**Федеральное государственное образовательное бюджетное**

**учреждение высшего образования**

**«Санкт-Петербургский государственный университет**

**телекоммуникаций им. проф. М. А. Бонч-Бруевича»**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Факультет

инфокоммуникационных сетей и систем

Кафедра

Защищенных сетей и систем

**КУРСОВАЯ РАБОТА**

учебная дисциплина «Технологии и методы программирования»

Тема: **«**Шифр Вернама**»**

Выполнил(а) студент(ка) II курса группы

ИКБ-63 Геращенко А.А.

Проверил Штеренберг С.И.

\_\_\_\_\_\_\_\_\_\_(подпись)

**Оглавление**

[**Введение** 3](#_Toc515485437)

[**1** **Реализация шифра Вернама.** 4](#_Toc515485438)

[**Заключение** 6](#_Toc515485439)

[**Список использованных интернет-ресурсов** 7](#_Toc515485440)

[**Приложение** 8](#_Toc515485441)

# **Введение**

В этой курсовой работе, будет рассмотрен один из вариантов реализации программы шифра Вернама.

**Задание по курсовой работе:** выполнить реализацию шифра Вернама на языке программирования “Python”.

# **Реализация шифра Вернама.**

С помощью библиотеки Tkinter был разработан GUI, а также написан сам код построения графиков.

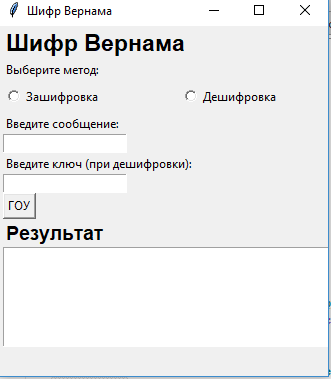


Рисунок 1. Графический интерфейс.

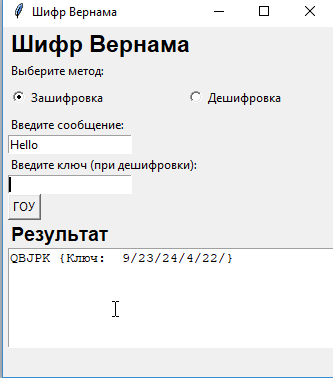


Рисунок 2. Пример шифровки сообщения.

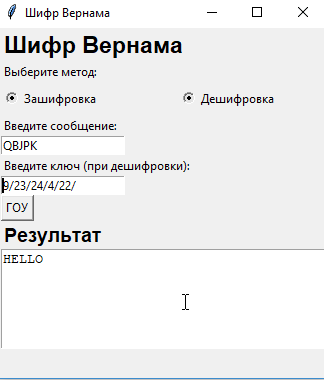


Рисунок 3. Пример дешифровки сообщения.

# **Заключение**

В данной курсовой работе была выполнена программа шифрования Вернама на языке программирования “Python”.

# **Список использованных интернет-ресурсов**

1. <http://www.cyberforum.ru/python/> - шифр замены.
2. <https://mindhalls.ru/exchange-code/> - реализация шифра простой замены.
3. <https://pythonworld.ru/gui> - графический интерфейс (GUI).

# **Приложение**

**from** tkinter **import** \*  
**from** random **import** randint  
**from** re **import** findall  
  
**class** Application(Frame):  
  
 **def** \_\_init\_\_(self, master):  
 *""" Initialize the Frame """* Frame.\_\_init\_\_(self, master)  
 self.grid()  
 self.create\_widgets()  
  
 **def** create\_widgets(self):  
  
 *# Титульник* self.instruction = Label(self, text=**"Шифр Вернама"**, font=(**"arial"**, 17, **"bold"**))  
 self.instruction.grid(row=1, column=0, columnspan=2, padx=5, sticky=W)  
  
 *# Выбор метода* self.method = Label(self, text=**"Выберите метод:"**)  
 self.method.grid(row=2, column=0, columnspan=2, padx=5, sticky=W)  
 self.var1 = IntVar()  
  
 self.option = Radiobutton(self, text=**"Зашифровка"**, pady=5, variable=self.var1, value=1)  
 self.option.grid(row=3, column=0, columnspan=1, padx=5, sticky=W)  
  
 self.option = Radiobutton(self, text=**"Дешифровка"**, pady=5, variable=self.var1, value=2)  
 self.option.grid(row=3, column=1, columnspan=1, padx=5, sticky=W)  
  
 *# Строка сообщения* self.instruction = Label(self, text=**"Введите сообщение: "**)  
 self.instruction.grid(row=4, column=0, columnspan=150, padx=5, sticky=W)  
  
 *# Ввод сообщения* self.message = Entry(self)  
 self.message.grid(row=5, column=0, padx=5, sticky=W)  
  
 *# Строка ключа* self.instruction = Label(self, text=**"Введите ключ (при дешифровки): "**)  
 self.instruction.grid(row=6, column=0, columnspan=2, padx=5, sticky=W)  
  
 *# Ввод ключа* self.key = Entry(self)  
 self.key.grid(row=7, column=0, padx=5, sticky=W)  
  
 *# Принятие* self.submit\_button = Button(self, text=**"ГОУ"**, command=self.caesar)  
 self.submit\_button.grid(row=8, column=0, padx=5, sticky=W)  
  
 *# Результата строка* self.instruction = Label(self, text=**"Результат"**, font=(**"arial"**, 14, **"bold"**))  
 self.instruction.grid(row=9, column=0, columnspan=2, padx=5, sticky=W)  
  
 *# Результат* self.result = Text(self, width=45, height=6, wrap=WORD)  
 self.result.grid(row=10, column=0, columnspan=3, padx=5, sticky=W)  
  
  
  
 **def** caesar(self):  
  
 startMessage = self.message.get().upper()  
 k = (self.key.get())  
 cryptMode = (self.var1.get())  
  
 **def** regular(text):  
 template = **r"[0-9]+"  
 return** findall(template, text)  
  
 **def** encryptDecrypt(mode, message, final=**""**, keys=**""**):  
 **if** mode == 1:  
 **for** symbol **in** message:  
 key = randint(0, 25)  
 keys += str(key) + **"/"** final += chr((ord(symbol) + key - 13) % 26 + ord(**'A'**))  
 keys = **"Ключ: "** + keys  
  
 **return** final, keys  
 **else**:  
 **for** index, symbol **in** enumerate(message):  
 final += chr((ord(symbol) - int(regular(k)[index]) - 13) % 26 + ord(**'A'**))  
 **return** final  
  
  
 self.result.delete(0.0, END)  
 self.option.deselect()  
 self.result.insert(0.0, encryptDecrypt(cryptMode, startMessage))  
  
 **return** (encryptDecrypt(cryptMode, startMessage))  
  
  
root = Tk()  
root.title(**"Шифр Вернама"**)  
root.geometry(**"330x350"**)  
app = Application(root)  
root.mainloop()