

PROGETTO S2/L5

Analisi di un Programma scritto con Linguaggio Python

INDICE

1. Introduzione	Pagina 2
2. Presentazione del codice	Pagina 2
3. Analisi del codice	Pagina 3
4. Conclusioni	Pagina 5

1. Introduzione

Questo documento è un'analisi di un codice scritto con linguaggio Python. Questa attività viene chiamata "Code Review" ed è un processo molto importante nello sviluppo di programmi, consiste infatti in un esame approfondito di un codice da parte di un programmatore differente dall'autore.

2. Presentazione del codice

Il codice fornito è il seguente:

```
import datetime
def assistente_virtuale(comando):
    if comando == "Qual è la data di oggi?":
        oggi = datetime.date.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))
```

Il codice va a creare un programma in grado di rispondere a tre domande specifiche: la data odierna, l'ora attuale e il nome dell'assistente. L'obiettivo è fornire un'interazione di base con l'utente, permettendo di ottenere informazioni utili ed in modo immediato.

Questo assistente virtuale può essere utilizzato come un esempio di come gestire input dell'utente e fornire risposte in base a comandi predefiniti..

3. Analisi del Codice

Il Codice è così strutturato:

```
import datetime
```

Importa il modulo "datetime", necessario per ottenere la data e l'ora attuali. Per completezza prende i dati direttamente dal Sistema Operativo.

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

Questa funzione riceve una stringa (comando) e restituisce una risposta basata sul contenuto del comando.

Di seguito i comandi possibili e le relative risposte che il programma può dare

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

```
    oggi = datetime.date.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."
```

(Questa è una risposta nel caso in cui il comando non sia riconosciuto.)

```
while True
```

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

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

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

```
        break
```

```
    else:
```

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

while True: crea un ciclo che continua a ripetersi all'infinito, a meno che non venga interrotto con un'istruzione break (in questo caso "esci"), in questo caso verrà restituito il messaggio "Arrivederci!"

Ci sono tuttavia degli errori che rendono inutilizzabile il programma e sono:

- l'errato inserimento del metodo `datetime.datetime.today()` che dovrebbe essere `datetime.date.today()`
- l'errato inserimento del metodo `datetime.datetime.now.time()` dovrebbe essere `datetime.datetime.now()`
- assenza di `:` alla fine della riga del `while`
- assenza del comando `return`, in assenza del quale ad una domanda non prevista il programma si blocca.

Di seguito il codice corretto

```
import datetime
```

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

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

```
        oggi = datetime.date.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))
```

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

.

4. Conclusioni

Il programma è stato scritto in modo che dia risposte a domande ben precise e scritte con una determinata sintassi, infatti nel momento in cui l'utente immetta una domanda diversa da quelle programmate, il sistema darà come risposta "Non ho capito la tua domanda."

Per ovviare a questo piccolo dettaglio, che comunque non compromette la corretta esecuzione, è consigliabile inserire all'inizio un elenco delle richieste effettuabili, con la raccomandazione all'utente di copiare le domande proposte, o, più semplicemente dare la possibilità all'utente di selezionare la domanda tra quelle presenti in un apposito elenco.

Qualora si scelga questa ultima opzione il codice sarà

```
import datetime

print("Seleziona una delle seguenti opzioni:\n")
print("""1 - Qual è la data di oggi?
2 - Che ore sono?
3 - Come ti chiami?
4 - Esci\n""")

def assistente_virtuale(comando):
    if comando == "1":
        oggi = datetime.date.today()
        risposta = "La data di oggi è " + oggi.strftime("%d/%m/%Y")
    elif comando == "2":
        ora_attuale = datetime.datetime.now().time()
        risposta = "L'ora attuale è " + ora_attuale.strftime("%H:%M")
    elif comando == "3":
        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() == "4":
        print("Arrivederci!")
        break
    else:
        print(assistente_virtuale(comando_utente))
:
```