



Ткаченко Кирилл Станиславович

Инженер 1-й категории

*Федерального государственного автономного
образовательного учреждения высшего образования
«Севастопольского государственного университета».*

Реализация на школьном алгоритмическом языке классического метода шифрования «Решётка Кардано»

Предлагается программная реализация классического метода шифрования «Решётка Кардано». Применяется школьный алгоритмический язык, приводятся полный исходный текст и результаты работы, что может быть полезно всем изучающим программирование.

Изучение основ программирования может быть хорошо закреплено при решении простых, но интересных примеров, имеющих отношение к практике. Таким примером является метод шифрования «Решётка Кардано». Этот метод оказал влияние на развитие систем шифрования [1–3] и иногда приводится как задача на программирование нетривиальной сложности [4]. Целью настоящей работы является программирование метода шифрования «Решетка Кардано» на школьном алгоритмическом языке.

Краткое описание метода шифрования «Решётка Кардано»

Имеется два квадрата. На одном из них будет записан зашифрованный текст, другой будет являться своеобразным трафаретом – шифровальной решеткой. В шифровальной решетке вырезаны окошки. Для шифрования решетка прикладывается к квадрату с текстом, затем в окошки трафарета вписываются последовательно фрагменты исходного текста. После этого решетка поворачивается по часовой стрелке на 90 градусов и последовательность действий по шифрованию повторяется еще три раза. Расшифровывание выполняется аналогично, но вместо записывания фрагментов текста в окошки, эти фрагменты последовательно из окошек извлекаются. Предполагается, что при шифровании и расшифровывании проход окошек шифровальной решетки происходит слева направо построчно, сверху вниз. Если количество окошек шифровальной решетки недостаточно для полного заполнения результирующего квадрата при совершении всех поворотов, то свободное пространство может быть заполнено, например, запутывающим текстом.

Программирование метода шифрования «Решётка Кардано»

На рисунке 1 знак «*» означает, что в указанном месте на данном повороте решетки будет записан очередной символ, знак «.» – это место на данном повороте не используется.

I	II	III	IV
. * . .	. . * .	* . . .	. . . *
* . . .	. . . *	. * . .	. . * .
. . * .	* . . .	. . . *	* . . .
. . . *	* . . .	. . * .	. * . .

Рис. 1

Пример работы с осмысленным текстом приводится на рисунке 2. В качестве текста, который надо зашифровать, возьмем строку «КИРИЛЛСТАНИСЛАВО».

Для упрощения демонстрации метода программа поддерживает работу только с полностью заполняющим все повороты решётки текстом.

I	II	III	IV
. К . .	. . Л .	А . . .	. . . Л
И . . .	. . . Л	. Н . .	. . А .
. . Р .	. С . .	. . . И	В . . .
. . . И	Т . . .	. . С .	. О . .

Рис. 2

Зашифрованный текст примет вид, как на рисунке 3.

А	К	Л	Л
И	Н	А	Л
В	С	Р	И
Т	О	С	И

Рис. 3

Стороны решётки в данной работе и число поворотов для произвольных случаев есть $N = 4$, шифровальная решётка располагается в двумерном массиве $N \times N$:

```
цел N = 4;  
сим таб Решетка[1 : N, 1 : N];
```

Основной алгоритм Потенциал осуществляет взаимосвязь всех остальных алгоритмов. В строковых переменных СтрокаИсходная, СтрокаЗашифрованная, СтрокаРасшифрованная находятся исходная, зашифрованная и расшифрованная строки:

алг Потенциал
нач

лит СтрокаИсходная, СтрокаЗашифрованная, СтрокаРасшифрованная;

Полагается, что символы решётки в исходном строковом представлении располагаются построчно снизу вверх и слева направо. Исходная решетка задается строкой `".*.*.....*...."`, после преобразования во внутреннее представление решетка выводится:

```
ИсходнаяРешетка (".*.*.....*.....*");
ВывестиРешетку;
```

Исходная строка для тестирования есть "АБВГДЕЁЖЗИЙКЛМНО", строка шифруется, результат шифрования расшифровывается:

```
СтрокаИсходная := "АБВГДЕЁЖЗИЙКЛМНО";
СтрокаЗашифрованная := Зашифровать(СтрокаИсходная);
СтрокаРасшифрованная := Расшифровать(СтрокаЗашифрованная);
```

Исходная, зашифрованная и расшифрованная строки выводятся:

```
вывод СтрокаИсходная, нс;  
вывод СтрокаЗашифрованная, нс;  
вывод СтрокаРасшифрованная, нс;
```

 KOH

Алгоритм ИсходнаяРешетка(лит РешеткаТекст) переводит решетку из строкового представления во внутреннее. Переменные i и j – индексы и счетчики:

алг ИсходнаяРешетка (лит РешеткаТекст)

Науч

цел i, j ;

Все символы в решётке заполняются таким образом, чтобы символ в решетке на i -й строке и j -м столбце был взят из строкового представления по индексу, равному $(i - 1) * N + j$ (решетка в исходном строковом представлении располагается построчно):

```

нц для i от 1 до N
  нц для j от 1 до N
    Решетка[i, j] := РешеткаТекст[(i - 1) * N + j];
  
```

КЦ

КЦ

KOH

Алгоритм ПовернутьРешетку производит поворот по часовой стрелке решетки на 90 градусов. Двумерный массив Рез NxN служит для временного хранения результата, переменные i и j – индексы и счетчики:

```
алг ПовернутьРешетку  
нач
```

```
    сим таб Рез[1 : N, 1 : N];  
    цел  $i, j$ ;
```

По всем строкам и столбцам решетки, символ помещается в место после совершения поворота по часовой стрелке на 90 градусов:

```
    нц для  $i$  от 1 до N  
        нц для  $j$  от 1 до N  
            Рез[ $j, N + 1 - i$ ] := Решетка[ $i, j$ ];  
        кц  
    кц
```

Результат поворота записывается на место исходной решетки:

```
    нц для  $i$  от 1 до N  
        нц для  $j$  от 1 до N  
            Решетка[ $i, j$ ] := Рез[ $i, j$ ];  
        кц  
    кц  
кон
```

Алгоритм ВывестиРешетку выводит решетку на экран. Переменные i и j представляют индексы и счетчики:

```
алг ВывестиРешетку  
нач
```

```
    цел  $i, j$ ;
```

Для всех строк и столбцов символ решетки выводится, каждая строка решетки размещается на отдельной строке:

```
    нц для  $i$  от 1 до N  
        нц для  $j$  от 1 до N  
            вывод Решетка[ $i, j$ ];  
        кц  
        вывод нс;  
    кц  
кон
```

Алгоритм лит Зашифровать(лит СтрокаИсходная) производит шифрование строки СтрокаИсходная. Результат шифрования помещается в СтрокаЗашифрованная, промежуточные итоги шифрования для удобства размещаются по отдельным ячейкам массива СтрокаЗашифрованнаяМассив. Текущий символ исходной строки отмечается в Счетчик, индексные переменные и счетчики – это i, j, k :

```
алг лит Зашифровать(лит СтрокаИсходная)  
нач  
    лит СтрокаЗашифрованная;
```

```

сим таб СтрокаЗашифрованнаяМассив [1 : N, 1 : N];
цел Счетчик;
цел i, j, k;

```

Зашифрованная строка вначале пустая, а Счетчик ничего не индексирует:

```

СтрокаЗашифрованная := "";
Счетчик := 0;

```

Для всех четырех поворотов решетки, по всем строкам и столбцам решетки:

```

нц для k от 1 до N
  нц для i от 1 до N
    нц для j от 1 до N

```

Если в решетке имеется на i-й строке j-м столбце прорезанное окошко, то сдвигаем отчет символов в исходной строке и помещает в это прорезанное окошко символ из исходной строки:

```

      если Решетка[i, j] = "*"
        то
          Счетчик := Счетчик + 1;
          СтрокаЗашифрованнаяМассив[i, j] := СтрокаИсход-
ная[Счетчик];
        все
      кц
    кц
  кц

```

Решётка поворачивается:

```

    ПовернутьРешетку;
  кц

```

По всем строкам и столбцам массива с результатом шифрования, символы конкатенируется в результирующую шифрованную строку сверху вниз и слева направо:

```

  нц для i от 1 до N
    нц для j от 1 до N
      СтрокаЗашифрованная := СтрокаЗашифрованная + СтрокаЗа-
шифрованнаяМассив[i, j];
    кц
  кц

```

Результат шифрования возвращается:

```

  знач := СтрокаЗашифрованная;
кон

```

Алгоритм лит Расшифровать(лит СтрокаЗашифрованная) производит рас-шифровывание строки СтрокаЗашифрованная. В переменную СтрокаРасшиф-рованная помещается результат расшифровывания, i, j, k – индексы и счетчики:

```

алг лит Расшифровать (лит СтрокаЗашифрованная)
нач
  лит СтрокаРасшифрованная;
  цел i, j, k;

```

Расшифрованная строка пуста:

```
СтрокаРасшифрованная := "";
```

Для всех четырех поворотов решетки, по всем строкам и столбцам решетки:

```
нц для k от 1 до N  
  нц для i от 1 до N  
    нц для j от 1 до N
```

Если в решётке на рассматриваемой позиции располагается прорезанное окошко, то к расшифрованной строке добавляется символ из зашифрованной строки, находящийся на позиции $(i - 1) * N + j$ (то есть при развороте зашифрованной строки построчно сверху вниз и слева направо):

```
    если Решетка[i, j] = "*"
    то
        СтрокаРасшифрованная := СтрокаРасшифрованная +
СтрокаЗашифрованная[(i - 1) * N + j];
    все
кц
кц
```

Решетка поворачивается:

```
    ПовернутьРешетку;
кц
```

Результат расшифровывания возвращается:

```
знач := СтрокаРасшифрованная;
кон
```

Полный исходный текст программы находится в приложении А, результат работы – в приложении Б и на снимке экрана.

Полученные результаты применения школьного алгоритмического языка могут быть полезны всем изучающим программирование, поскольку на практическом примере демонстрируют использование основных алгоритмических конструкций.

Литература

1. Перельман Я.И. Живая математика. – М.: Гос. изд. физ.-мат. лит., 1958. – 185 с.
2. Решётка Кардано // Википедия. URL: https://ru.wikipedia.org/wiki/Решётка_Кардано (дата обращения: 05.12.2018).
3. Cardan grille // Wikipedia. URL: https://en.wikipedia.org/wiki/Cardan_grille (дата обращения: 05.12.2018).
4. 1712. Шифровальная решётка // Timus Online Judge. URL: <http://acm.timus.ru/print.aspx?space=1&num=1712> (дата обращения: 05.12.2018).
5. Душкин Р.В. Решётки Кардано // Потенциал (Математика. Физика. Информатика). – 2019. – № 07 (175). – С. 54–60.

Приложение А**Исходный текст программы**

```
цел N = 4;  
сим таб Решетка[1 : N, 1 : N];  
  
алг Потенциал  
нач  
    лит СтрокаИсходная, СтрокаЗашифрованная, СтрокаРасшифрован-  
ная;  
  
    ИсходнаяРешетка(".*...*.....*.....*");  
    ВывестиРешетку;  
  
    СтрокаИсходная := "АБВГДЕЕЖЗИЙКЛМНО";  
    СтрокаЗашифрованная := Зашифровать(СтрокаИсходная);  
    СтрокаРасшифрованная := Расшифровать(СтрокаЗашифрованная);  
  
    вывод СтрокаИсходная, нс;  
    вывод СтрокаЗашифрованная, нс;  
    вывод СтрокаРасшифрованная, нс;  
кон  
  
алг ИсходнаяРешетка(лит РешеткаТекст)  
нач  
    цел i, j;  
  
    нц для i от 1 до N  
        нц для j от 1 до N  
            Решетка[i, j] := РешеткаТекст[(i - 1) * N + j];  
        кц  
    кц  
кон  
  
алг ПовернутьРешетку  
нач  
    сим таб Рез[1 : N, 1 : N];  
    цел i, j;  
  
    нц для i от 1 до N  
        нц для j от 1 до N  
            Рез[j, N + 1 - i] := Решетка[i, j];  
        кц  
    кц  
    нц для i от 1 до N  
        нц для j от 1 до N  
            Решетка[i, j] := Рез[i, j];  
        кц  
    кц  
кон
```

алг ВывестиРешетку

нач

цел i, j ;

нц для i от 1 до N

нц для j от 1 до N

вывод Решетка[i, j];

кц

вывод нс;

кц

кон

алг лит Зашифровать (лит СтрокаИсходная)

нач

лит СтрокаЗашифрованная;

сим таб СтрокаЗашифрованнаяМассив[1 : N , 1 : N];

цел Счетчик;

цел i, j, k ;

СтрокаЗашифрованная := "";

Счетчик := 0;

нц для k от 1 до N

нц для i от 1 до N

нц для j от 1 до N

если Решетка[i, j] = "*"

то

Счетчик := Счетчик + 1;

СтрокаЗашифрованнаяМассив[i, j] := СтрокаИсход-

ная[Счетчик];

все

кц

кц

ПовернутьРешетку;

кц

нц для i от 1 до N

нц для j от 1 до N

СтрокаЗашифрованная := СтрокаЗашифрованная + СтрокаЗашифрованнаяМассив[i, j];

кц

кц

знач := СтрокаЗашифрованная;

кон

```
алг лит Расшифровать (лит СтрокаЗашифрованная)
нач
  лит СтрокаРасшифрованная;
  цел i, j, k;

  СтрокаРасшифрованная := "";

  нц для k от 1 до N
    нц для i от 1 до N
      нц для j от 1 до N

        если Решетка[i, j] = "*"
          то
            СтрокаРасшифрованная := СтрокаРасшифрованная +
СтрокаЗашифрованная[(i - 1) * N + j];
          все
        кц
      кц
    кц

    ПовернутьРешетку;
  кц

  знач := СтрокаРасшифрованная;
кон
```

Приложение Б

Результаты работы программы

```
. * . .
* . . .
. . * .
. . . *
АБВГДЕЁЖЗИЙКЛМНО
ЗАДЛВИМЕНЁВЙЖОКГ
АБВГДЕЁЖЗИЙКЛМНО
```

Юмор**Юмор****Юмор****Юмор****Юмор****Юмор**

* * *

Системный администратор должен думать на один-два клика дальше мышки.

* * *

Программисты на работе общаются двумя фразами: «непонятно» и «вроде работает».

* * *

Программист — это профессиональный конвертер галлюцинаций заказчика в жесткую формальную систему.