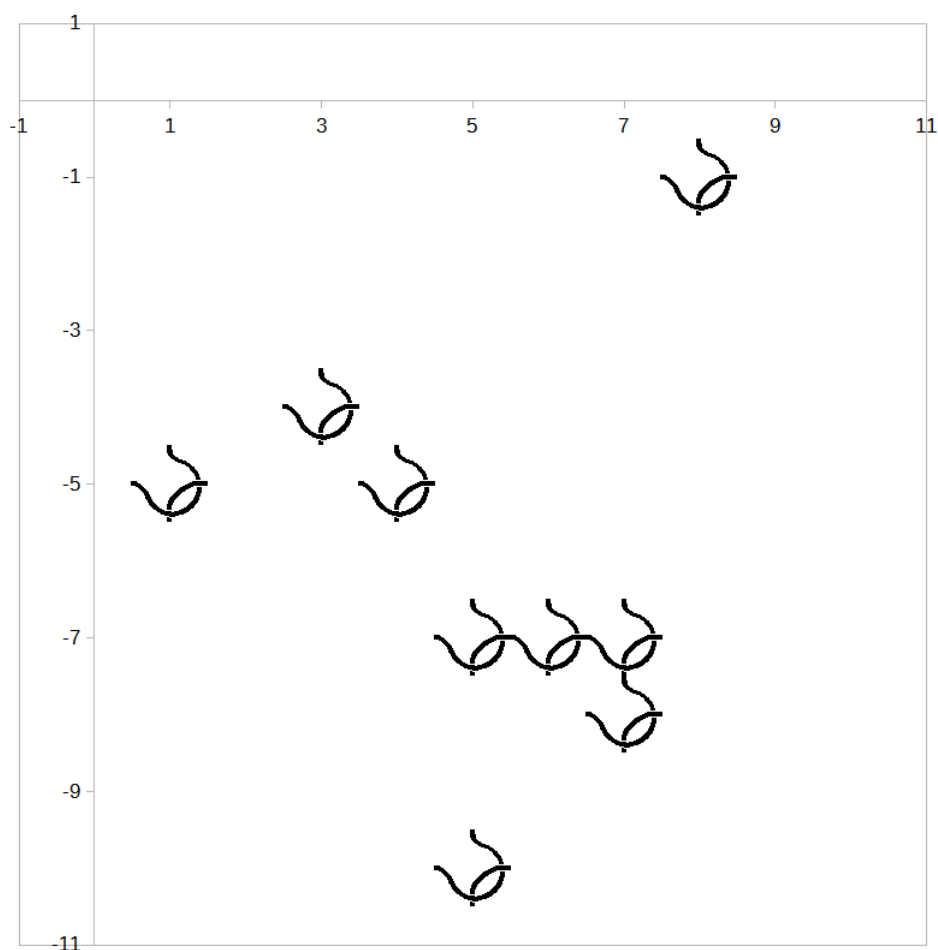
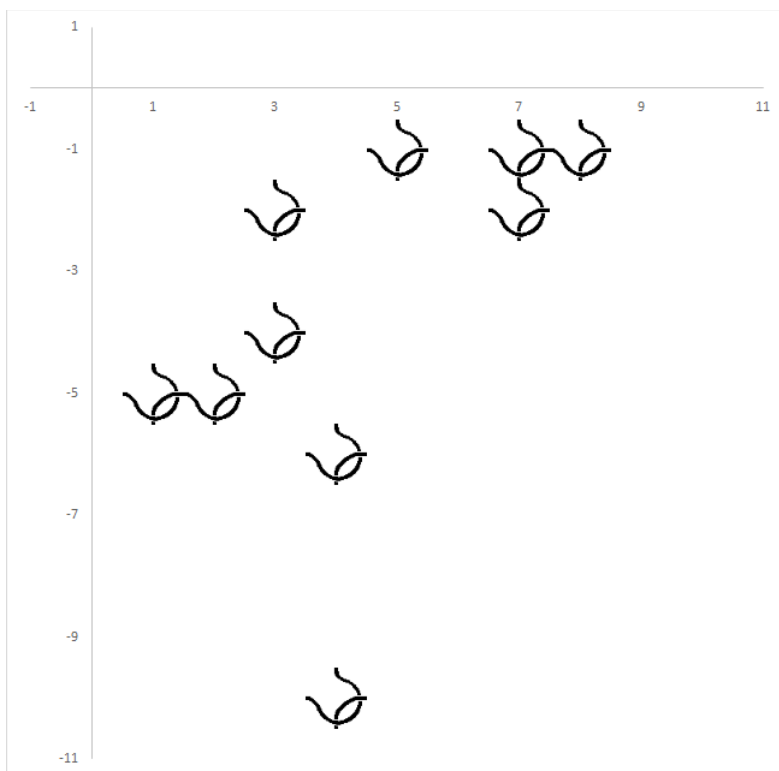


Теперь стала понятна необходимость других дополнительных настроек.

Прежде всего следует удалить элемент «Название диаграммы», линии координатной сетки, а в свойствах маркера убрать обрамление синего цвета.

В параметрах каждой оси надо явно указать границы от -1 до 11 (для горизонтальной) и от -11 до 1 (для вертикальной), благодаря чему маркер точки с координатами $(-3; 3)$ перестанет

показываться. После этого останется подправить размеры самой диаграммы, добившись, чтобы края маркеров (паттернов) при соседстве лишь касались друг друга, а не перекрывались:

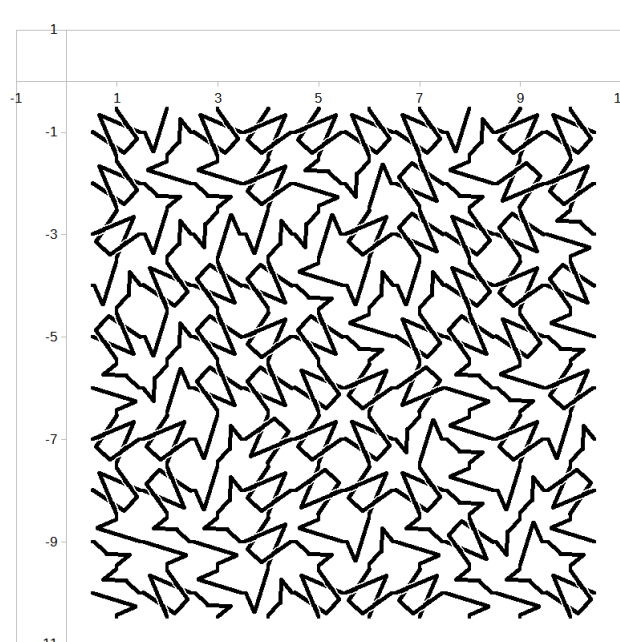
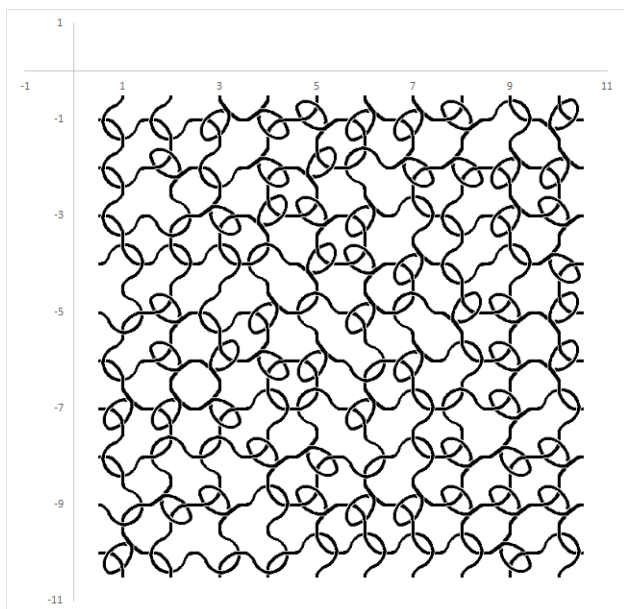


Ну что ж, самое главное сделано. Теперь можно заняться сортировкой всех выпадающих в “E2:N11” значений числа 2. Для этого достаточно последовательно скопировать через буфер

обмена формулы из “R3” и “S3” в “T3” и “U3” соответственно, а затем размножить их вниз, до 102-й строки включительно – благодаря абсолютной адресации в выражениях формул, они сами подстроятся нужным образом для правильной работы:

	L	M	N	O	P	Q	R	S	T	U	V
1	8	9	10			Паттерн:	1		2		
2	2	6	6		Столбец	Строка	x	y	x	y	
3	7	2	4		1	1	1	-1	-3	3	
4	5	1	5		1	2	-3	3	1	-2	
5	3	3	6		1	3	-3	3	-3	3	
6	1	4	4		1	4	-3	3	-3	3	
7	6	3	6		1	5	-3	3	-3	3	
8	4	4	3		1	6	-3	3	1	-6	
9	2	2	3		1	7	-3	3	1	-7	
10	7	7	2		1	8	-3	3	-3	3	
11	6	2	4		1	9	-3	3	-3	3	
12					1	10	-3	3	-3	3	
13					2	1	-3	3	-3	3	
14					2	2	-3	3	-3	3	

После можно будет на основе значений в “T3:U102” добавить на диаграмму ещё один ряд данных (*Пособие*, с. 78) и настроить его маркеры в виде изображения из файла “2.png”. Действуя далее аналогичным образом, следует добавить отображение остальных паттернов и получить в итоге готовый генератор узоров, способный создавать рисунки, подобные тем, что приведены ниже.



Вариант 4. Узор из шестиугольных паттернов с визуализацией в среде табличного процессора

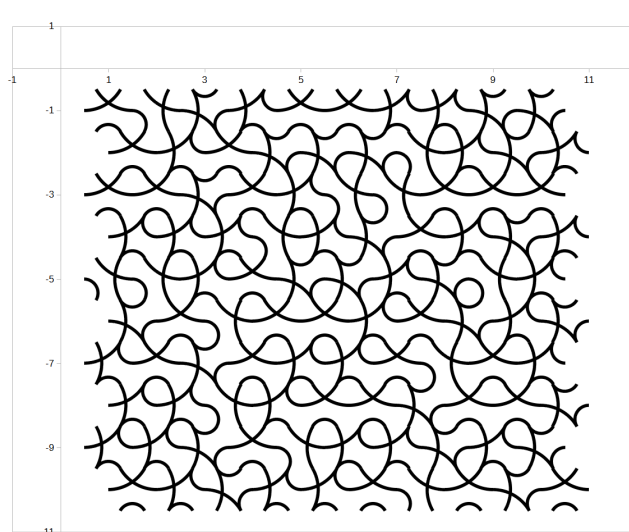
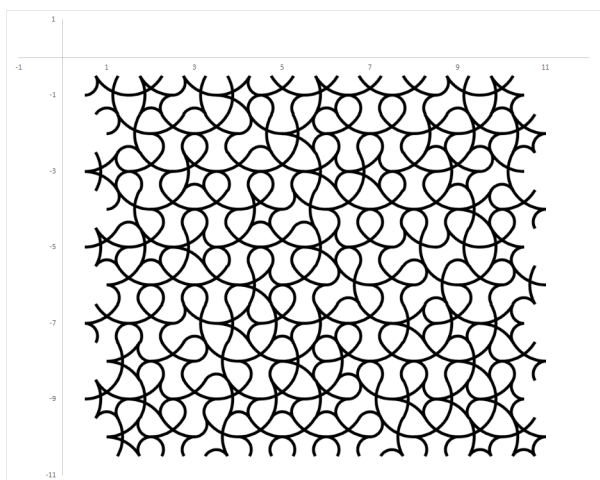
После того, как пример быстрой визуализации с квадратными паттернами разобран, вполне понятно, что генератор узоров «на шестиугольниках» делается аналогично, хотя формулы для координаты x нужно подкорректировать, ведь здесь необходимо, чтобы каждый маркер-рисунок, стоящий в чётном ряду, был смещён на половину единицы вправо. Рассмотрим в качестве примера формулу в ячейке “R3” для данного варианта:

=ЕСЛИ (ИНДЕКС (\$E\$2:\$N\$11;\$Q3;\$P3)=R\$1;
\$P3+ЕСЛИ (\$Q3/2=ЦЕЛОЕ (\$Q3/2) ;0,5;0) ; -3)

=IF (INDEX (\$E\$2:\$N\$11;\$Q3;\$P3)=R\$1;
\$P3+IF (\$Q3/2=INT (\$Q3/2) ;0,5;0) ; -3)

Фрагмент, выделенный жирным шрифтом, проверяет номер строки в блоке “E2:N11” на чётность (если чётное число разделить пополам, то частное будет целым, в случае же нечётного числа получится дробная величина, целая часть которой, разумеется, не равна ей самой) – в этом случае он каждый раз дополнительно прибавляет к координате x значение 0,5. По этой причине также желательно в параметрах горизонтальной оси диаграммы явно задать границы в интервале от -1 до 12 .

Примеры получающихся «каракулей» приведены ниже.



© Широков Александр, 26.07.2022