

Relazione sul Codice Python: Analisi e Correzione di un Assistente Virtuale

INDICE

1. Funzione del Programma

2. Caratteristiche Non Standard

3. Errori identificati

4. Soluzioni Proposte

5. Conclusioni

1. Funzione del Programma

Il programma in esame simula un assistente virtuale in grado di rispondere a domande specifiche, come la data corrente, l'ora attuale e il nome dell'assistente. L'interazione con l'utente avviene tramite input da tastiera, e il programma si interrompe quando l'utente digita "esci". L'obiettivo principale dell'attività è sviluppare una capacità critica nell'analizzare e correggere il codice, identificando errori di sintassi, gestendo casistiche non standard e migliorando la robustezza del programma.

2. Caratteristiche Non Standard

Il codice presenta alcune casistiche non standard che richiedono attenzione:

- **Spazio vuoto come input:** Se l'utente inserisce uno spazio vuoto, il programma dovrebbe rispondere con un messaggio specifico, ad esempio "Per favore, cambia comando". Attualmente, il codice non gestisce questa situazione, il che potrebbe portare a comportamenti indesiderati.
- **Gestione delle eccezioni:** Il programma non è robusto rispetto a input non riconosciuti. Se l'utente inserisce un comando non valido, il programma potrebbe crashare. Invece, dovrebbe rispondere con un messaggio come "Si è verificato un errore" e continuare a funzionare.
- **Lista di domande e risposte limitata:** Il codice attuale gestisce solo poche domande e risposte. Una soluzione migliore sarebbe utilizzare una lista o un dizionario per memorizzare coppie di domande e risposte, rendendo il programma più flessibile e scalabile.

3. Errori identificati

Durante l'analisi del codice, sono stati identificati i seguenti errori:

- **Mancanza dei due punti nel comando** : La sintassi del ciclo è incompleta perché mancano i due punti () alla fine della condizione. Questo è un errore di sintassi che impedisce l'esecuzione del programma.
- **Comando errato**: Il codice contiene un comando non valido, , che dovrebbe essere sostituito con per ottenere la data corrente. Questo è un altro errore di sintassi che causa un crash del programma.

```
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
```

3

```
while True

    comando_utente = input("Cosa vuoi sapere? ")

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

        print("Arrivederci!")

        break

    else:

        print(assistente_virtuale(comando_utente))
```

4

4. Soluzioni Proposte

Per risolvere i problemi identificati e migliorare il programma, vengono proposte le seguenti soluzioni:

Gestione degli spazi vuoti: Aggiungere un controllo tramite un'istruzione per verificare se l'input dell'utente è uno spazio vuoto. In tal caso, il programma dovrebbe rispondere con un messaggio appropriato, ad esempio:

```
python
```

Copy

```
if not user_input.strip():
```

```
    print("Per favore, cambia comando.")
```

Gestione delle eccezioni: Utilizzare un blocco `try` per catturare eventuali errori e prevenire il crash del programma. Ad esempio:

```
python
```

Copy

```
try:
```

```
    # Codice che potrebbe generare un'eccezione
```

```
except Exception as e:
```

```
    print("Si è verificato un errore.")
```

Utilizzo di una lista o dizionario: Per rendere il programma più flessibile, è possibile utilizzare una lista o un dizionario per memorizzare le domande e le risposte. Ad esempio:

```
python
```

Copy

```
domande_risposte = {  
    "data": "Oggi è " + datetime.date.today().strftime("%d/%m/%Y"),  
    "ora": "Sono le " + datetime.datetime.now().strftime("%H:%M"),  
    "nome": "Il mio nome è Assistente Virtuale."  
}
```

Correzione degli errori di sintassi: Aggiungere i due punti mancanti nel ciclo `while` e correggere il comando `input` in `input()`.

5. Conclusioni

Il codice analizzato rappresenta un buon punto di partenza per un assistente virtuale, ma presenta alcune lacune che ne limitano l'efficacia. Gli errori di sintassi, come la mancanza dei due punti nel ciclo `while` e l'uso errato del comando `break`, devono essere corretti per garantire il corretto funzionamento del programma. Inoltre, la gestione di input non validi, come spazi vuoti o comandi sconosciuti, richiede l'implementazione di controlli aggiuntivi e l'uso di blocchi `try-catch` per prevenire crash imprevisti. Infine, l'uso di una lista o di un dizionario per memorizzare le domande e le risposte renderebbe il programma più flessibile e scalabile, permettendo di aggiungere facilmente nuove funzionalità in futuro. Con queste modifiche, il programma diventerebbe più robusto e adatto a un utilizzo pratico.