

REPORT PROGETTO S2/L5 ANDREA CORBELLINI

1. Capire cosa fa il programma senza eseguirlo.

Il programma implementa un assistente virtuale basato su input testuale. L'utente può interagire con il sistema ponendo domande specifiche, il programma risponde con informazioni sulla data, l'ora o il nome dell'assistente. Se il comando non è riconosciuto, il sistema restituisce un messaggio di errore. Il programma esegue un loop continuo finché l'utente non digita "esci".

2. Individuare nel codice sorgente le casistiche non standard che il programma non gestisce (esempio, comportamenti potenziali che non sono stati contemplati).

2.1 Errori di sintassi

1. Errore nella funzione `assistente_virtuale`:

- Il metodo `datetime.datetime()` non esiste. Il corretto metodo da usare è `datetime.date.today()`.

2. Errore nella sintassi del costrutto `while True`:

- Manca il carattere `:` alla fine della riga `while True`, che causa un errore di sintassi.

3. Errore nell'operatore di formattazione dell'ora:

- Il carattere `%` nella formattazione dell'ora è errato. Il corretto formato è `ora_attuale.strftime("%H:%M")`.

4. Indentazione errata nel return della funzione `assistente_virtuale`:

- Il `return risposta` è indentato in modo scorretto e dovrebbe essere allineato agli altri rami della funzione.

2.2 Comportamenti non gestiti

1. Gestione degli input non standard:

- Il programma non verifica se l'input è vuoto.
- Non è prevista una gestione per input con errori di battitura o sinonimi.

2. Gestione della data e dell'ora:

- Il programma non gestisce possibili eccezioni legate all'acquisizione della data e dell'ora, anche se poco probabili.

3. Individuare eventuali errori di sintassi / logici.

3.1 Correzione degli errori di sintassi

- Modificare `datetime.datetime.today()` con `datetime.date.today()`.
- Aggiungere `:` alla fine della riga `while True:`.
- Correggere la stringa di formattazione dell'ora con `strftime("%H:%M")`.
- Correggere l'indentazione del `return risposta`.

3.2 Miglioramenti nella gestione degli input

- Aggiungere un controllo per input vuoti.
- Normalizzare l'input rimuovendo spazi superflui e gestendo maiuscole/minuscole.
- Utilizzare una corrispondenza fuzzy per gestire piccoli errori di battitura.

3.3 Codice Corretto

```
28 import datetime
29
30 1 usage
31 def assistente_virtuale(comando):
32     if comando == "Qual è la data di oggi?":
33         oggi = datetime.date.today()
34         risposta = "La data di oggi è " + oggi.strftime("%d/%m/%Y")
35     elif comando == "Che ore sono?":
36         ora_attuale = datetime.datetime.now()
37         risposta = "L'ora attuale è " + ora_attuale.strftime("%H:%M")
38     elif comando == "Come ti chiami?":
39         risposta = "Mi chiamo Assistente Virtuale"
40     else:
41         risposta = "Non ho capito la tua domanda."
42     return risposta
43
44 while True:
45     comando_utente = input("Cosa vuoi sapere? ")
46     if comando_utente.lower() == "esci":
47         print("Arrivederci!")
48         break
49     else:
50         print(assistente_virtuale(comando_utente))
```

4. Conclusioni

Il codice originale conteneva errori sintattici e logici che impedivano il corretto funzionamento del programma. Con le correzioni apportate, il programma ora può gestire input più flessibili e garantire un funzionamento corretto.

