

Programmazione per Hacker – Python pt1

W6D4

Ho chiamato entrambi i programmi Epicode Geometry e incluso tutta la spiegazione dei passaggi nel codice come suggerito in call

```
Welcome  W6D4_Facoltativo.py  W6D4.py  X
home > kali > progetti > W6D4.py > ...
1  import math # Importa il modulo math per accedere alla costante pi greco necessaria per il calcolo della circonferenza del cerchio
2
3  # Definisce la funzione per calcolare il perimetro di un quadrato
4  def quadrato(lato):
5      # Il perimetro del quadrato si ottiene moltiplicando la lunghezza del lato per 4
6      return lato * 4
7
8  # Definisce la funzione per calcolare la circonferenza di un cerchio
9  def cerchio(raggio):
10     # La circonferenza si calcola con la formula 2 * pi * raggio, usando math.pi per il valore di pi greco
11     return 2 * math.pi * raggio
12
13 # Definisce la funzione per calcolare il perimetro di un rettangolo
14 def rettangolo(base, altezza):
15     # Il perimetro del rettangolo è dato dalla somma del doppio della base e del doppio dell'altezza
16     return (base * 2) + (altezza * 2)
17
18 # Definisce la funzione principale che gestisce l'interazione con l'utente
19 def main():
20     # Avvia un ciclo infinito per mostrare il menu finché l'utente non sceglie di uscire
21     while True:
22         # Stampa il menu delle opzioni disponibili
23         print("\nEpicode Geometry 📐 Calcola il perimetro delle tue figure geometriche!:)")
24         print("1. Quadrato") # Opzione per calcolare il perimetro del quadrato
25         print("2. Cerchio") # Opzione per calcolare la circonferenza del cerchio
26         print("3. Rettangolo") # Opzione per calcolare il perimetro del rettangolo
27         print("4. Esci") # Opzione per terminare il programma
28
29         # Richiede all'utente di inserire la scelta (1, 2, 3 o 4)
30         scelta = input("Scegli un'opzione (1-4): ")
31
32         # Gestisce la scelta del quadrato
33         if scelta == "1":
34             try:
35                 # Chiede all'utente di inserire la lunghezza del lato e la converte in un numero decimale (float)
36                 lato = float(input("Inserisci la lunghezza del lato del quadrato: "))
37                 # Verifica che il lato sia maggiore di zero per garantire un input valido
38                 if lato <= 0:
39                     print("Sbagliato!!! il lato deve essere maggiore di zero!")
40                     continue # Torna all'inizio del ciclo per mostrare nuovamente il menu
41                 # Calcola il perimetro chiamando la funzione quadrato
42                 perimetro = quadrato(lato)
43                 # Mostra il risultato con due cifre decimali
44                 print(f"il perimetro del quadrato è: {perimetro:.2f}")
45             except ValueError:
46                 # Gestisce l'errore nel caso l'utente inserisca un valore non numerico
47                 print("Sbagliato!!! inserisci un valore numerico valido!")
48
49         # Gestisce la scelta del cerchio
50         elif scelta == "2":
51             try:
52                 # Chiede all'utente di inserire il raggio e lo converte in un numero decimale (float)
53                 raggio = float(input("Inserisci il raggio del cerchio: "))
54                 # Verifica che il raggio sia maggiore di zero per garantire un input valido
55                 if raggio <= 0:
56                     print("Sbagliato!!! il raggio deve essere maggiore di zero!")
57                     continue # Torna all'inizio del ciclo per mostrare nuovamente il menu
58                 # Calcola la circonferenza chiamando la funzione cerchio
59                 circonferenza = cerchio(raggio)
60                 # Mostra il risultato con due cifre decimali
61                 print(f"La circonferenza del cerchio è: {circonferenza:.2f}")
62             except ValueError:
63                 # Gestisce l'errore nel caso l'utente inserisca un valore non numerico
64                 print("Sbagliato!!! inserisci un valore numerico valido!")
65
66         # Gestisce la scelta del rettangolo
67         elif scelta == "3":
68             try:
69                 # Chiede all'utente di inserire la base e l'altezza, convertendoli in numeri decimali (float)
70                 base = float(input("Inserisci la base del rettangolo: "))
71                 altezza = float(input("Inserisci l'altezza del rettangolo: "))
72                 # Verifica che base e altezza siano maggiori di zero per garantire input validi
73                 if base <= 0 or altezza <= 0:
74                     print("Sbagliato!!! base e altezza devono essere maggiori di zero!")
75                     continue # Torna all'inizio del ciclo per mostrare nuovamente il menu
76                 # Calcola il perimetro chiamando la funzione rettangolo
77                 perimetro = rettangolo(base, altezza)
78                 # Mostra il risultato con due cifre decimali
79                 print(f"il perimetro del rettangolo è: {perimetro:.2f}")
80             except ValueError:
81                 # Gestisce l'errore nel caso l'utente inserisca valori non numerici
82                 print("Sbagliato!!! inserisci valori numerici validi!")
83
84         # Gestisce la scelta di uscita
85         elif scelta == "4":
86             # Stampa un messaggio di conferma e termina il ciclo
87             print("Hai terminato il programma!")
88             break # Esce dal ciclo while, terminando il programma
89
90         # Gestisce scelte non valide
91         else:
92             # Se l'utente inserisce un'opzione non compresa tra 1 e 4, mostra un messaggio di errore
93             print("Sbagliato!!! scegli un'opzione valida (1-4)!")
94
95 # Verifica se il programma viene eseguito direttamente (non importato come modulo)
96 if __name__ == "__main__":
97     # Esegue la funzione principale per avviare il programma
98     main()
```

Test e descrizione

Ho svolto il test sul terminale per verificare che soddisfasse tutti i requisiti richiesti dall'esercizio, allegando il test di seguito dopo la traccia dell'esercizio, testando anche risposta sbagliata e comando chiusura programma:



Esercizio
Traccia

Traccia:

Si scriva un programma in Python che in base alla scelta dell'utente permetta di calcolare il perimetro di diverse figure geometriche (scegliete pure quelle che volete voi). Per la risoluzione dell'esercizio abbiamo scelto:

- Quadrato (perimetro = lato*4)
- Cerchio (circonferenza = 2*pi greco*r)
- Rettangolo (perimetro= base*2 + altezza*2)

```
(kali@kali)-[~/progetti]
$ python W6D4.py

Epicode Geometry - Calcola il perimetro delle tue figure geometriche!:
1. Quadrato
2. Cerchio
3. Rettangolo
4. Esci
Scegli un'opzione (1-4): 1
Inserisci la lunghezza del lato del quadrato: 4
Il perimetro del quadrato è: 16.00

Epicode Geometry - Calcola il perimetro delle tue figure geometriche!:
1. Quadrato
2. Cerchio
3. Rettangolo
4. Esci
Scegli un'opzione (1-4): 2
Inserisci il raggio del cerchio:
Sbagliato!!! inserisci un valore numerico valido!

Epicode Geometry - Calcola il perimetro delle tue figure geometriche!:
1. Quadrato
2. Cerchio
3. Rettangolo
4. Esci
Scegli un'opzione (1-4): 2
Inserisci il raggio del cerchio: 2
La circonferenza del cerchio è: 12.57

Epicode Geometry - Calcola il perimetro delle tue figure geometriche!:
1. Quadrato
2. Cerchio
3. Rettangolo
4. Esci
Scegli un'opzione (1-4): 3
Inserisci la base del rettangolo: 2
Inserisci l'altezza del rettangolo: 2
Il perimetro del rettangolo è: 8.00

Epicode Geometry - Calcola il perimetro delle tue figure geometriche!:
1. Quadrato
2. Cerchio
3. Rettangolo
4. Esci
Scegli un'opzione (1-4): 5
Sbagliato!!! scegli un'opzione valida (1-4)!

Epicode Geometry - Calcola il perimetro delle tue figure geometriche!:
1. Quadrato
2. Cerchio
3. Rettangolo
4. Esci
Scegli un'opzione (1-4): 4
Hai terminato il programma!
```

W6D4 Facoltativo

Ho incluso tutta la spiegazione dei passaggi nel codice come suggerito in call
(In Verde)

```
Welcome | W6D4.py | W6D4_Facoltativo.py X
home > kali > progetti > W6D4_Facoltativo.py > main
1 import math # Importa il modulo math per accedere alla costante pi greco necessaria per il calcolo della circonferenza e dell'area del cerchio
2
3 # Definisce la funzione per calcolare perimetro e area di un quadrato
4 def quadrato(valore):
5     # Il valore immesso rappresenta il lato del quadrato
6     # Perimetro = lato * 4
7     perimetro = valore * 4
8     # Area = lato * lato
9     area = valore * valore
10    return perimetro, area
11
12 # Definisce la funzione per calcolare circonferenza e area di un cerchio
13 def cerchio(valore):
14     # Il valore immesso rappresenta il raggio del cerchio
15     # Circonferenza = 2 * pi * raggio
16     circonferenza = 2 * math.pi * valore
17     # Area = pi * raggio * raggio
18     area = math.pi * valore * valore
19     return circonferenza, area
20
21 # Definisce la funzione per calcolare perimetro e area di un rettangolo
22 def rettangolo(base, altezza):
23     # Il valore immesso rappresenta la base, l'altezza è richiesta separatamente
24     # Perimetro = (base * 2) + (altezza * 2)
25     perimetro = (base * 2) + (altezza * 2)
26     # Area = base * altezza
27     area = base * altezza
28     return perimetro, area
29
30 # Funzione principale che gestisce l'interazione con l'utente
31 def main():
32     # Dizionario delle figure disponibili, con chiave come opzione e valore come nome
33     figure_disponibili = {
34         "1": "Quadrato",
35         "2": "Cerchio",
36         "3": "Rettangolo"
37     }
38
39     try:
40         # Richiede il valore iniziale solo la prima volta
41         valore = float(input("Epicode Geometry | Inserisci il valore iniziale (lato, raggio, base): "))
42         # Verifica che il valore sia positivo
43         if valore <= 0:
44             print("Sbagliato!!! il valore deve essere maggiore di zero!")
45             return # Termina il programma se il valore non è valido
46     except ValueError:
47         # Gestisce errori di input non numerico
48         print("Sbagliato!!! inserisci un valore numerico valido!")
49         return # Termina il programma
50
51 # Ciclo che continua finché ci sono figure disponibili
52 while figure_disponibili:
53     # Stampa il menu delle figure ancora disponibili
54     print("\nFigure geometriche disponibili:")
55     for key, figura in figure_disponibili.items():
56         print(f"{key}. {figura}")
57     print(f"{' ' * (len(figure_disponibili) + 1)}. Esci") # Opzione per uscire
58
59     # Richiede la scelta dell'utente
60     scelta = input(f"Scegli un'opzione (1-{len(figure_disponibili) + 1}): ")
61
62     # Gestisce la scelta del quadrato
63     if scelta == "1" and "1" in figure_disponibili:
64         try:
65             # Calcola perimetro e area usando il valore corrente come lato
66             perimetro, area = quadrato(valore)
67             # Mostra i risultati con due cifre decimali
68             print(f"Quadrato (lato = {valore:.2f}):")
69             print(f"  Perimetro: {perimetro:.2f}")
70             print(f"  Area: {area:.2f}")
71             # Aggiorna il valore per la prossima figura con l'area calcolata
72             valore = area
73             # Rimuove il quadrato dalle opzioni disponibili
74             del figure_disponibili["1"]
75         except Exception as e:
76             # Gestisce eventuali errori imprevisti
77             print(f"Sbagliato!!! {e}")
78
79     # Gestisce la scelta del cerchio
80     elif scelta == "2" and "2" in figure_disponibili:
81         try:
82             # Calcola circonferenza e area usando il valore corrente come raggio
83             circonferenza, area = cerchio(valore)
84             # Mostra i risultati con due cifre decimali
85             print(f"Cerchio (raggio = {valore:.2f}):")
86             print(f"  Circonferenza: {circonferenza:.2f}")
87             print(f"  Area: {area:.2f}")
88             # Aggiorna il valore per la prossima figura con l'area calcolata
89             valore = area
90             # Rimuove il cerchio dalle opzioni disponibili
91             del figure_disponibili["2"]
92         except Exception as e:
93             # Gestisce eventuali errori imprevisti
94             print(f"Sbagliato!!! {e}")
95
96     # Gestisce la scelta del rettangolo
97     elif scelta == "3" and "3" in figure_disponibili:
98         try:
99             # Usa il valore corrente come base
```

```

100     base = valore
101     # Richiede l'altezza all'utente
102     altezza = float(input("Inserisci l'altezza del rettangolo: "))
103     # Verifica che l'altezza sia positiva
104     if altezza <= 0:
105         print("Sbagliato!!! l'altezza deve essere maggiore di zero!")
106         continue # Torna al menu
107     # Calcola perimetro e area
108     perimetro, area = rettangolo(base, altezza)
109     # Mostra i risultati con due cifre decimali
110     print(f"Rettangolo (base = {base:.2f}, altezza = {altezza:.2f}):")
111     print(f"    Perimetro: {perimetro:.2f}")
112     print(f"    Area: {area:.2f}")
113     # Aggiorna il valore per la prossima figura con l'area calcolata
114     valore = area
115     # Rimuove il rettangolo dalle opzioni disponibili
116     del figure_disponibili["3"]
117 except ValueError:
118     # Gestisce errori di input non numerico per l'altezza
119     print("Sbagliato!!! inserisci un valore numerico valido per l'altezza!")
120 except Exception as e:
121     # Gestisce eventuali errori imprevisti
122     print(f"Sbagliato!!! {e}")
123
124 # Gestisce la scelta di uscita
125 elif scelta == str(len(figure_disponibili) + 1):
126     # Termina il programma
127     print("Programma terminato.")
128     break
129
130 # Gestisce scelte non valide
131 else:
132     # Messaggio di errore per opzioni non disponibili
133     print(f"Sbagliato!!! scegli un'opzione valida (1-{len(figure_disponibili) + 1})!")
134
135 # Se non ci sono più figure disponibili, termina il programma
136 if not figure_disponibili:
137     print("\nHai terminato il programma e hai calcolato tutte le figure.")
138
139 # Verifica se il programma viene eseguito direttamente
140 if __name__ == "__main__":
141     # Esegue la funzione principale
142     main()

```

Test e descrizione

Ho svolto il test sul terminale per verificare che soddisfasse tutti i requisiti richiesti dall'esercizio, allegando il test di seguito dopo la traccia dell'esercizio.



Esercizio
Traccia

Facoltativo:

Si adatti il precedente esercizio in modo che acquisisca da tastiera il valore immesso dall'utente, calcoli il perimetro e l'area di una figura geometrica scelta dall'utente, e utilizzi automaticamente il valore dell'area come valore per calcolare il perimetro e l'area della prossima figura geometrica scelta nuovamente dall'utente.

- Creare dunque una selezione multipla di figure da proporre all'utente ad ogni nuovo calcolo.
- Il valore iniziale viene immesso dall'utente solo la prima volta allo start del software.
- Ogni volta che l'utente seleziona una figura, questa viene tolta dalle prossime opzioni presentate.

Ripetere il procedimento per almeno 3 figure geometriche (es. quadrato, rettangolo, cerchio).

Es.

1° figura: **Quadrato** → lato = 3; perimetro = 12; area = 9.

2° figura: **Rettangolo** → lato = 9; perimetro = 27 (9*2 + 4,5*2); area = 40,5 (9*4,5).

ecc..

```

(kali@kali)-[~/progetti]
$ python3 W6D4_Facoltativo.py

Epicode Geometry - Inserisci il valore iniziale (lato, raggio, base): 3

Figure geometriche disponibili:
1. Quadrato
2. Cerchio
3. Rettangolo
4. Esci
Scegli un'opzione (1-4): 1
Quadrato (lato = 3.00):
  Perimetro: 12.00
  Area: 9.00

Figure geometriche disponibili:
2. Cerchio
3. Rettangolo
3. Esci
Scegli un'opzione (1-3): 2
Cerchio (raggio = 9.00):
  Circonferenza: 56.55
  Area: 254.47

Figure geometriche disponibili:
3. Rettangolo
2. Esci
Scegli un'opzione (1-2): 3
Inserisci l'altezza del rettangolo: 3
Rettangolo (base = 254.47, altezza = 3.00):
  Perimetro: 514.94
  Area: 763.41

Hai terminato il programma e hai calcolato tutte le figure.

```

Test programma con opzione sbalgiata e scelta chiusura programma.

```

Epicode Geometry - Inserisci il valore iniziale (lato, raggio, base): 5

Figure geometriche disponibili:
1. Quadrato
2. Cerchio
3. Rettangolo
4. Esci
Scegli un'opzione (1-4): 5
Sbagliato!!! scegli un'opzione valida (1-4)!

Figure geometriche disponibili:
1. Quadrato
2. Cerchio
3. Rettangolo
4. Esci
Scegli un'opzione (1-4): 4
Programma terminato.

```