

Calcolatore di perimetri

Codice:

```
import math # Libreria con molte funzioni matematiche

def calcola_circonferenza_cerchio(raggio):
    circonferenza = 2 * math.pi * raggio
    return circonferenza

def calcola_perimetro_quadrato(lato):
    perimetro = 4 * lato
    return perimetro

def calcola_perimetro Rettangolo(base, altezza):
    perimetro = 2 * (base + altezza)
    return perimetro

def main():
    print("Calcolo del perimetro/circonferenza delle forme geometriche")
    print("Scegli una forma:")
    print("1. Cerchio")
    print("2. Quadrato")
    print("3. Rettangolo")
    scelta = input("Inserisci il numero corrispondente alla forma desiderata: ")

    # Calcolo della circonferenza del cerchio
    if scelta == "1":
        raggio = float(input("Inserisci il raggio del cerchio: ")) # Utilizziamo float per convertire i
        # numeri in decimali
        circonferenza = calcola_circonferenza_cerchio(raggio)
        print(f"La circonferenza del cerchio è: {circonferenza}") # Utilizzo le f-string (leggibilità
        # del codice)
        print("Spiegazione: La circonferenza di un cerchio si calcola moltiplicando il doppio di  $\pi$ 
        (pi greco) per il raggio del cerchio.")

    # Calcolo del perimetro del quadrato
    elif scelta == "2":
        lato = float(input("Inserisci la lunghezza di un lato del quadrato: ")) # Utilizziamo float
        # per convertire i numeri in decimali
        perimetro = calcola_perimetro_quadrato(lato)
        print(f"Il perimetro del quadrato è: {perimetro}") # Utilizzo le f-string (leggibilità del
        # codice)
        print("Spiegazione: Il perimetro di un quadrato si calcola moltiplicando la lunghezza di
        uno dei suoi lati per 4.")
```

```

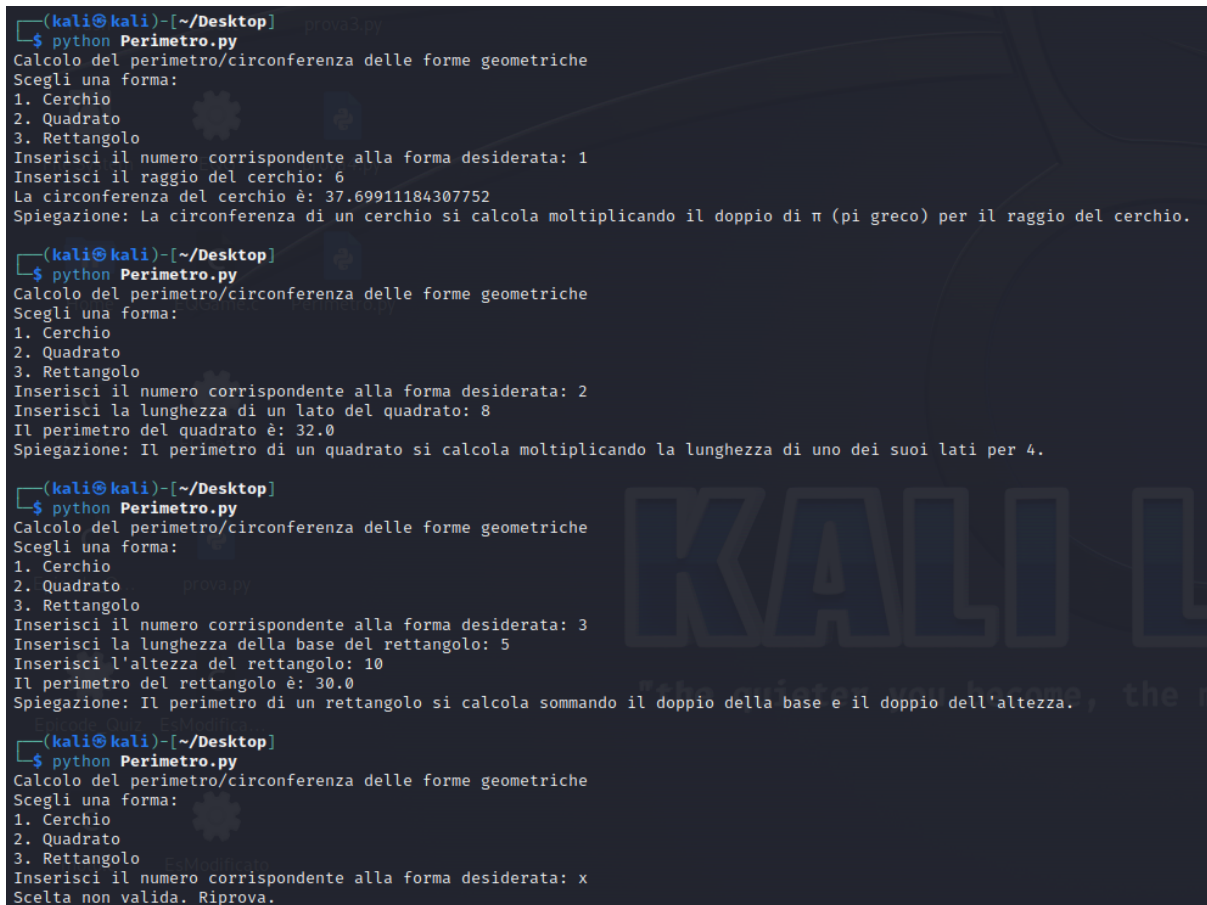
# Calcolo del perimetro del rettangolo
elif scelta == "3":
    base = float(input("Inserisci la lunghezza della base del rettangolo: ")) # Utilizziamo float
    # per convertire i numeri in decimali
    altezza = float(input("Inserisci l'altezza del rettangolo: "))
    perimetro = calcola_perimetro_rettangolo(base, altezza)
    print(f"Il perimetro del rettangolo è: {perimetro}") # Utilizzo le f-string (leggibilità del
    codice)
    print("Spiegazione: Il perimetro di un rettangolo si calcola sommando il doppio della
    base e il doppio dell'altezza.")

# Else, per scelte diverse da 1,2,3
else:
    print("Scelta non valida. Riprova.")

# Garantisce che "main" venga eseguito solo quando il programma viene eseguito
direttamente
if __name__ == "__main__":
    main()

```

Codice in azione:



```

(kali@kali)~/Desktop] python Perimetro.py
Calcolo del perimetro/circonferenza delle forme geometriche
Scegli una forma:
1. Cerchio
2. Quadrato
3. Rettangolo
Inserisci il numero corrispondente alla forma desiderata: 1
Inserisci il raggio del cerchio: 6
La circonferenza del cerchio è: 37.69911184307752
Spiegazione: La circonferenza di un cerchio si calcola moltiplicando il doppio di π (pi greco) per il raggio del cerchio.

(kali@kali)~/Desktop] python Perimetro.py
Calcolo del perimetro/circonferenza delle forme geometriche
Scegli una forma:
1. Cerchio
2. Quadrato
3. Rettangolo
Inserisci il numero corrispondente alla forma desiderata: 2
Inserisci la lunghezza di un lato del quadrato: 8
Il perimetro del quadrato è: 32.0
Spiegazione: Il perimetro di un quadrato si calcola moltiplicando la lunghezza di uno dei suoi lati per 4.

(kali@kali)~/Desktop] python Perimetro.py
Calcolo del perimetro/circonferenza delle forme geometriche
Scegli una forma:
1. Cerchio
2. Quadrato
3. Rettangolo
Inserisci il numero corrispondente alla forma desiderata: 3
Inserisci la lunghezza della base del rettangolo: 5
Inserisci l'altezza del rettangolo: 10
Il perimetro del rettangolo è: 30.0
Spiegazione: Il perimetro di un rettangolo si calcola sommando il doppio della base e il doppio dell'altezza.

(kali@kali)~/Desktop] python Perimetro.py
Calcolo del perimetro/circonferenza delle forme geometriche
Scegli una forma:
1. Cerchio
2. Quadrato
3. Rettangolo
Inserisci il numero corrispondente alla forma desiderata: x
Scelta non valida. Riprova.

```

