

7. Na Figura 2, está apresentado um programa, em linguagem Python, desenvolvido para simular lançamentos de uma moeda cujas faces são designadas por «cara» e «coroa». O programa calcula e exibe a frequência absoluta de «cara» e de «coroa», depois de solicitar ao utilizador o número de lançamentos pretendidos.

```
import random

n = 0
# Repete enquanto o utilizador não digitar um número maior do que zero
while n <= 0:
    n = int(input("Quantos lançamentos (número inteiro positivo)? "))

cara = 0
coroa = 0

for _ in range(n):
    if random.randint(0, 1) == 0:
        cara += 1
    else:
        coroa += 1

print(f"Número de caras: {cara}")
print(f"Número de coroas: {coroa}")
```

Com o objetivo de obter ainda a percentagem de «caras», foi adicionada ao programa uma linha de código.

Qual das linhas de código seguintes cumpre esse objetivo?

- (A) `print(f"Percentagem de caras: {(cara / coroa) * 100:}%")`
- (B) `print(f"Percentagem de caras: {(coroa / cara) * 100:}%")`
- (C) `print(f"Percentagem de caras: {(n / cara) * 100:}%")`
- (D) `print(f"Percentagem de caras: {(cara / n) * 100:}%")`

Resolução mat.absolutamente.net



7. Observando o programa apresentado, temos que:

- o número total de lançamentos simulados é armazenado na variável "*n*";
- o número de simulações em que o resultado é "cara" é armazenado na variável "*cara*".

Assim, como a percentagem de caras obtida é dada pelo quociente entre o número de caras obtidas e o número total de lançamentos simulados multiplicado por 100; ou seja $\frac{cara}{n} \times 100$, a linha de código que apresenta esta relação entre as duas variáveis é:

```
print(f"Percentagem de caras: {(cara / n) * 100:}%")
```

Resposta: Opção D

**Sugestões/comentários
abaixo:**



E26EE-Questão 7

Conteúdo

Probabilidades. Programação **Python**.
Lançamentos aleatórios de “caras” e “coroas”.
Percentagem de “caras”.

(Assunto 7 ‘Probabilidades’ -11º ano)

Comentário:

O objetivo é lançar uma moeda um certo número de vezes e procurar a frequência absoluta, em percentagem da face “Caras”.

Vamos explorar um pouco estas linhas de programação.

Para podermos usar a função e **randint**, (número inteiro aleatório) é necessário importar a biblioteca “**random**”.

import random

Inicialmente, o valor de n é zero

n=0

até que o utilizador introduza o valor que lhe será solicitado, e que deverá ser um número inteiro positivo.

n = int(input("Quantos lançamentos (número inteiro positivo)? "))

também inicialmente contamos zero caras e zero coroas:

cara = 0

coroa = 0

Uma vez introduzido o número de lançamentos (**n**), vamos percorrer os números entre 1 e n e, em cada vez, vamos gerar um número aleatório inteiro entre zero e um, isto é, dará zero ou dará 1.

Se der “0”, vamos adicionar uma unidade ao “cara”, (**+=**) ser der “1”, somamos uma unidade à “coroa”. (**+=**)

```
for _ in range(n):  
    if random.randint(0, 1) == 0:  
        cara += 1  
    else:  
        coroa += 1
```

Depois de fazer este procedimento para todas as n possibilidades, vamos apresentar o número de “caras” e o número de “coroas” contabilizadas:

```
print(f"Número de caras: {cara}")  
print(f"Número de coroas: {coroa}")
```

Reparemos ainda que o “f” que está antes das aspas, indica que o conteúdo dentro de chavetas deve se convertido para texto.

Se quisermos obter a percentagem de “caras”, é necessário dividir o número de “caras” pelo total n, e multiplicar por 100. Este procedimento corresponde à alínea D.

(D) `print(f"Percentagem de caras: {(cara / n) * 100:}%)")`

Reparemos ainda que o “f” que está antes das aspas, indica que o resultado do cálculo dentro de chavetas deve se convertido para texto.

(Consultar a resposta completa)