

Python - Automat do kawy

© Copyright by 3bird Projects 2023, <http://edukacja.3bird.pl>

Uwagi ogólne

Uwaga! Wcięcia w kodzie, mają dla Pythona znaczenie (są konieczne w odpowiednich miejscach)! Python do pobrania (dla Windows): <https://www.python.org/downloads/windows/>

Uruchamianie skryptu:

```
C:\> python naszSkrypt.py
```

Uwaga: Nigdy nie wolno kopiować kodu z PDF-a, gdyż zawiera on niewidoczne znaki końca linii i tzw. twarde odstępny. Kod należy przepisać ze zrozumieniem.

Nasz skrypt

Zadaniem programu jest określenie, ile monet i o jakich nominałach powinien automat wydać klientowi, aby było to najbardziej ekonomiczne (optymalne). Poniższy skrypt reprezentuje filozofię „**algorytmu zachłannego**” (wybór najlepszego, optymalnego rozwiązania na każdym etapie algorytmu; na każdym etapie szukamy najlepszego rozwiązania dla następnego kroku, a nie dla całego problemu, stąd algorytm zachłanny nie zawsze daje najbardziej optymalny wynik dla całego problemu).

Kod skryptu - wersja podstawowa

```
#!/usr/bin/env python
```

```
# Powyższa linia, tylko dla systemu Linux.
```

```
def ile_monet(resztaDoWydania):
```

```
    nominaly = [50,20,10,5,2,1] # Nominały monet (grosze)
```

```
    iloscWydanychMonet = 0 # Zerowanie zmiennej (automat nie wydaje reszty dopóki jej nie obliczy)
```

```
    print("Automat wyda klientowi następujące monety:")
```

```
    for kolejnyElement in nominaly:
```

```
        # Poniższa linia pokazuje ile razy dana moneta zostanie wydana.
```

```
        # Operator // oznacza wynik dzielenia zaokrąglony w dół do pełnych wartości,
```

```
        # np. 11 // 50 = 0,22... czyli 0; 11 // 20 = 0,55... czyli 0:
```

```
        print("\n - o nominale", kolejnyElement, "gr wyda", resztaDoWydania // kolejnyElement, "raz(y).")
```

```
        iloscWydanychMonet = iloscWydanychMonet + resztaDoWydania // kolejnyElement
```

```
        # Przeliczenie kwoty: reszta z dzielenia, czyli 11 % 20 = 11, gdyż w przypadku
```

```
        # dzielenia liczb mniejszych przez większe, wynikiem zawsze będzie dzielna.
```

```
        # W przypadku 11 % 20, nadal do wydania pozostaje 11gr.
```

```
        resztaDoWydania = resztaDoWydania % kolejnyElement
```

return iloscWydanychMonet

Pobieramy z klawiatury wartość reszty do wydania przez automat:

resztaDoWydania = int(input("Wprowadź kwotę (w groszach), jaką automat musi wydać klientowi: "))

Wypisujemy komunikat, ale jednocześnie uruchamiamy funkcję z wartością wpisaną powyżej przez użytkownika. Funkcja zwraca nam najmniejszą ilość monet:

print("Wydana reszta będzie składać się z", ile_monet(resztaDoWydania), "monet(y).\n")

input('\n\nNaciśnij ENTER, aby zakończyć...\n')

Kod skryptu - wersja rozbudowana

#!/usr/bin/env python

Powyższa linia, tylko dla systemu Linux.

from os import system *# Wymagane do kolorowania składni w systemie Windows 10/11:*

system(" ")

Uwaga: Tekst nie będzie kolorowany, gdy uruchomimy go w „IDLE Shell” (bo to nie jest prawdziwy terminal) oraz w Windows 7/8.

Tytuł programu:

print("\n\n\033[1;37;44m AUTOMAT SZKOLNY Z KAWĄ \033[0m \n")

def ile_monet(resztaDoWydania):

nominaly = [50,20,10,5,2,1] *# Nominały monet (grosze)*

iloscWydanychMonet = 0 *# Zerowanie zmiennej (automat nie wydaje reszty dopóki jej # nie obliczy)*

print("\033[0;33;40m \nAutomat wyda klientowi następujące monety:")

for kolejnyElementListyNominaly in nominaly:

Poniższa linia pokazuje ile razy dana moneta zostanie wydana.

Operator // oznacza wynik dzielenia zaokrąglony w dół do pełnych wartości,

np. 11gr // 50gr = 0,22... czyli 0 monet; 11gr // 20gr = 0,55... czyli 0 monet;

ale 11gr // 10gr = 1,10... czyli 1 moneta.

print("\n - o nominale", kolejnyElementListyNominaly, "gr wyda",

resztaDoWydania // kolejnyElementListyNominaly, "raz(y).")

Poniżej: 0 wydanych monet = 0 + (kwotaDoWydania (czyli 11gr) // kolejno na

50gr / 20gr / 10gr itd.). Przy dzieleniu 11gr // 10gr (nominat) otrzymamy 1. Czyli

zmienna "iloscWydanychMonet" będzie równa 1 (w tym przejściu pętli).

Zmienna ta zwiększy wartość za każdym razem, gdy zostanie wydana kolejna

moneta z reszty.

iloscWydanychMonet = iloscWydanychMonet + resztaDoWydania //

kolejnyElementListyNominaly

Ile jeszcze zostało do wydania reszty po tym przejściu pętli?

```
# Przeliczenie kwoty: reszta z dzielenia, czyli np. 11gr % 20gr = nadal 11gr, gdyż w
# przypadku dzielenia liczb mniejszych przez większe, wynikiem zawsze będzie
# dzielna. W przypadku 11gr % 20gr, nadal do wydania pozostaje 11gr. Ale już w
# przypadku 11gr % 10gr = 1gr (do wydania w kolejnym przebiegu pętli).
resztaDoWydania = resztaDoWydania % kolejnyElementListyNominaly
```

```
return iloscWydanychMonet
```

```
try:
```

```
# Pobieramy z klawiatury wartość reszty do wydania przez automat:
```

```
resztaDoWydania = int(input("Wprowadź kwotę (w groszach), jaką automat musi  
wydać klientowi: "))
```

```
if resztaDoWydania >= 0:
```

```
    # Wypisujemy komunikat, ale jednocześnie uruchamiamy funkcję z wartością
    # wpisaną powyżej przez użytkownika. Funkcja zwraca nam najmniejszą
    # ilość monet:
```

```
    print("\n\033[1;33;40mWydana reszta będzie składać się z",  
        ile_monet(resztaDoWydania), "monet(y).\n")
```

```
else:
```

```
    print("Reszta, którą ma wydać automat, musi być liczbą dodatnią!")
```

```
except ValueError:
```

```
    print("Musisz wprowadzić dodatnią liczbę całkowitą!")
```

```
finally:
```

```
    input('\n\nNaciśnij ENTER, aby zakończyć...\n')
```

Ostatnia aktualizacja: 10 grudnia 2023.