

Review del codice di un assistente virtuale

Introduzione

Nell'esercitazione di oggi ci è stato dato un codice e dobbiamo:

1. Capire cosa fa il programma senza eseguirlo.
2. Individuare nel codice sorgente le casistiche non standard che il programma non gestisce.
3. Individuare eventuali errori di sintassi / logici.
4. Proporre una soluzione per ognuno di essi.

Il codice è il seguente:

```
import datetime
```

```
def assistente_virtuale(comando):
```

```
    if comando == "Qual è la data di oggi?":
```

```
        oggi = datetime.datetime.today()
```

```
        risposta = "La data di oggi è " + oggi.strftime("%d/%m/%Y")
```

```
    elif comando == "Che ore sono?":
```

```
        ora_attuale = datetime.datetime.now().time()
```

```
        risposta = "L'ora attuale è " + ora_attuale.strftime("%H:%M")
```

```
    elif comando == "Come ti chiami?":
```

```
        risposta = "Mi chiamo Assistente Virtuale"
```

```
    else:
```

```
        risposta = "Non ho capito la tua domanda."
```

```
    return risposta
```

```
while True
```

```
    comando_utente = input("Cosa vuoi sapere? ")
```

```
    if comando_utente.lower() == "esci":
```

```
        print("Arrivederci!")
```

```
        break
```

```
    else:
```

```
        print(assistente_virtuale(comando_utente))
```

Dopo una prima lettura del codice possiamo dedurre sia il programma di un Assistente Virtuale che risponde a comandi predefiniti, se uno di questi non è nel programma, risponde con un messaggio di errore.

Il programma è attualmente strutturato per rispondere a 3 comandi specifici:

1. “Qual è la data di oggi?”: Fornisce la data attuale mostrando **giorno/mese/anno**.
2. “Che ore sono?”: Fornisce l’ora attuale mostrando **ore e minuti**.
3. “Come ti chiami?” : Questo comando risponde con il nome dell’**assistente virtuale**.

Troviamo altre 2 funzionalità del programma:

1. Se l’input dato non corrisponde esattamente a uno dei comandi predefiniti (“**Qual è la data di oggi?**”, “**Che ore sono?**”, “**Come ti chiami?**”), il programma risponde con il messaggio: “**Non ho capito la tua domanda**”.
2. Il programma ha un ciclo **while True** che riceve comandi dall’utente fino a quando non viene utilizzato il comando “**esci**” che interrompe il ciclo e ci risponde con “Arrivederci”.

Problemi o errori nel codice:

Durante l'analisi del codice originale, sono emersi diversi problemi che compromettono il corretto funzionamento del programma. Questi si dividono in tre categorie principali: **errori di sintassi**, **limitazioni logiche** e **mancanza di flessibilità**.

1. Errori di Sintassi

datetime.datetime.today(): Il metodo **datetime.datetime.today()** non esiste nel modulo **datetime**. La riga **oggi = datetime.datetime.today()** genera un errore durante l'esecuzione. Questa funzione deve essere sostituita con **datetime.datetime.now()**, che restituisce l'ora e la data correnti.

Mancanza del Punto e Virgola in while True: La dichiarazione **while True** manca del punto e virgola (;) alla fine, causando un errore di sintassi. Senza questa correzione, il ciclo infinito non verrà mai avviato.

2. Limitazioni

Sensibilità alle Maiuscole: Il programma non gestisce correttamente le differenze tra lettere maiuscole e minuscole. Ad esempio, se l'utente inserisce "qual è la data di oggi?" (con la "q" minuscola), il programma non riconoscerà il comando. Questo problema può essere risolto convertendo tutti gli input in minuscolo utilizzando **comando.lower()**.

Riconoscimento Esatto delle Frasi: Il programma richiede che l'utente inserisca esattamente le frasi predefinite ("Qual è la data di oggi?", "Che ore sono?", ecc.). Questa limitazione rende l'assistente poco intuitivo per l'utente.

Input Non Gestiti: Il programma risponde solo a tre comandi specifici e non offre alcuna flessibilità per riconoscere parole chiave o contesti alternativi. Ad esempio, una domanda come "Quale giorno è oggi?" non viene riconosciuta, anche se ha lo stesso intento di "Qual è la data di oggi?".

Miglioramenti:

Correggendo semplicemente questi 2 errori il codice dovrebbe funzionare, anche se consiglio di implementare altro, ad esempio:

1. Aggiungendo **comando = comando.lower()** facciamo in modo che il programma non abbia problemi a riconoscere le parole se scritte con uso incauto riguardo le maiuscole.
2. Modificando **if** e **elif** possiamo fare in modo che il programma utilizzi parole chiave e non frasi complete. Questo riduce di molto i problemi riguardanti un errore di digitazione dell'utente.
Le stringhe di comando sono state scritte così:

if "giorno" in comando or "oggi" in comando:

elif "ora" in comando or "ore" in comando:

elif "chiami" in comando or "nome" in comando:

Qui abbiamo aggiunto 2 parole chiave per comando, facendo in modo che il programma accetti input differenti. Alla domanda **"che ore sono?"** Il programma risponde con: **"L'ora attuale è 13:30"**, in quanto la parola **"ore"** è dentro la frase inserita dall'utente.

3. Per semplicità e una migliore leggibilità del codice modifichiamo la prima riga di codice in:

from datetime import datetime

Fa in modo che il programma importi le classi direttamente dal modulo **datetime**, utilizzando direttamente **datetime.now()** senza dover scrivere **datetime.datetime.now()**, semplificando le righe del codice.

Codice Corretto

Una volta effettuate tutte le modifiche andiamo a vedere il codice con le modifiche.

```
1 from datetime import datetime
2
3 def assistente_virtuale(comando):
4     comando = comando.lower()
5
6     if "giorno" in comando or "oggi" in comando:
7         oggi = datetime.now()
8         risposta = "La data di oggi è " + oggi.strftime("%d/%m/%Y")
9
10    elif "ora" in comando or "ore" in comando:
11        ora_attuale = datetime.now().time()
12        risposta = "L'ora attuale è " + ora_attuale.strftime("%H:%M")
13
14    elif "chiami" in comando or "nome" in comando:
15        risposta = "Mi chiamo Assistente Virtuale"
16
17    else:
18        risposta = "Non ho capito la tua domanda."
19
20    return risposta
21
22 while True:
23
24     comando_utente = input("Cosa vuoi sapere? ")
25
26     if comando_utente.lower().strip() == "esci":
27         print("Arrivederci!")
28         break
29
30     else:
31         print(assistente_virtuale(comando_utente))
```

Test e risultati

Abbiamo testato il programma inserendo diversi input, sia validi che non validi, per valutare la gestione delle risposte.

Input Inserito	Output Atteso	Esito
<i>Che giorno è oggi?</i>	<i>La data di oggi è 07/02/2025</i>	<i>Positivo</i>
<i>Che ore sono?</i>	<i>L'ora attuale è 14:30</i>	<i>Positivo</i>
<i>Come ti chiami?</i>	<i>Mi chiamo Assistente Virtuale</i>	<i>Positivo</i>
<i>Ora attuale</i>	<i>L'ora attuale è 14:30</i>	<i>Positivo</i>
<i>Ciao, che giorno è oggi?</i>	<i>La data di oggi è 07/02/2025</i>	<i>Positivo</i>
<i>Esci</i>	<i><u>Arrivederci!</u> <u>e chiusura del programma</u></i>	<i>Positivo</i>
<i>Ciao, che fai?</i>	<i>Non ho capito la tua domanda.</i>	<i>Positivo</i>

1. Il programma riconosce parole chiave e non solo le frasi esatte.
2. Il programma si interrompe quando viene inserito il comando “**esci**”.
3. Gli input non validi o inaspettati vengono gestiti correttamente.

Conclusioni

In questa esercitazione, abbiamo analizzato in modo approfondito il codice fornito per un assistente virtuale, identificando miglioramenti sia funzionali che strutturali. Ecco un riassunto dei principali risultati: Abbiamo compreso che il programma funge da semplice assistente virtuale che risponde a comandi predefiniti. Gestisce tre comandi specifici: fornire la data corrente, l'ora attuale e identificarsi con il proprio nome. Inoltre, un comando di "uscita" ("esci") consente all'utente di terminare il programma.

Durante l'analisi sono emersi alcuni problemi:

- **Errori di sintassi :**
 - **datetime.datetime.today()** era usato in modo errato invece di **datetime.datetime.now()**.
 - La dichiarazione **while True** mancava del punto e virgola (:), causando un errore di sintassi.
- **Sensibilità alle maiuscole :**
 - Il programma non gestiva le variazioni di capitalizzazione, rendendolo meno intuitivo per l'utente.
- **Gestione limitata degli input :**
 - Il programma riconosceva solo frasi esatte, fallendo nel riconoscere input simili o parafrasati.

Per risolvere questi problemi, abbiamo proposto le seguenti soluzioni:

- Corretto gli errori di sintassi per garantire un funzionamento corretto.
- Migliorato la gestione delle maiuscole convertendo tutti gli input in minuscolo tramite **comando.lower()**.
- Semplificato la struttura del codice importando direttamente **datetime** con **from datetime import datetime**, riducendo la ridondanza.

In conclusione, il codice rivisto risolve efficacemente i problemi iniziali, risultando in un assistente virtuale più affidabile e facile da usare.