

# ПРОБЛЕМНО-ОРИЕНТИРОВАННЫЙ ЯЗЫК ОПИСАНИЯ ГЕНЕРАЦИИ QR-КОДОВ

## PROBLEM-ORIENTED DESCRIPTION LANGUAGE FOR GENERATION OF QR CODES

**V. Marinin  
S. Koryagin**

*Summary.* This article shows and explains the implementation of a problem-oriented language based on the idea of encrypting data into a QR representation. The set of interpreted information is described as an object placed in YAML container, that consists a set of properties and their inherent values. The methods of obtaining the result using various tools for correcting encrypted messages are considered.

*Keywords:* QR-code generation, translation methods, domain-specific languages.

**Маринин Вячеслав Петрович**

Российский Технологический Университет МИРЭА  
marininvp@yandex.ru.

**Корягин Сергей Викторович**

кандидат технических наук,  
Российский Технологический Университет МИРЭА  
dongenealog2003@mail.ru

*Аннотация.* В данной статье показана и объяснена реализация проблемно-ориентированного языка на основании идеи шифрования данных в QR-представление. Совокупность интерпретируемой информации описана в виде объекта, помещённого в YAML контейнер, состоящего из набора свойств и присущих им значений. Рассмотрены способы получения результата с использованием разных инструментов коррекции зашифрованных сообщений.

*Ключевые слова:* генерация QR-кодов, методы трансляции, проблемно-ориентированные языки программирования.

### Введение

Постепенное совершенствование технологий и появление новых огромных спектров информации с каждым годом только увеличивают требования к её хранению, передаче и обработке. В связи с этим происходит постоянная реконструкция в формировании надлежащих мер для обеспечения должного уровня защищённости. Основным способом реализации безопасности в информационной среде является кодирование, позволяющее отправителю обеспечивать должный уровень надёжности на всех этапах взаимодействия с данными.

Значительная массовость использования штрих-кода привела к тому, что объем информации, который возможно закодировать в нем, стал недостаточным для большинства пользователей. Тогда многие мировые умы начали экспериментировать с преобразованием небольшого количества информации по новому способу, отображая всё на единой картинке. И уже в 1994 году, в Японии, появился полноценный проект, серьёзно повлиявший на мировую индустрию в области оперативной трансляции, компактности хранения и доступности лёгкой передачи информации. Он получил название QR-код, что расшифровывается как «quick response», в переводе с английского — «быстрый отклик (ответ)» [1].

QR-коды — это квадратные двухмерные штрих-коды. Их разработчик — Масахиро Хара, главный инженер

японской компании Denso Wave, являвшейся дочерней компанией Toyota. Изначально QR-код использовался для транспортных задач, с целью отслеживания движения автомобилей и составных конструкций в процессе производства. Основным достоинством QR-кода стало то, что он быстро и легко считывается при помощи сканирующего оборудования [2].

Само информационное поле, представляющее из себя матрицу, хоть и кажется совершенно неразборчивой для человеческого глаза совокупностью множества квадратов и линий, предоставляет широкие возможности эксплуатации различных наборов данных. В сравнении с ранее используемыми штрих-кодами, содержимое хранится в некоторой плоскости, задействующей оба измерения — как горизонталь, так и вертикаль. Значащие квадраты, используемые в матрице, могут вмещать в себя тысячи буквенно-цифровых символов. Те же штрих-коды позволяют зашить в себя только несколько чисел [3]. Благодаря системе коррекции ошибок, прочесть QR-код можно даже в том случае, если часть кода отсутствует или повреждена. Чем выше степень коррекции ошибок, тем меньше данных можно поместить в QR-код [4]. Основными элементами хранимой информации являются символы, которые могут быть закодированы при помощи различных возможностей форматирования текста, что в связке с аппаратом для сканирования таких изображений позволяет безошибочно распознавать назначение кода и предлагать необходимое решение.

### Обзор реализованных решений

Реализованные решения разработанного программного обеспечения основываются на использовании YAML-контейнера как хранилища кодируемых данных, изучения и переработки модуля qrcode в языке программирования Python, а также формата png, используемого для отображения полученного QR-кода.

YAML-контейнер является форматом сериализации данных, концептуально близким к языкам разметки, но ориентированным на удобство ввода-вывода типичных структур данных многих языков программирования.[5]

Модуль qrcode в языке программирования Python описывает большинство возможностей работы с QR-кодами.

Формат png используется в качестве растрового формата хранения графической информации.

### Критика обзора реализованных решений

Особенности YAML-контейнера:

- 1) Поддержка структур и форматов данных, понятных для языков программирования.
- 2) Возможность лёгкой интерпретации содержимой информации как между элементами интегрируемой системы.
- 3) Использование записи объектов данных, а не отдельных полей.
- 4) Простота в восприятии для пользователя.

Из недостатков следует отметить: [6]

- 1) Отсутствие собственного определителя типов данных. Он наследует их от языка-родителя при помощи обработки методом синтаксического анализа.
- 2) Сложности, связанные с совместимостью использования разными языками программирования.

Модуль qrcode является многофункциональной структурой для получения итогового QR кода, но при этом весьма тяжеловесен и объёмен.

Png является универсальным хранилищем графической информации и используется повсеместно.

### Алгоритмические особенности проектировки

На примере показанных ниже пунктов объясню логику работы шифрования полученных программой символов в элементы QR-кода.

#### 1. Размеры кодов [7]

Существует 40 возможных размеров QR-кодов. Официальная, самая базовая вариация называется Версия.

Версия 1 — это матрица 21 x 21, Версия 2 — это матрица 25 x 25, а Версия 3 — матрица 29 x 29. Каждая дополнительная версия увеличивает размерность на 4.

$$(X - 1) * 4 + 21 \text{ (где } X \text{ — номер версии)}$$

Формула 1 — Расчёт размерности QR-кода

Наивысшая ступень, 40, если рассчитать по составленной формуле:  $(40 - 1) * 4 + 21 = 177$ , следовательно, имеет размеры 177 x 177. [8]

Пример QR-кода со всеми составными частями:



Рис. 1. Элементы QR-кода

Шаблоны определения местоположения (Рис. 1) — это три крайних квадрата, являющихся обязательными элементами, используются для обозначения прямоугольного размера двумерного кода. Три шаблона позиционирования имеют белые границы, называемые разделителями для шаблонов определения положения. Причина использования трех вместо четырех означает, что они нужны для идентификации правильности считывания прямоугольника. [9]

Шаблоны синхронизации также используются для позиционирования. Причина в том, что QR-код имеет 40 размеров, и если размер слишком велик, требуется стандартная строка, в противном случае она может быть перекошена при сканировании.

#### 2. Информация о формате

Для хранения информации о версии, выше 6-ой, отведено определённое место в матрице размерами 3 x 6, занимаемое в двух областях.

#### 3. Информация о коде данных и коде исправления ошибок

В остальных местах хранится код данных и код исправления ошибок.

#### 4. Цифровое кодирование [7]

Используемый диапазон: от 0 до 9. Если количество кодируемых цифр не кратно 3, то оставшиеся 1 или

Таблица кодирования

| Char. : Value |
|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|
| 0:0           | 6:6           | C             | :12           | I             | :18           | O             | :24           | U             | :30           | SP            | :36           | ..            | :42           |
| 1:1           | 7:7           | D             | :13           | J             | :19           | P             | :25           | V             | :31           | \$            | :37           | /             | :43           |
| 2:2           | 8:8           | E:14          |               | K:20          |               | Q:26          |               | W:32          |               | %:38          |               | ::44          |               |
| 3:3           | 9:9           | F:15          |               | L:21          |               | R:27          |               | X:33          |               | .:39          |               |               |               |
| 4:4           | A:10          | G:16          |               | M:22          |               | S:28          |               | Y:34          |               | +:40          |               |               |               |
| 5:5           | B:11          | H:17          |               | N:23          |               | T:29          |               | Z:35          |               | —:41          |               | .             |               |

2 цифры будут преобразованы в 4 или 7 битов, а все остальные 3 цифры будут скомпилированы в 10, 12, 14 битов. Длина зависит от размера QR-кода (Табл. 1).

#### 5. Кодировка символов [7]

Совокупность знаков: цифры от 0 до 9, прописные буквы от A до Z (без строчных букв) и символы \$% \* + - . / : , также пробелы. Они составляют таблицу индексов символов. SP — пробел, Char — символ, а Value — значение индекса (Табл. 1). Процесс кодирования состоит в том, чтобы сгруппировать символы попарно, а затем преобразовать их в 45-ричное представление, и как итог, привести в двоичный 11-битный код. Режим кодирования и количество символов должны быть скомпилированы в 9, 11 или 13 двоичных файлов в соответствии с различными размерами версии (Табл. 1).

#### 6. Байтовая кодировка [7]

Символы от 0 до 255. Некоторые сканеры двумерного кода могут автоматически определять, является ли это кодировкой UTF-8.

#### 7. Программный модуль

Программный модуль реализован на языке программирования Python.

#### Постановка задачи

С учётом всех обозначенных факторов, создаётся программное обеспечение, которое будет реализовывать поставленную задачу.

Уникализация описания приведённых объектов базируется на механизме выбора нужных пользователю свойств, набор из которых и составит описывающий их QR-код. Каждое из допустимых языком свойств отвечает за определённую черту шифра, которую после его получения можно распознать. Эти свойства формируют единую запись в YAML-контейнер.

Описание возможных для использования свойств: номер версии (отвечает за размер QR-кода), текст кодируемой информации, величина сторон квадратов, размер боковой границы относительно QR-кода, цвет шифра и фона. Результат формирования QR-кода будет представлен в виде png-изображения.

Основные качества проблемно-ориентированного языка описания генерации QR-кодов:

- Простота работы с приложением, гарантированное получение намеченного результата;
- Связь текстового и визуального отображений, способных создать полный цикл преобразования и его проверки при помощи сканирующего средства;

Отличительной чертой такого программно-ориентированного языка является преобразование объекта данных в визуальное зашифрованное представление.

#### Формальное описание проблемно-ориентированного языка

Формат записи объекта в YAML-контейнер:

```
— version: 1
text: «Супер»
box_size: 10
border: 4
fill_color: «black»
back_color: «white»
```

Правила записи:

- 1) Перед головным элементом ставится флаг записи «—»
- 2) Каждая запись производится через Tab
- 3) После указанного свойства ставится «:» и пробел, только после этого указывается значение свойства, удовлетворяющее нижеприведённым условиям

Входной язык включает в себя следующие возможные конструкции, являющиеся свойствами генерируемого QR-кода:

- 1) `version` — Параметр представляет собой целое число от 1 до 40, которое управляет размером QR-кода, в зависимости от указанного значения версии (наименьшая версия 1-матрица 21x21)
- 2) `text` — Русские и латинские буквы
- 3) `box_size` — Параметр управляет количеством пикселей в каждой «коробке» QR-кода.
- 4) `border` — Параметр определяет, сколько ячеек должно быть толщиной границы (по умолчанию 4, что является минимальным в соответствии со спецификациями)
- 5) `fill_color` — Цвет окраски QR
- 6) `back_color` — Цвет окраски фона QR

Форматы записи значений для каждого из свойств, описанных выше:

- 1) None или целочисленное значение от 0 до 40
- 2) Произвольный кодируемый текст, набор символов, длиной не более 20-ти
- 3) Целочисленное значение
- 4) Целочисленное значение
- 5) Текстовое представление цвета на английском языке (Например: `black`)
- 6) Текстовое представление цвета на английском языке (Например: `white`)

Пример 1:

Набор желаемых свойств:

```
1 - version: 2
2 text: "Нужный результат"
3 box_size: 15
4 fill_color: "black"
5 back_color: "white"
```

Рис. 2.1. Свойства объекта

Полученное программой QR-представление:



Рис. 2.2. Сгенерированный QR-код

Проверка сканирующим устройством:

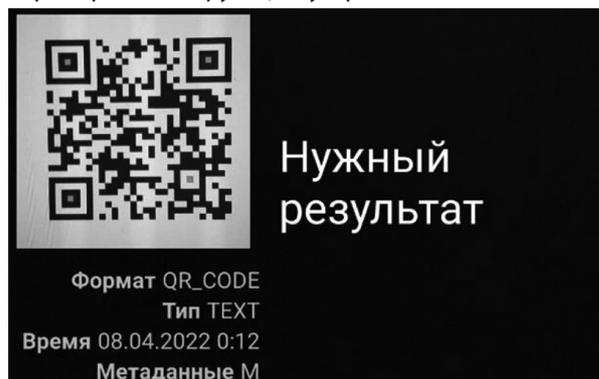


Рис. 2.3. Раскодированный QR-код

Пример 2:

Набор желаемых свойств:

```
1 - version: 5
2 text: "QR-code"
3 box_size: 11
4 border: 10
```

Рис. 2.4. Свойства объекта

Полученное программой QR-представление:



Рис. 2.5. Сгенерированный QR-код

Проверка сканирующим устройством:

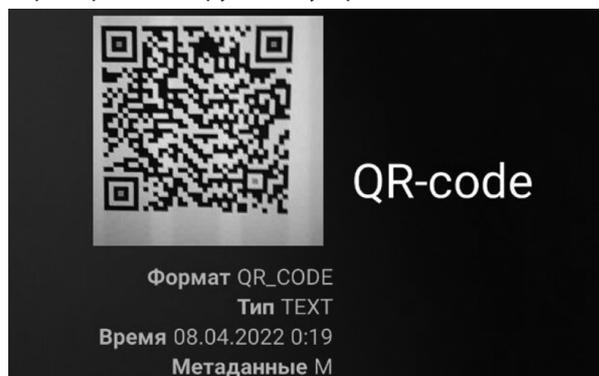


Рис. 2.6. Раскодированный QR-код

Пример 3:

Набор желаемых свойств:

```
1 - version: 1
2 text: "То, что нужно"
3 box_size: 10
4 border: 5
5 fill_color: "green"
6 back_color: "blue"
```

Рис. 2.7. Свойства объекта

Полученное программой QR-представление:



Рис. 2.8. Сгенерированный QR-код

## Выводы

Как результат проверки работы программы на рассмотренных выше примерах с разными составляющими значений объектов данных, везде был получен корректный результат в виде работающего QR-кода, ко-

Проверка сканирующим устройством:

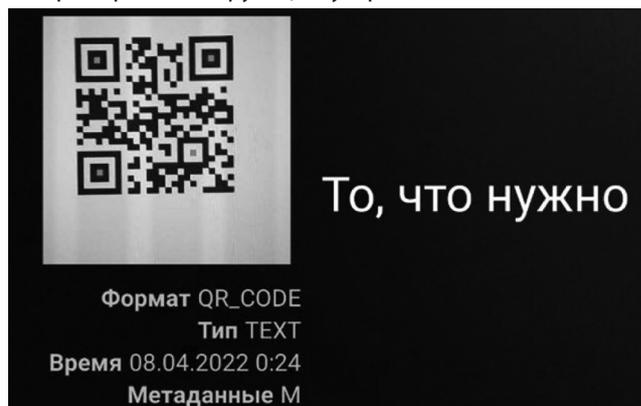


Рис. 2.9. Раскодированный QR-код

торый удалось считать и распознать закодированные данные.

По итогам проведённой работы был разработан проблемно-ориентированный язык программирования со специализированным интерпретирующим средством, соответствующим данному языку. В него заложен необходимый функционал, требуемый для формирования результатов, присутствует возможность дальнейшего его расширения. Например, данную программу можно дооснастить навыком автоматизации внесения полученных QR-представлений на информационные ресурсы, или расширения спектра свойств под требуемые задачи.

Разработанное приложение построено на принципах объектно-ориентированного программирования, что позволяет более гибко использовать корректно введенные пользователем данные при дальнейшей их обработке и кодировании.

## ЛИТЕРАТУРА

1. Рассел Д. «QR-код» — Пер. с англ. — М.: Книга по Требованию — 2012. — 918 с., ил. ISBN: 978-5-5128-2491-7
2. Ковалёв А. «QR-коды, их свойства и применение»: [Электронный ресурс] // Электронная статья — 2016 — URL: <https://moluch.ru/archive/114/29398/>.
3. «Что такое QR-код и как с ним работать»: [Электронный ресурс] // Электронная статья — М: авт. редакция Computer Bild, Журнал ComputerBild — №12 — 2011 — URL: <http://oventamarket.ru/chto-takoe-qr-kod-i-kak-s-nim-rabotat/>.
4. «Что внутри любого QR-кода»: [Электронный ресурс] // Электронная статья — 2022 — URL: <https://thecode.media/qr-code/>.
5. «YAML»: [Электронный ресурс] // Электронная статья — 2017 — URL: <https://ru.wikipedia.org/wiki/YAML>.
6. «Документ предварительного исследования YAML»: [Электронный ресурс] // Электронная статья — 2020 — URL: <https://russianblogs.com/article/86921116577/>.
7. «Детали и принципы генерации QR-кода»: [Электронный ресурс] // Электронная статья — 2020 — URL: <https://russianblogs.com/article/5122315167/>.
8. Ткачева М. «Оценка допустимых преобразований QR-code»: [Электронный ресурс] // Электронная статья — М: Журнал Известия Тульского государственного университета. Технические науки. — №3 — 2013 — URL: <https://cyberleninka.ru/article/n/otsenka-dopustimyh-preobrazovaniy-qr-code/viewer>.
9. Семенов А. «Введение в коды Рида-Соломона: принципы, архитектура и реализация.»: [Электронный ресурс] // Электронная статья — 2013 — URL: <https://intuit.ru/studies/courses/9/9/lecture/268?page=2>.

© Маринин Вячеслав Петрович (marininvp@yandex.ru); Корягин Сергей Викторович (dongenealog2003@mail.ru)  
Журнал «Современная наука: актуальные проблемы теории и практики»