

Простой шифратор сообщений (шифр Цезаря) на языке Python

Кокухина Софья Александровна

Приамурский государственный университет им.Шолом-Алейхема

Студент

Аннотация

Цель статьи – разработка программы на языке Python для шифрования и дешифрования сообщений с использованием шифра Цезаря. Программа реализована в среде PyCharm и позволяет пользователю вводить текст и ключ сдвига, а затем получать зашифрованную или расшифрованную строку. Проект демонстрирует основы работы со строками, циклами, условиями и функциями в Python, а также базовые принципы симметричной криптографии.

Ключевые слова: Python, шифр Цезаря, криптография, шифрование, дешифрование, PyCharm.

Caesar Cipher: Simple Message Encryptor in Python

Kokukhina Sofya Alexandrovna

Sholom-Aleichem Priamursky State University

Student

Abstract

The article presents the development of a Python program for encrypting and decrypting messages using the Caesar cipher. The program is implemented in PyCharm and allows the user to input text and a shift key, then outputs the encrypted or decrypted string. The project demonstrates string manipulation, loops, conditions, and functions in Python, as well as basic principles of symmetric cryptography.

Keywords Python, Caesar cipher, cryptography, encryption, decryption, PyCharm.

1. Введение

1.1 Актуальность

Криптография – это наука о защите информации, которая находит применение в повседневной жизни, от защиты паролей до обеспечения безопасности переписки. Шифр Цезаря, предложенный более двух тысяч лет назад, является одним из самых простых и наглядных примеров симметричного шифрования. Его изучение помогает понять основные принципы кодирования данных, работу с символьными наборами и алгоритмами преобразования.

1.2 Обзор исследований

В статье Д.В. Капулина и Р.Ю. Царева изложены основные принципы проектирования и разработки программного, технического и

информационного видов обеспечения отказоустойчивых информационных систем управления предприятиями. Рассмотрены особенности построения защищенной информационной инфраструктуры предприятия для организации безопасного обмена производственной информацией. Предложены метод и устройство организации зашифрованного канала беспроводного обмена данными в рамках сетевой инфраструктуры предприятия. Приведены подходы к оценке и повышению надежности программной, информационной и аппаратной составляющих информационных систем [1]. В публикации Д.Д.Осипова и Ю.А. Журавлева рассмотрен алгоритм RSA как средство безопасной передачи информации. Особое внимание уделяется устройству для шифрования и дешифрования данных, а также режимам работы этого устройства [2]. Статья К.Д. Ермакова и Ю.В. Кузнецова посвящена материалам разработки криптографического алгоритма блочного шифрования «Люцифер». Приведены описания двух версий указанного криптоалгоритма, а также отчет, содержащий основные принципы создания и программно-аппаратной реализации данного блочного шифра [3].

1.3 Цель исследования

Целью данного исследования является разработка простой программы на Python для шифрования и дешифрования сообщений методом Цезаря с возможностью задания пользователем ключа сдвига.

1.4 Материалы и методы

В качестве инструмента для разработки программы выбран язык программирования Python версии 3.12. Выбор языка обусловлен его простотой, обширной стандартной библиотекой и широким применением в образовательных проектах. Для написания и тестирования кода выбрана среда разработки PyCharm Community Edition, которая предоставляет удобный интерфейс, подсветку синтаксиса и встроенный терминал для запуска программ.

Алгоритм шифра Цезаря основан на циклическом сдвиге символов алфавита на фиксированное количество позиций. При шифровании каждый символ исходного текста заменяется символом, находящимся в алфавите на заданное количество шагов вперед, а при дешифровании — на то же количество шагов назад. Для реализации выбран английский алфавит в верхнем регистре (A–Z), что упрощает алгоритм и позволяет сосредоточиться на механизме шифрования.

Подготовка проекта выполнялась в несколько этапов. Сначала в PyCharm создавался новый проект, в котором указывался установленный ранее интерпретатор Python 3.12. Затем в корне проекта создавался файл `main.py`, в который добавлялись все необходимые функции и основной код программы. Структура проекта оставалась минимальной, так как все функции шифрования и дешифрования реализовывались в одном файле (Рис.1).

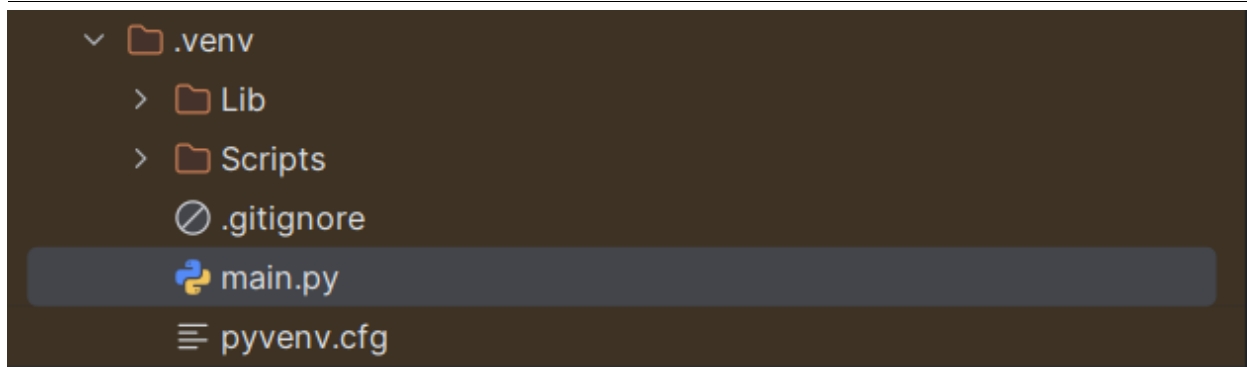


Рисунок 1 – Структура проекта

2 Практическая часть

Для начала нужно перейти к объявлению и инициализации основных переменных для реализации программы шифрования методом Цезаря. Определяется переменная `alphabet`, содержащая все буквы английского алфавита в верхнем регистре. Создаются переменные `shift` для хранения величины сдвига, `text` для исходного текста, а также `result` для итоговой строки (Рис. 2).

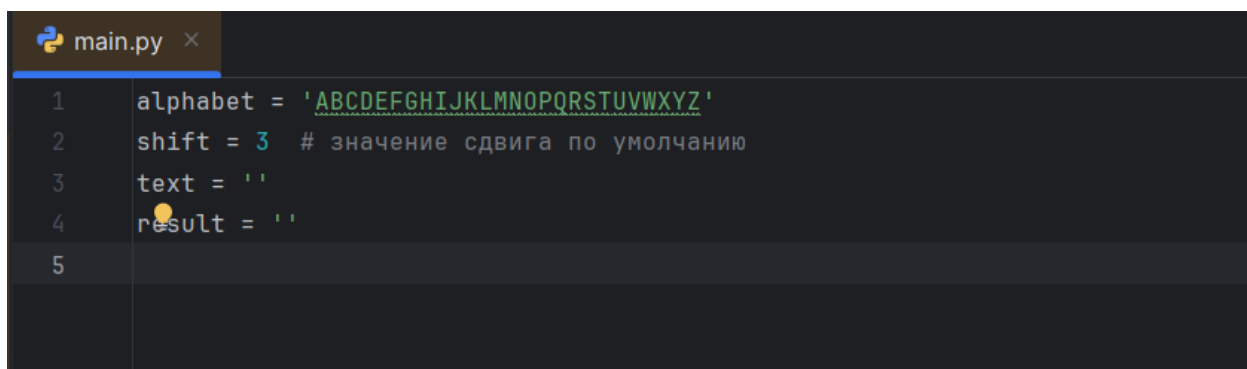


Рисунок 2 – Объявление переменных программы

Далее реализуется функция `encrypt (text, shift)`, выполняющая шифрование. Для каждого символа проверяется, присутствует ли он в алфавите. Если да, вычисляется его новый индекс с учётом сдвига по модулю длины алфавита. Если символ отсутствует в алфавите (например, пробел или знак препинания), он добавляется в результат без изменений (Рис. 3).

```
5   def encrypt(text, shift):
6       result = ''
7       for char in text:
8           if char.upper() in alphabet:
9               index = alphabet.index(char.upper())
10              new_index = (index + shift) % len(alphabet)
11              result += alphabet[new_index]
12           else:
13               result += char
14       return result
15
```

Рисунок 3 – Функция шифрования сообщений

Аналогично реализуется функция `decrypt(text, shift)`, которая сдвигает символы в обратную сторону, восстанавливая исходное сообщение (Рис. 4).

```
16  def decrypt(text, shift):
17      result = ''
18      for char in text:
19          if char.upper() in alphabet:
20              index = alphabet.index(char.upper())
21              new_index = (index - shift) % len(alphabet)
22              result += alphabet[new_index]
23          else:
24              result += char
25      return result
26
27
```

Рисунок 3 – Функция дешифрования сообщений

В основной части программы пользователю предлагается ввести текст и ключ сдвига. Затем он выбирает режим работы — шифрование или дешифрование. В зависимости от выбора вызывается соответствующая функция, и результат выводится на экран (Рис. 5).

```
27 > if __name__ == "__main__":  
28     text = input("Введите сообщение: ")  
29     shift = int(input("Введите ключ (сдвиг): "))  
30     mode = input("Выберите режим (e - шифрование, d - дешифрование): ")  
31  
32     if mode.lower() == 'e':  
33         print("Зашифрованное сообщение:", encrypt(text, shift))  
34     elif mode.lower() == 'd':  
35         print("Расшифрованное сообщение:", decrypt(text, shift))  
36     else:  
37         print("Неверный режим работы!")  
38
```

Рисунок 5 – Основная функция программы

После завершения написания кода программа запускается в PyCharm с помощью комбинации клавиш Shift + F10. В консоли появляется запрос на ввод данных. При выборе режима e («encrypt») и вводе строки HELLO WORLD с ключом 3 программа выведет KHOOR ZRUOG (Рис. 6).

```
Введите сообщение: Hello world  
Введите ключ (сдвиг): 3  
Выберите режим (e - шифрование, d - дешифрование): e  
Зашифрованное сообщение: KHOOR ZRUOG  
  
Process finished with exit code 0
```

Рисунок 6 – Пример работы программы в режиме шифрования

При выборе режима d («decrypt») с теми же параметрами будет восстановлен исходный текст (Рис. 7).

```
Введите сообщение: KHOOR ZRUOG  
Введите ключ (сдвиг): 3  
Выберите режим (e - шифрование, d - дешифрование): d  
Расшифрованное сообщение: HELLO WORLD  
  
Process finished with exit code 0
```

Рисунок 7 – Пример работы программы в режиме дешифрования

3

Выводы В ходе работы был реализован простой, но наглядный пример симметричного шифрования на Python. Программа демонстрирует принципы работы со строками, циклами, условиями и функциями, а также даёт базовое

представление о криптографических алгоритмах. Реализация в PyCharm обеспечивает удобство разработки и тестирования.

Разработанная программа успешно выполняет шифрование и дешифрование строк методом Цезаря. Она корректно обрабатывает как буквы, так и пробелы и знаки препинания, не изменяя символы, отсутствующие в алфавите. Возможность задавать ключ сдвига делает приложение универсальным для различных вариантов шифрования.

Библиографический список

1. Капулин Д. В., Царев Р. Ю., Дрозд О. В., Черниговский А. С. Разработка высоконадежных интегрированных информационных систем управления предприятием. Красноярск: Сибирский федеральный университет, 2015. 181 с.
2. Осипов Д. Д., Журавлева Ю. А., Микаева С. А., Шигапов А. Э. Актуальные проблемы и перспективы развития радиотехнических и инфокоммуникационных систем ("Радиоинфоком-2024") // Сборник научных статей по материалам VIII Международной научно-практической конференции, Москва, 18–22 ноября 2024 года. МИРЭА - Российский технологический университет, 2024. С. 455-458.
3. Ермаков К. Д. и др. Криптоалгоритм «Люцифер». Основы теории современного шифрования. М.: Всероссийский научно-исследовательский институт экспериментальной физики, 2022. 137 с.