Муниципальное бюджетное общеобразовательное учреждение Городского округа Балашиха "Гимназия №11"

**Проект на тему:**

‘’Создание кастомных QR-кодов’’

**Выполнил:**

учащийся 10/04 класса

Захарова Т. А.

**Научный руководитель:**

учитель информатики

Галкина О. А.

г. Балашиха, 2024.

**Содержание**

[Введение: 3](#_Toc181268797)

[Приложение 1: Установка необходимых библиотек 4](#_Toc181268798)

[Глава 1. Генерация простого QR-кода: 7](#_Toc181268799)

[Заключение: 9](#_Toc181268800)

**Введение:**

QR-коды (Quick Response codes) — это двухмерные штрих-коды, которые содержат закодированную информацию, которую можно быстро считать с помощью смартфонов и других устройств. За последние годы они стали популярным инструментом для передачи данных, от ссылок на веб-ресурсы до информации о продуктах. Использование кастомных QR-кодов позволяет не только передавать информацию, но и добавить уникальный визуальный стиль, что делает их привлекательными для брендинга и маркетинга. В данном проекте мы рассмотрим, как сгенерировать кастомные QR-коды на языке программирования Python, а также добавим возможности для изменения их дизайна.

**Цели и задачи**

Цель проекта — изучить и разработать программное решение для генерации кастомных QR-кодов с использованием языка программирования Python.

**Задачи**:

1. Изучить базовые принципы генерации QR-кодов.

2. Исследовать библиотеку Python для работы с QR-кодами.

3. Реализовать генерацию QR-кодов с костюмным дизайном.

4. Настроить возможность изменения цвета, размера и логотипов в QR-кодах.

5. Провести тестирование сгенерированных QR-кодов на совместимость с различными устройствами.

**Приложение 1:** Установка необходимых библиотек

Для начала работы с проектом необходимо предварительно установить следующие библиотеки:

1. Qrcode – библиотека для Python, позволяет генерировать QR-коды с помощью нескольких строк кода;
2. Pillow - библиотека для Python для работы с изображениями в Python;
3. Numpy - это открытая бесплатная **библиотека** для **Python**. Она ускоряет работу с многомерными массивами и матрицами, а также позволяет вычислять много высокоуровневых математических функций при работе с массивами данных;
4. Matplotlib – это библиотека на языке программирования Python для визуализации данных двумерной и трёхмерной графикой.

Установим библиотеку qrcode для Python в среде программирования PyCharm IDE, для введем в командной строке команду:

* pip install qrcode.

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

Результат выполнения установки библиотеки qrcode в среде программирования PyCharm IDE показан на рис. 1.

Установим библиотеку pillow для Python в среде программирования PyCharm IDE, для введем в командной строке команду:

* pip install pillow.

Изображение выглядит как текст, Шрифт, снимок экрана

Автоматически созданное описание

Результат выполнения установки библиотеки pillow в среде программирования PyCharm IDE показан на рис. 2.

Установим библиотеку numpy для Python в среде программирования PyCharm IDE, для введем в командной строке команду:

* pip install numpy.

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

Результат выполнения установки библиотеки numpy в среде программирования PyCharm IDE показан на рис. 3.

Установим библиотеку matplotlib для Python в среде программирования PyCharm IDE, для введем в командной строке команду:

* pip install matplotlib.

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

Результат выполнения установки библиотеки matplotlib в среде программирования PyCharm IDE показан на рис. 4.

**Глава 1.** Генерация простого QR-кода:

import qrcode  
  
*# Данные для кодирования*data0 = "http://yandex.ru"  
data1 = "http://ya.ru"  
data2 = "http://rbc.ru"  
data3 = "http://vc.ru"  
data4 = "https://inf-ege.sdamgia.ru/"  
data5 = "https://inf-oge.sdamgia.ru/"  
  
name\_qr\_code0 = 'img/qr\_0.png'  
name\_qr\_code1 = 'img/qr\_1.png'  
name\_qr\_code2 = 'img/qr\_2.png'  
name\_qr\_code3 = 'img/qr\_3.png'  
name\_qr\_code4 = 'img/qr\_4.png'  
name\_qr\_code5 = 'img/qr\_5.png'  
  
*# Генерация QR-кода*qr0 = qrcode.QRCode(  
 version=1,  
 error\_correction=qrcode.constants.ERROR\_CORRECT\_L,  
 box\_size=10,  
 border=4  
)  
  
qr1 = qrcode.QRCode(  
 version=1,  
 error\_correction=qrcode.constants.ERROR\_CORRECT\_L,  
 box\_size=10,  
 border=4  
)  
qr2 = qrcode.QRCode(  
 version=1,  
 error\_correction=qrcode.constants.ERROR\_CORRECT\_L,  
 box\_size=10,  
 border=4  
)  
  
qr3 = qrcode.QRCode(  
 version=1,  
 error\_correction=qrcode.constants.ERROR\_CORRECT\_L,  
 box\_size=10,  
 border=4  
)  
  
qr4 = qrcode.QRCode(  
 version=1,  
 error\_correction=qrcode.constants.ERROR\_CORRECT\_L,  
 box\_size=10,  
 border=4  
)  
  
qr5= qrcode.QRCode(  
 version=1,  
 error\_correction=qrcode.constants.ERROR\_CORRECT\_L,  
 box\_size=10,  
 border=4  
)  
  
qr0.add\_data(data0)  
qr1.add\_data(data1)  
qr2.add\_data(data2)  
qr3.add\_data(data3)  
qr4.add\_data(data4)  
qr5.add\_data(data5)  
  
qr0.make(fit=True)  
qr1.make(fit=True)  
qr2.make(fit=True)  
qr3.make(fit=True)  
qr4.make(fit=True)  
qr5.make(fit=True)  
  
*# Создание и сохранение изображения*img0 = qr0.make\_image(fill='black', back\_color='white')  
img1 = qr1.make\_image(fill='black', back\_color='white')  
img2 = qr2.make\_image(fill='black', back\_color='white')  
img3 = qr3.make\_image(fill='black', back\_color='white')  
img4 = qr4.make\_image(fill='black', back\_color='white')  
img5 = qr5.make\_image(fill='black', back\_color='white')  
  
img0.save(name\_qr\_code0)  
img1.save(name\_qr\_code1)  
img2.save(name\_qr\_code2)  
img3.save(name\_qr\_code3)  
img4.save(name\_qr\_code4)  
img5.save(name\_qr\_code5)

**Заключение**

В данном проекте был рассмотрен процесс генерации QR-кодов на языке программирования Python с использованием библиотек **qrcode**, **myqr**. Мы рассмотрели, как создать простой QR-код, а также как кастомизировать его, изменяя цвета, добавляя логотипы, создания qr-кода по изображению. Это позволяет использовать QR-коды не только для передачи информации, но и для маркетинговых и брендовых задач. Будущие улучшения могут включать генерацию динамических QR-кодов и оптимизацию для более сложных дизайнерских решений.