

- С. 69-74. DOI: [https://doi.org/10.31521/modecon.V39\(2023\)-10](https://doi.org/10.31521/modecon.V39(2023)-10). (дата звернення : 07.03.2024 р.)
2. Termin.in.ua : веб-сайт. URL : https://termin.in.ua/shtuchnyu-intelekt/#google_vignette (дата звернення: 07.03.2024 р.)
 3. SPEKA : веб-сайт. URL : <https://speka.media/ai/vid-s-do-i-shho-take-stucnii-intelekt-ta-yak-vin-transformuje-svit-xv7039> (дата звернення: 07.03.2024 р.)
 4. АГРОЦЕНТР. Новини сільського господарства та світу : веб-сайт. URL : <https://news.agro-center.com.ua/agri-policy/shtuchnij-intelekt-ta-roboti-jakimi-budut-fermi-majbutnogo.html> (дата звернення: 07.03.2024 р.)

Abstract: *This topic is devoted to artificial intelligence (AI) and its effects on various fields. The key aspects of AI are considered, including the first humanoid robot Sofia. The areas of use of AI are studied, in particular, agriculture, medicine, astronomy, and others. The research provides concrete examples of AI applications, such as Sophia the robot, the use of drones in agronomy, smart forecasting systems in agriculture to improve processes, reduce risks and increase productivity.*

Keywords: *artificial intelligence, agronomy, robot Sophia, agriculture, technologies.*

Науковий керівник: Борян Л.О.,

*старший викладач кафедри економічної кібернетики,
комп'ютерних наук та інформаційних технологій,
Миколаївський національний аграрний університет
м. Миколаїв, Україна*

УДК 004.9

РЕАЛІЗАЦІЯ АЛГОРИТМУ ШИФРУ ЦЕЗАРЯ МОВОЮ ПРОГРАМУВАННЯ PYTHON

Венгура Людмила Олександрівна,

*здобувач вищої освіти спеціальності 122 «Комп'ютерні науки»
Миколаївський національний аграрний університет
м. Миколаїв, Україна*

Анотація: *Шифр Цезаря є одним із найдавніших та найпростіших криптографічних алгоритмів, який використовувався для захисту конфіденційної інформації. Незважаючи на свою простоту, цей метод шифрування все ще широко застосовується для ознайомлення з основами криптографії та розуміння принципів роботи шифрів. У цій доповіді ми розглянемо реалізацію алгоритму шифру Цезаря мовою програмування Python, а також створимо графічний інтерфейс користувача (GUI) за допомогою модуля tkinter.*

Ключові слова: *шифр Цезаря, шифрування, Python, модуль tkinter.*

Шифр Цезаря є типом підстановочного шифру, де кожна літера в відкритому тексті замінюється іншою літерою, яка знаходиться на фіксованій відстані від неї в алфавіті. Ця відстань, яка називається зсувом, є ключем шифрування. Наприклад, якщо зсув дорівнює 3, то літера 'A' буде замінена на 'D', 'B' на 'E' і так далі.

```
# Функція для шифрування
def caesar_encrypt(plaintext, shift):
    ciphertext = ""
    for char in plaintext:
        if char.isalpha(): # Перевірка, чи символ є літерою
            if char.isupper(): # Верхній регістр
                ciphertext += chr((ord(char) + shift - 65) % 26 + 65)
            else: # Нижній регістр
                ciphertext += chr((ord(char) + shift - 97) % 26 + 97)
        else:
            ciphertext += char # Залишаємо непробільні символи без змін
    return ciphertext
```

Функція `caesar_encrypt` приймає два аргументи: `plaintext` (відкритий текст) та `shift` (зсув). Створюється порожній рядок `ciphertext` для зберігання зашифрованого тексту. За допомогою циклу `for` проходимо по кожному символу в `plaintext`. Перевіряємо, чи символ є літерою, використовуючи метод `isalpha()`. Якщо символ є літерою верхнього регістру, використовуємо формулу $(\text{ord}(\text{char}) + \text{shift} - 65) \% 26 + 65$ для зсуву літери та зберігаємо її в `ciphertext`. Якщо символ є літерою нижнього регістру, використовуємо формулу $(\text{ord}(\text{char}) + \text{shift} - 97) \% 26 + 97$ для зсуву літери та зберігаємо її в `ciphertext`. Якщо символ не є літерою, він просто додається до `ciphertext` без змін. Після циклу функція повертає зашифрований текст `ciphertext`. Для розшифрування тексту, зашифрованого шифром Цезаря, потрібно використовувати ту саму функцію, але з від'ємним значенням зсуву.

```
# Функція для розшифрування
def caesar_decrypt(ciphertext, shift):
```

```
    return caesar_encrypt(ciphertext, -shift)
```

Далі створюємо графічний інтерфейс користувача (GUI). Для створення зручного інтерфейсу для взаємодії з нашим алгоритмом шифру Цезаря, ми використаємо модуль `tkinter`, вбудований в Python.

```
import tkinter as tk
```

```
# Функція для обробки події кнопки "Зашифрувати"
```

```
def encrypt():
    plaintext = plaintext_entry.get()
    shift = int(shift_entry.get())
    ciphertext = caesar_encrypt(plaintext, shift)
    result_text.delete(1.0, tk.END)
    result_text.insert(tk.END, ciphertext)
```

```
# Функція для обробки події кнопки "Розшифрувати"
```

```

def decrypt():
    ciphertext = result_text.get(1.0, tk.END)
    shift = int(shift_entry.get())
    plaintext = caesar_decrypt(ciphertext, shift)
    result_text.delete(1.0, tk.END)
    result_text.insert(tk.END, plaintext)
# Створення головного вікна
root = tk.Tk()
root.title("Шифр Цезаря")
# Створення вікна для вводу відкритого тексту
plaintext_label = tk.Label(root, text="Відкритий текст:")
plaintext_label.pack()
plaintext_entry = tk.Entry(root, width=50)
plaintext_entry.pack()
# Створення вікна для вводу зсуву
shift_label = tk.Label(root, text="Зсув:")
shift_label.pack()
shift_entry = tk.Entry(root, width=10)
shift_entry.pack()
# Створення кнопок
encrypt_button = tk.Button(root, text="Зашифрувати", command=encrypt)
encrypt_button.pack()
decrypt_button = tk.Button(root, text="Розшифрувати", command=decrypt)
decrypt_button.pack()
# Створення вікна для відображення результату
result_label = tk.Label(root, text="Результат:")
result_label.pack()
result_text = tk.Text(root, height=5, width=50)
result_text.pack()
# Запуск головного циклу GUI
root.mainloop()

```

Імпортуємо модуль `tkinter`. Визначаємо функції `encrypt()` та `decrypt()`, які обробляють події кнопок "Зашифрувати" та "Розшифрувати" відповідно. Створюємо головне вікно `root` та встановлюємо його заголовок. Створюємо вікно для вводу відкритого тексту за допомогою `tk.Label` та `tk.Entry`. Створюємо вікно для вводу зсуву за допомогою `tk.Label` та `tk.Entry`. Створюємо кнопки "Зашифрувати" та "Розшифрувати" за допомогою `tk.Button` та зв'язуємо їх з відповідними функціями. Створюємо вікно для відображення результату за допомогою `tk.Label` та `tk.Text`. Запускаємо головний цикл GUI за допомогою `root.mainloop()`.

У цій доповіді ми розглянули реалізацію алгоритму шифру Цезаря мовою програмування Python. Ми створили функції для шифрування та розшифрування тексту, а також розробили графічний інтерфейс користувача за допомогою модуля `tkinter`. Хоча шифр Цезаря є надзвичайно простим і не забезпечує достатнього рівня безпеки для захисту важливої інформації, він

слугує чудовим прикладом для ознайомлення з основами криптографії та розуміння принципів роботи шифрів. Ця реалізація може бути використана як основа для подальшого вивчення та розробки більш складних криптографічних алгоритмів.

Список використаних джерел:

1. Fastovets V. Аналіз та програмна реалізація модифікованого криптографічного шифру Вернама та шифру Цезаря. Vehicle and electronics. Innovative technologies, 2021, (20), 62-67.
2. Bhanot R., Hans R. A review and comparative analysis of various encryption algorithms. 2015. Int. J. Secur. its Appl. 9. 4. 289–306.

***Annotation:** The Caesar cipher is one of the oldest and simplest cryptographic algorithms used to protect confidential information. Despite its simplicity, this method of encryption is still widely used to introduce the basics of cryptography and understand how ciphers work. In this talk, we will look at the implementation of the Caesar cipher algorithm in the Python programming language, and we will also create a graphical user interface (GUI) using the tkinter module.*

***Key words:** Caesar cipher, encryption, Python, tkinter module.*

Науковий керівник: Тищенко С.І.,

к.п.н., доцент,

доцент кафедри економічної кібернетики, комп'ютерних наук та інформаційних технологій,

Миколаївський національний аграрний університет

м. Миколаїв, Україна

УДК 004.41

РЕАЛІЗАЦІЯ ПОПУЛЯРНИХ АЛГОРИТМІВ СОРТУВАННЯ МОВОЮ PYTHON

Вишневський Олег Олександрович,

здобувач вищої освіти спеціальності 122 «Комп'ютерні науки»

Миколаївський національний аграрний університет

м. Миколаїв, Україна

***Анотація:** В дослідженні проаналізовано найпростіші алгоритми сортування та наведено програмний код їх реалізації в мові програмування Python. Розглянуто алгоритми сортування бульбашкою або Bubble Sort, сортування за вибором або Selection Sort та сортування вставками або Insertion Sort.*