## **Державний університет інформаційно-комунікаційних технологій**

**Практична робота №8**

**з дисципліни: Технології штучного інтелекту**

**Тема: Розробка інтерактивних діалогів з моделлю ChatGPT 3.5 на Python**

Виконав:  
студент ДУІКТ  
Тертишний В.Ю.

група: ШІДМ-51

м.Київ

**Мета:** навчитися створювати інтерактивні діалоги з використанням моделі ChatGPT 3.5 на Python, розвиваючи навички роботи з API OpenAI, а також поглибити розуміння можливостей діалогових моделей для розв'язання різних задач; отримати практичний досвід у використанні мовних моделей для створення чат-ботів, інтелектуальних асистентів, навчальних програм, а також систем, які адаптуються до різних сценаріїв і стилів комунікації.

**Код:**

import openai

# Завдання 1: Базовий діалог про замовлення корму для декоративних тварин

def pet\_food\_order\_dialog(api\_key):

openai.api\_key = api\_key

dialog\_context = [{"role": "system",

"content": "Ви помічник, спеціалізуєтесь на замовленнях корму для декоративних тварин. Допомагайте користувачам із вибором, замовленням та пов'язаними питаннями."}]

while True:

try:

user\_input = input("Користувач: ")

if user\_input.lower() == "вихід":

print("Завершення чату.")

break

dialog\_context.append({"role": "user", "content": user\_input})

response = openai.ChatCompletion.create(

model="gpt-3.5-turbo",

messages=dialog\_context

)

assistant\_message = response['choices'][0]['message']['content']

print("Помічник:", assistant\_message)

dialog\_context.append({"role": "assistant", "content": assistant\_message})

except KeyboardInterrupt:

print("\nСеанс чату було перервано. Завершення програми.")

break

except Exception as e:

print("Сервіс тимчасово недоступний:", e)

# Завдання 2: Чат із обмеженим контекстом для замовлення корму

def limited\_context\_pet\_food\_chat(api\_key):

openai.api\_key = api\_key

dialog\_context = [{"role": "system",

"content": "Ви помічник із замовлення корму для декоративних тварин із обмеженою пам'яттю. Тримайте контекст коротким і зосередженим на останніх запитах."}]

while True:

try:

user\_input = input("Користувач: ")

if user\_input.lower() == "вихід":

print("Завершення чату.")

break

dialog\_context.append({"role": "user", "content": user\_input})

if len(dialog\_context) > 6:

dialog\_context = dialog\_context[-6:]

response = openai.ChatCompletion.create(

model="gpt-3.5-turbo",

messages=dialog\_context

)

assistant\_message = response['choices'][0]['message']['content']

print("Помічник:", assistant\_message)

dialog\_context.append({"role": "assistant", "content": assistant\_message})

except KeyboardInterrupt:

print("\nСеанс чату було перервано. Завершення програми.")

break

except Exception as e:

print("Сервіс тимчасово недоступний:", e)

# Завдання 3: Чат для підтримки замовлень корму

def pet\_food\_order\_support\_chat(api\_key):

openai.api\_key = api\_key

dialog\_context = [{"role": "system",

"content": "Ви помічник підтримки, який допомагає із замовленнями корму для декоративних тварин. Надавайте інформацію про статус замовлення, ціни та доступність товарів."}]

while True:

try:

user\_input = input("Користувач: ")

if user\_input.lower() == "вихід":

print("Завершення чату.")

break

dialog\_context.append({"role": "user", "content": user\_input})

if len(dialog\_context) > 10:

dialog\_context = dialog\_context[-10:]

response = openai.ChatCompletion.create(

model="gpt-3.5-turbo",

messages=dialog\_context

)

assistant\_message = response['choices'][0]['message']['content']

print("Помічник:", assistant\_message)

dialog\_context.append({"role": "assistant", "content": assistant\_message})

except KeyboardInterrupt:

print("\nСеанс чату було перервано. Завершення програми.")

break

except Exception as e:

print("Сервіс тимчасово недоступний:", e)

# Завдання 4: Чат із настроюваним тоном для замовлень корму

def tone\_adjustable\_pet\_food\_chat(api\_key):

openai.api\_key = api\_key

tone = "професійний"

dialog\_context = [{"role": "system", "content": f"Відповідайте в {tone} тоні на запити щодо замовлення корму для декоративних тварин."}]

while True:

try:

user\_input = input("Користувач: ")

if user\_input.startswith("/змінити\_тон"):

tone = user\_input.split()[1]

dialog\_context = [{"role": "system", "content": f"Відповідайте в {tone} тоні на запити щодо замовлення корму для декоративних тварин."}]

print(f"Тон змінено на {tone}.")

continue

elif user\_input.lower() == "вихід":

print("Завершення чату.")

break

dialog\_context.append({"role": "user", "content": user\_input})

response = openai.ChatCompletion.create(

model="gpt-3.5-turbo",

messages=dialog\_context

)

assistant\_message = response['choices'][0]['message']['content']

print("Помічник:", assistant\_message)

dialog\_context.append({"role": "assistant", "content": assistant\_message})

except KeyboardInterrupt:

print("\nСеанс чату було перервано. Завершення програми.")

break

except Exception as e:

print("Сервіс тимчасово недоступний:", e)

# Завдання 7: Генерація ідей для корму

def generate\_pet\_food\_ideas(api\_key, prompt):

openai.api\_key = api\_key

try:

response = openai.ChatCompletion.create(

model="gpt-3.5-turbo",

messages=[{"role": "system", "content": "Ви творчий помічник із замовлення корму для декоративних тварин."},

{"role": "user", "content": prompt}]

)

print("Ідея для корму:", response['choices'][0]['message']['content'])

except Exception as e:

print("Помилка під час генерації ідей для корму:", e)

# Приклад запуску основних функцій

if \_\_name\_\_ == "\_\_main\_\_":

api\_key = "свій API ключ OpenAI СЮДИ"

print("Ласкаво просимо до помічника із замовлення корму для декоративних тварин!")

print("Доступні команди: ")

print("1. pet\_food\_order\_dialog(api\_key) - Базовий діалог.")

print("2. limited\_context\_pet\_food\_chat(api\_key) - Чат із обмеженим контекстом.")

print("3. pet\_food\_order\_support\_chat(api\_key) - Підтримка замовлень.")

print("4. tone\_adjustable\_pet\_food\_chat(api\_key) - Чат із налаштуванням тону.")

print("5. generate\_pet\_food\_ideas(api\_key, 'Запит') - Генерація ідей для корму.")

pet\_food\_order\_dialog(api\_key)

**Робота коду:**  
(**>Скріни сюди<**)

**Висновки**

У результаті виконання мети, було здобуто цінний досвід у створенні інтерактивних діалогів за допомогою моделі ChatGPT 3.5 на Python, що дозволило глибше зрозуміти принципи роботи з API OpenAI. Практичне використання API допомогло розвинути навички налаштування та інтеграції мовних моделей у різноманітні програми, зокрема чат-боти та інтелектуальні асистенти. Опанування цих інструментів сприяло розширенню знань щодо застосування діалогових моделей для вирішення широкого спектра завдань, таких як автоматизація підтримки клієнтів, створення персоналізованих навчальних програм і розробка систем, здатних адаптуватися до конкретних умов та стилів спілкування. Цей процес дозволив не тільки зрозуміти технічні аспекти роботи з мовними моделями, але й розвинути здатність застосовувати їх у реальних проектах, що розширює можливості автоматизації і покращення взаємодії з користувачами в різних сферах діяльності.