



CONSEGNA S3/L1

di Giuseppe Lupoi

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 scelte:

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

Vediamo il codice da vicino

```
GNU nano 7.2 7.2 test.py * calcolo_perimetro.py
#programma per calcolare il perimetro di diverse forme geometriche
    return 2* (lato)
print ("Ciao, sono un calcolatore di perimetri.")
while True:
    def la_perimetro_cerchio(raggio):
        print ("Scegli la forma che ti interessa:\n 1) Quadrato\n 2) Cerchio\n 3) Rettangolo\n 4) Esci \nQuale perimetro vuoi calcolare?")
    def try: la_perimetro Rettangolo(base, altezza):
        scelta = int(input("~"))
        if scelta = 1:
            print ("Bene, calcoliamo il perimetro del quadrato!")
            lato = int(input("Inserisci un lato del quadrato in cm: "))
            print ("Il perimetro del quadrato e': ",lato *4, "\n")
            #break

        elif scelta = 2:
            print ("Bene, calcoliamo il perimetro del cerchio!")
            raggio = int(input("Inserisci il raggio del cerchio in cm: "))
            diametro = int(raggio *2)
            print ("Il perimetro del cerchio e': ", diametro *3.14, "\n")
            #break

        elif scelta = 3:
            print ("Bene, calcoliamo il perimetro del rettangolo!")
            base = int(input("Inserisci la base del rettangolo in cm: "))
            altezza = int(input("Inserisci l'altezza del rettangolo in cm: "))
            print ("Il perimetro del rettangolo e':", base *2 + altezza*2, "\n")
            #break

        elif scelta = 4:
            break
        else:
            print ("Puoi scegliere solo numeri.")

    except ValueError:
        print ("Scegli solo 1, 2 o 3.")
```

SPIEGAZIONE

Come prima cosa, dopo una piccola frase di intro al programma, ho impostato un ciclo “while” per permettere all’utente di avere una “via di fuga” e non permettere alle domande di andare in loop.

Dopodichè con un altra frase chiederemo all’utente quale forma vuole scegliere per calcolarne il perimetro.

Ho impostato una variabile “scelta” per ricevere la risposta dall’utente, dopo aver inserito l’input ci sarà un ciclo “if-elif-else” appunto per far partire la funzione che l’utente ha scelto.

Ho inserito anche una quarta opzione di uscita dal programma.

L’ “else” dirà all’utente di scegliere solo numeri in caso di immissione di caratteri, l’ “except ValueError” invece manderà un messaggio all’utente in caso inserisse un numero non contemplato.

Vediamo in fine come funziona il programma

```
(kali@kali)-[~/Desktop]
$ python calcolo_perimetro.py to(lato):
Ciao, sono un calcolatore di perimetri.
Scegli la forma che ti interessa:
1) Quadrato la_perimetro_cerchio(raggio):
2) Cerchio return 2* math.pi*raggio
3) Rettangolo _perimetro_rettangolo(base, altez
4) Esci
Quale perimetro vuoi calcolare?
~1
Bene, calcoliamo il perimetro del quadrato!
Inserisci un lato del quadrato in cm: 5
Il perimetro del quadrato e': 20

Scegli la forma che ti interessa:
1) Quadrato
2) Cerchio
3) Rettangolo
4) Esci
Quale perimetro vuoi calcolare?
~2
Bene, calcoliamo il perimetro del cerchio!
Inserisci il raggio del cerchio in cm: 3
Il perimetro del cerchio e': 18.84
```

```
(kali@kali)-[~/Desktop]
$ python calcolo_perimetro.py to(lato):
Ciao, sono un calcolatore di perimetri.
Scegli la forma che ti interessa:
1) Quadrato la_perimetro_cerchio(raggio):
2) Cerchio return 2* math.pi*raggio
3) Rettangolo _perimetro_rettangolo(base, altez
4) Esci
Quale perimetro vuoi calcolare?
~3
Bene, calcoliamo il perimetro del rettangolo!
Inserisci la base del rettangolo in cm: 2
Inserisci l'altezza del rettangolo in cm: 5
Il perimetro del rettangolo e': 14

Scegli la forma che ti interessa:
1) Quadrato
2) Cerchio
3) Rettangolo
4) Esci
Quale perimetro vuoi calcolare?
~4

(kali@kali)-[~/Desktop]
$ █
```