

Progetto S2/L5: Analisi di un codice in Python

Traccia:

Per agire come un Hacker bisogna capire come pensare fuori dagli schemi. L'esercizio di oggi ha lo scopo di allenare l'osservazione critica.

Dato il codice si richiede allo studente di:

1. Capire cosa fa il programma senza eseguirlo.
2. Individuare nel codice sorgente le casistiche non standard che il programma non gestisce (esempio, comportamenti potenziali che non sono stati contemplati).
3. Individuare eventuali errori di sintassi / logici.
4. Proporre una soluzione per ognuno di essi.

Codice:

```
1     import datetime
2     def assistente_virtuale(comando):
3         if comando == "Qual è la data di oggi?":
4             oggi = datetime.datetime.today()
5             risposta = "La data di oggi è" + oggi.strftime("%d/%m/%Y")
6         elif comando == "Che ore sono?":
7             ora_attuale = datetime.datetime.now().time()
8             risposta = "L'ora attuale è" + ora_attuale.strftime("%H:%M")
9         elif comando == "Come ti chiami":
10            risposta = "Mi chiamo Assistente Virtuale"
11        else:
12            risposta = "Non ho capito la tua domanda."
13        return risposta
14    while True
15        comando_utente = input("Cosa vuoi sapere?")
16        if comando_utente.lower() == "esci":
17            print("Arrivederci!")
18            break
19        else:
20            print(assistente_virtuale(comando_utente))
```

1: Descrizione del codice

Il codice sovrastante implementa un assistente virtuale di base. Permette all'utente di porre tre semplici domande, ma molto specifiche, e ottenere risposte in tempo reale. Nel programma principale è presente un ciclo infinito 'while True', che permette all'utente di usare il codice finché non desidera uscire. All'interno di questo ciclo, il programma richiede un input 'comando_utente'. Questo input viene passato alla funzione 'assistente_virtuale' quando questa viene chiamata. In base all'input la funzione genera una risposta, restituita tramite 'return risposta'. Infine, la risposta viene presentata a schermo con 'print()'.

2: Casistiche non standard non gestite

Nel codice presentato sono presenti diverse dimenticanze che potrebbero causare problemi.

- Case sensitivity di Python: i comandi sono scritti con un'ortografia specifica che potrebbe causare problemi se l'utente non scrive le domande esattamente come sono nel codice. Per esempio, se l'utente invece di scrivere "Qual è la data di oggi?" scrivesse "QUAL È LA DATA DI OGGI" il codice non riconoscerebbe il comando. Una soluzione potrebbe essere convertire in minuscolo l'input tramite '.lower()' definendo la stringa comando_minuscolo.
- Spazi tra le parole: se l'utente aggiunge spazi aggiuntivi tra le parole per errore, per esempio ("Qual è la data di oggi?") il programma non riconoscerebbe il comando. Una soluzione potrebbe essere eliminare gli spazi tra le parole dell'input con '.strip()' e definendo la stringa comando_ottimizzato.
- Punto interrogativo: se l'utente dovesse dimenticare o omettere il punto interrogativo alla fine del comando ("Qual è la data di oggi") il programma non riconoscerebbe il comando. La soluzione più semplice sarebbe quella di ricordare all'utente che il punto interrogativo è obbligatorio per porre la domanda al programma.

3: Errori di sintassi e logici

Nel codice sono presenti anche un paio di errori di sintassi e logici.

1. Riga 4: la funzione 'datetime.datetoday()' non è valida. Per ottenere la data si deve usare 'datetime.date.today()'
2. Riga 7: la funzione datetime.datetime.now().time() non è valida. Per ottenere l'ora si deve usare datetime.datetime.now()
3. Riga 14: mancano i ':' dopo while True

4: Codice corretto

Unendo le soluzioni alle casistiche non standard e le correzioni degli errori un codice funzionante e più efficiente potrebbe essere quello sottostante.

```
1      import datetime
2      def assistente_virtuale(comando):
3          comando_minuscolo = comando.lower()
4          comando_ottimizzato = comando_minuscolo.strip()
5          if comando == "qual è la data di oggi?":
6              oggi = datetime.date.today()
7              risposta = "La data di oggi è" + oggi.strftime("%d/%m/%Y")
8          elif comando == "che ore sono?":
9              ora_attuale = datetime.datetime.now()
10             risposta = "L'ora attuale è" + ora_attuale.strftime("%H:%M")
11         elif comando == "come ti chiami":
12             risposta = "Mi chiamo Assistente Virtuale"
13         else:
14             risposta = "Non ho capito la tua domanda."
15         return risposta
16     while True:
17         comando_utente = input("Cosa vuoi sapere? (Ricorda il punto interrogativo alla fine)")
18         if comando_utente.lower() == "esci":
19             print("Arrivederci!")
20             break
21         else:
22             print(assistente_virtuale(comando_utente))
```

Conclusioni

Correggendo solo gli errori logici e di sintassi il programma sarebbe funzionato ugualmente, ma analizzando correggendo anche le casistiche non standard si è aggiunta una migliore praticità nell'uso del codice.