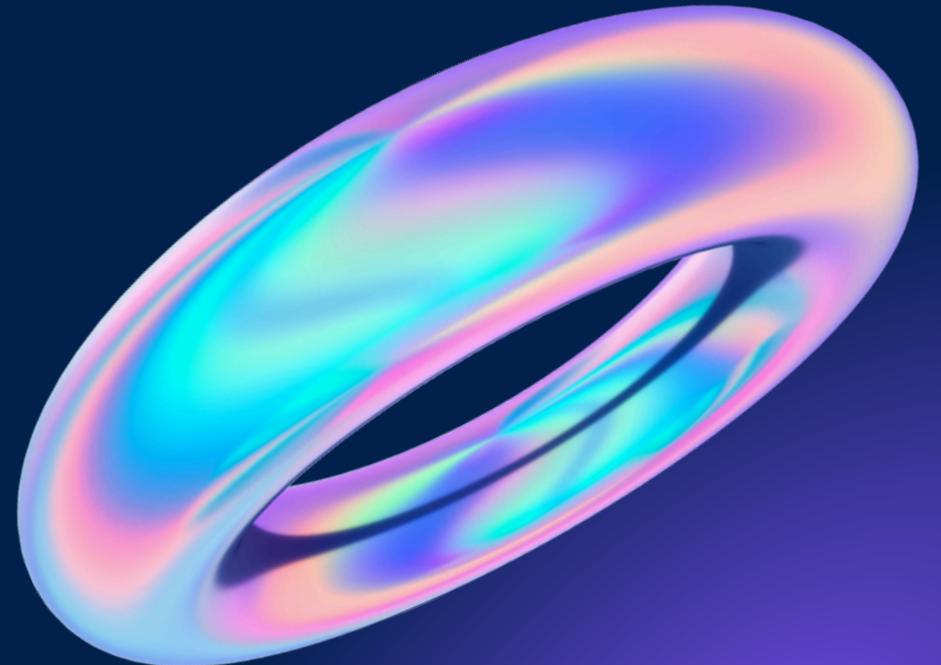




Esercizio S3L3

Presentazione



Traccia esercizio



**EPICODE**

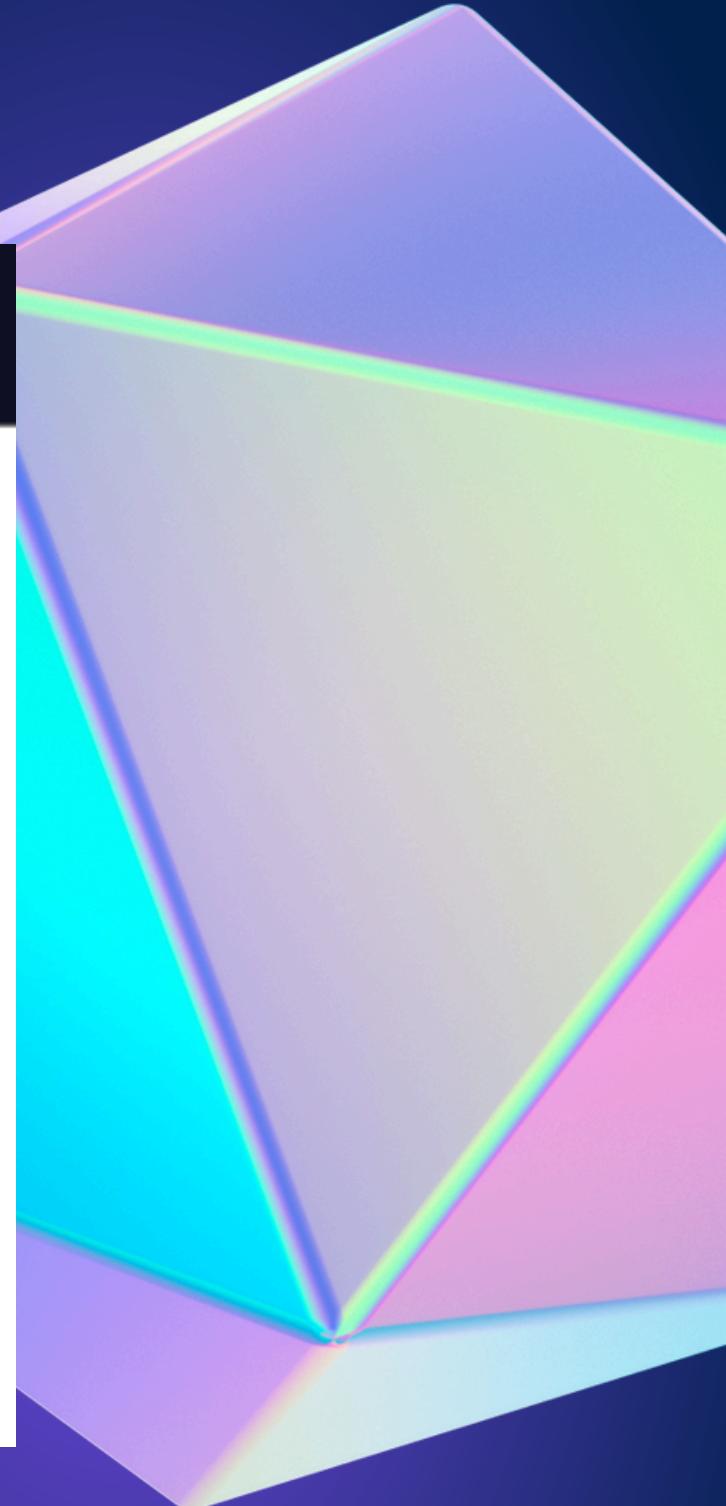
[Pratica S3/L3 PDF](#)

Esercizio
Python per Hacker

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 = $2\pi r$).
- Rettangolo (perimetro= base*2 + altezza*2).



Team

Gianpaolo Miliccia

Team leader

Luca Gaspari

Fabio Nobili

Antonio Perna

Andrè Vinícius

Samuel Sette

Romano Cascialli



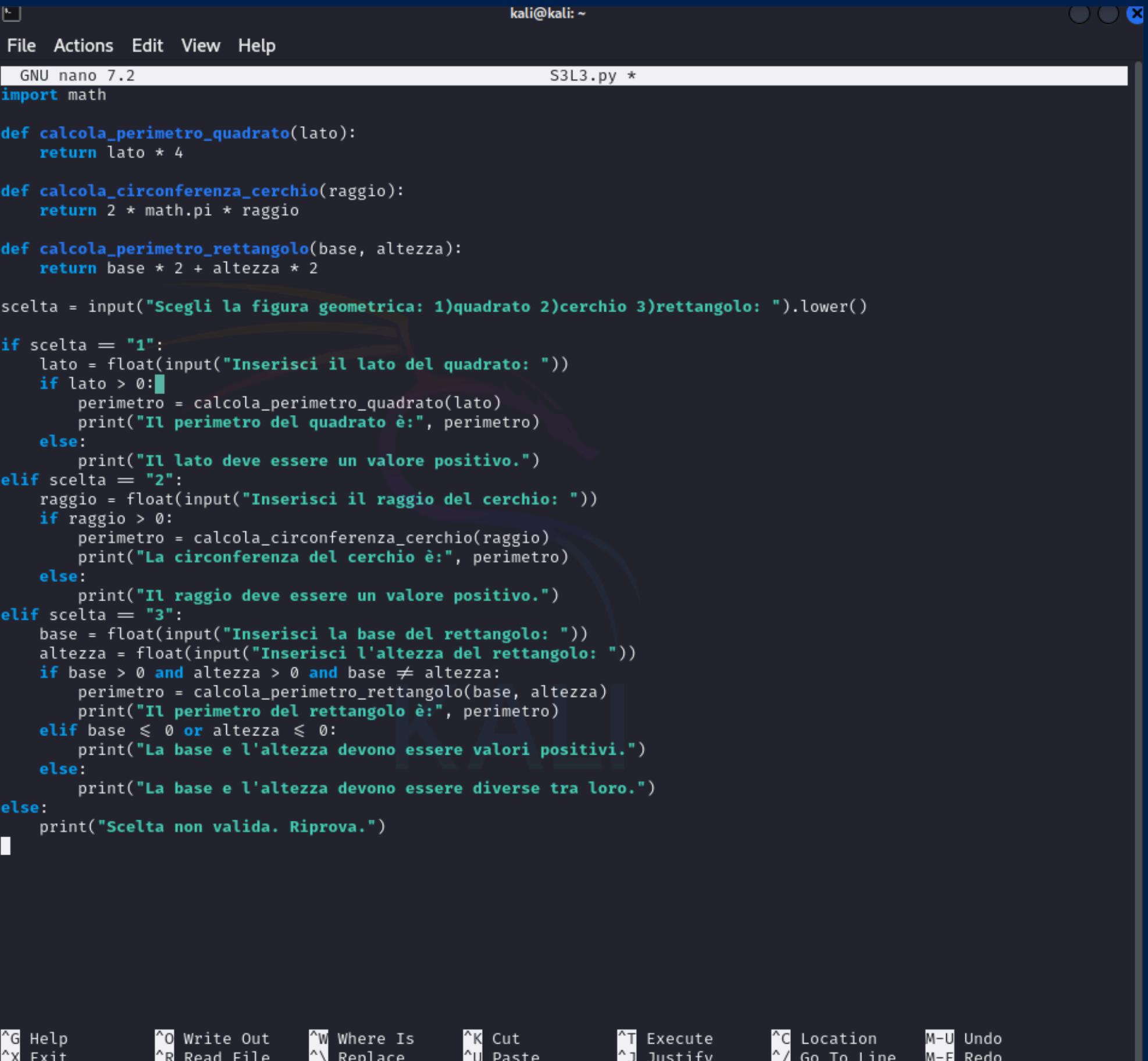
Cosa abbiamo fatto

Abbiamo scritto un programma in Python che in base alla scelta dell'utente permetta di calcolare il perimetro di diverse figure geometriche, abbiamo scelto:

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



Codice



The screenshot shows a terminal window titled "S3L3.py *". The window contains Python code for calculating the perimeter of different geometric shapes. The code uses functions to calculate the perimeter of a square, a circle, and a rectangle based on user input. It includes validation for positive values and distinct base and height for rectangles.

```
File Actions Edit View Help
GNU nano 7.2
import math

def calcola_perimetro_quadrato(lato):
    return lato * 4

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

def calcola_perimetro_rettangolo(base, altezza):
    return base * 2 + altezza * 2

scelta = input("Scegli la figura geometrica: 1)quadrato 2)cerchio 3)rettangolo: ").lower()

if scelta == "1":
    lato = float(input("Inserisci il lato del quadrato: "))
    if lato > 0:
        perimetro = calcola_perimetro_quadrato(lato)
        print("Il perimetro del quadrato è:", perimetro)
    else:
        print("Il lato deve essere un valore positivo.")
elif scelta == "2":
    raggio = float(input("Inserisci il raggio del cerchio: "))
    if raggio > 0:
        perimetro = calcola_circonferenza_cerchio(raggio)
        print("La circonferenza del cerchio è:", perimetro)
    else:
        print("Il raggio deve essere un valore positivo.")
elif scelta == "3":
    base = float(input("Inserisci la base del rettangolo: "))
    altezza = float(input("Inserisci l'altezza del rettangolo: "))
    if base > 0 and altezza > 0 and base != altezza:
        perimetro = calcola_perimetro_rettangolo(base, altezza)
        print("Il perimetro del rettangolo è:", perimetro)
    elif base <= 0 or altezza <= 0:
        print("La base e l'altezza devono essere valori positivi.")
    else:
        print("La base e l'altezza devono essere diverse tra loro.")
else:
    print("Scelta non valida. Riprova.")

^G Help      ^O Write Out      ^W Where Is      ^K Cut      ^T Execute      ^C Location      M-U Undo
^X Exit      ^R Read File      ^\ Replace      ^U Paste      ^J Justify      ^/ Go To Line      M-E Redo
```

Spiegazione

```
import math

def calcola_perimetro_quadrato(lato):
    return lato * 4

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

def calcola_perimetro_rettangolo(base, altezza):
    return base * 2 + altezza * 2
```

Per prima cosa abbiamo importato la libreria math per usare ‘math.pi’ per il calcolo della circonferenza del cerchio

All'inizio del codice sono presenti tre definizioni di funzioni (def) che servono a calcolare i perimetri di diverse forme geometriche. Ogni funzione è un blocco di codice riutilizzabile che svolge un compito specifico:

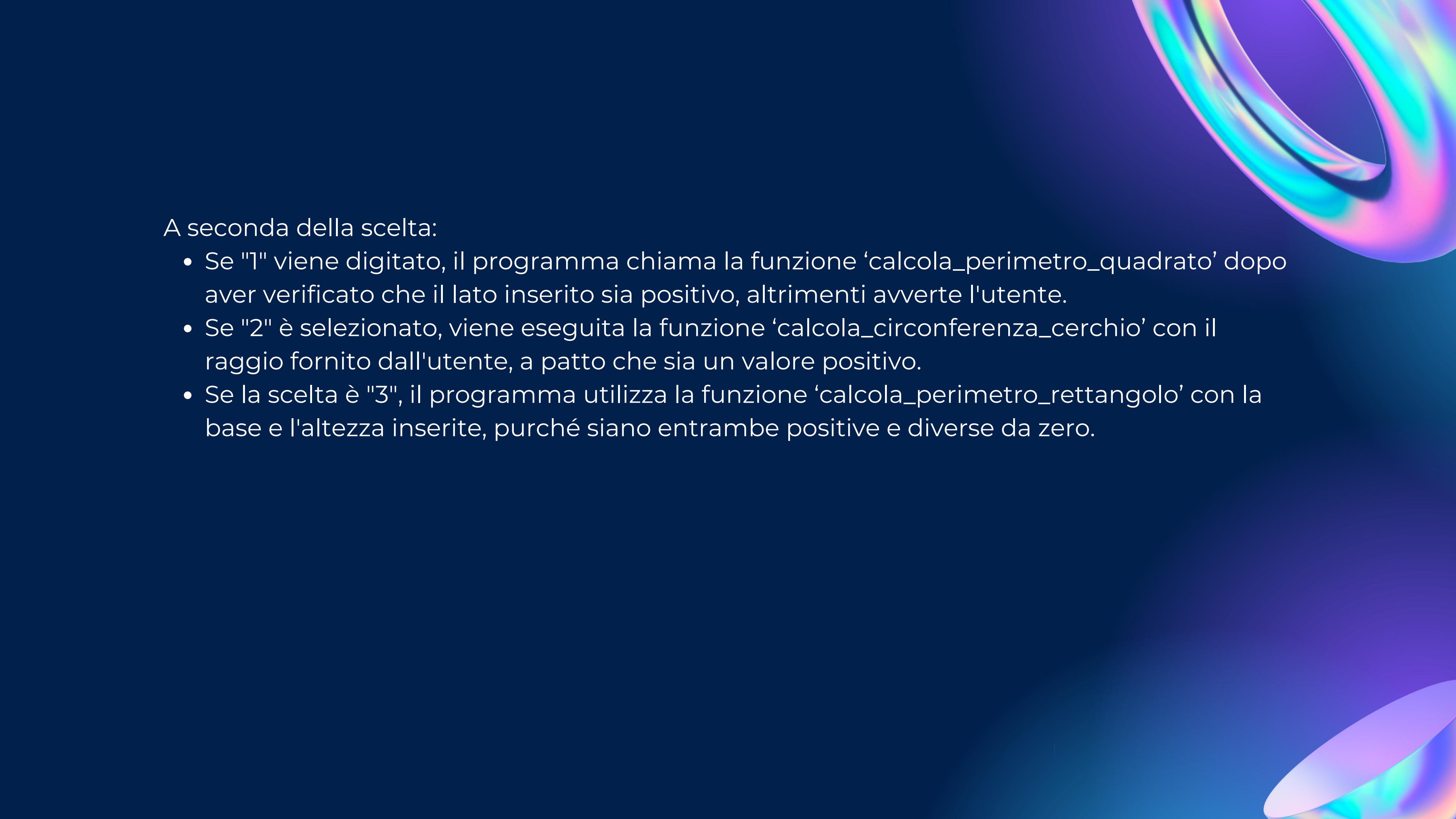
- Una calcola il perimetro di un quadrato, moltiplicando la lunghezza del lato per quattro.
- Un'altra determina la circonferenza di un cerchio, utilizzando il valore di pi greco e raddoppiando il prodotto del raggio per pi greco.
- La terza funzione calcola il perimetro di un rettangolo, sommando il doppio della base con il doppio dell'altezza.

Queste funzioni sono progettate per essere chiamate altrove nel programma, dove sono necessari questi specifici calcoli geometrici, rendendo il codice più chiaro e facile da gestire.

Nella seconda parte del codice, l'utente interagisce con il programma attraverso una serie di input per calcolare il perimetro di una figura geometrica scelta. Viene richiesto di inserire un valore corrispondente alla figura desiderata: "1" per il quadrato, "2" per il cerchio e "3" per il rettangolo.

```
scelta = input("Scegli la figura geometrica: 1)quadrato 2)cerchio 3)rettangolo: ").lower()

if scelta == "1":
    lato = float(input("Inserisci il lato del quadrato: "))
    if lato > 0:
        perimetro = calcola_perimetro_quadrato(lato)
        print("Il perimetro del quadrato è:", perimetro)
    else:
        print("Il lato deve essere un valore positivo.")
elif scelta == "2":
    raggio = float(input("Inserisci il raggio del cerchio: "))
    if raggio > 0:
        perimetro = calcola_circonferenza_cerchio(raggio)
        print("La circonferenza del cerchio è:", perimetro)
    else:
        print("Il raggio deve essere un valore positivo.")
elif scelta == "3":
    base = float(input("Inserisci la base del rettangolo: "))
    altezza = float(input("Inserisci l'altezza del rettangolo: "))
    if base > 0 and altezza > 0 and base != altezza:
        perimetro = calcola_perimetro_rettangolo(base, altezza)
        print("Il perimetro del rettangolo è:", perimetro)
    elif base <= 0 or altezza <= 0:
        print("La base e l'altezza devono essere valori positivi.")
    else:
        print("La base e l'altezza devono essere diverse tra loro.")
```



A seconda della scelta:

- Se "1" viene digitato, il programma chiama la funzione 'calcola_perimetro_quadrato' dopo aver verificato che il lato inserito sia positivo, altrimenti avverte l'utente.
- Se "2" è selezionato, viene eseguita la funzione 'calcola_circonferenza_cerchio' con il raggio fornito dall'utente, a patto che sia un valore positivo.
- Se la scelta è "3", il programma utilizza la funzione 'calcola_perimetro_rettangolo' con la base e l'altezza inserite, purché siano entrambe positive e diverse da zero.

Se l'input non è valido o i valori non sono positivi, il programma non procede con il calcolo e fornisce messaggi di errore specifici per guidare l'utente verso un input corretto.

Per il calcolo del perimetro del quadrato:

- Se l'utente inserisce un valore per il lato del quadrato che non è positivo, il codice entra nel blocco 'else' corrispondente e stampa il messaggio: "Il lato deve essere un valore positivo." Questo previene il calcolo del perimetro con un valore non valido.

Per il calcolo della circonferenza del cerchio:

- Se l'utente fornisce un raggio che non è un valore positivo, il codice come per il calcolo del quadrato emette un avviso: "Il raggio deve essere un valore positivo." Il programma dunque non procede con la formula della circonferenza.

Per il calcolo del perimetro del rettangolo:

- Se l'utente inserisce per la base e per l'altezza del rettangolo dei valori che non sono entrambi positivi o se uno di questi è zero, il codice entra nel blocco else comunicando un messaggio d'errore

In ogni situazione di input non valido, il programma informa l'utente del tipo di errore e non esegue i calcoli fino a quando non vengono forniti valori adeguati. Questo assicura che il programma sia robusto e che gestisca in modo appropriato gli errori di input, guidando l'utente verso la correzione.



Grazie

Gianpaolo Miliccia

Team leader

Fabio Nobili

Antonio Perna

Luca Gaspari

Andrè Vinícius

Samuel Sette

Romano Cascialli