ІНФОРМАТИКА І ПРОГРАМУВАННЯ

СТАНДАРТНА БІБЛІОТЕКА. ІМПОРТ

Деякі модулі стандартної бібліотеки

`collections, copy, datetime, fractions, itertools, json, math, multiprocessing, pickle, random, re, statistics, tkinter, turtle, time, ...`

Основна ідея

Не треба програмувати те, що вже хтось запрограмував - користуйтеся на здоров'я!

Модуль – файл з розширенням .py

Атрибут – змінна або функція, яка описана в модулі

Точкова нотація (для доступу до атрибуту модуля):

```
модуль.атрибут
```

Інструкція import

Припустимо, існує модуль modul, який містить змінну var і функцію func()

```
import modul      # імпорт модуля

modul.var         # доступ до змінної

modul.func()      # виклик функції
```

Інструкція from … import ...

Дає змогу уникнути громіздкої точкової нотації

```
from modul import *      # імпорт всього вмісту модуля
var                      # доступ до змінної
func()                   # виклик функції
```

```
from modul import var, func      # імпорт окремих атрибутів
var                              # доступ до змінної
func()                           # виклик функції
```

Припустимо, що змінна var модуля modul має значення 1, а функція func() збільшує передане їй число на одиницю

```
var = 10
from modul import *
result = func(var)
print(result)              # Що буде надруковано?
```

```
var = 10
from modul import func      # Чи виправить це ситуацію?
result = func(var)
print(result)
```

Для зменшення ймовірності виникнення конфлікту імен інструкції імпорту ставляться завжди на початку програми і відокремлюються від решти коду порожнім рядком

```
from modul import *

var = 10
result = func(var)
print(result)
```

```
import модуль

from модуль import атрибут1, атрибут2, ...

from модуль import *
```

Інструкції import … as ... і from … import … as ...

Дають нові імена модулям і атрибутам (в межах програми)

```
import modul as m
m.var                                   # доступ до змінної
m.func()                                # виклик функції
```

```
from modul import var as v
from modul import func as f
v                                       # доступ до змінної
f()                                     # виклик функції
```

Припустимо, що обидва модулі modul_1 і modul_2, містять змінну var, значення якої 1 і 2 відповідно

```
from modul_1 import var
from modul_2 import var

print(var + var)
```

У чому тут проблема? Як її вирішити?

```
import modul_1
import modul_2

print(modul_1.var + modul_2.var)
```

```
from modul_1 import var as var1
from modul_2 import var as var2

print(var1 + var2)
```

Модуль math

```
e               # 2.718281828459045
pi              # 3.141592653589793
inf             # нескінченність
sin(x)          # синус числа x (в радіанах)
degrees(x)      # перетворює кут x з радіанної міри в градусну
radians(x)      # перетворює кут x з градусної міри в радіанну
ceil(x)         # заокруглює x до найближчого більшого цілого
floor(x)        # заокруглює x до найближчого меншого цілого
factorial(x)    # x! (факторіал числа x)
gcd(x,y)        # найбільший спільний дільник чисел x і y
hypot(x,y)      # гіпотенуза трикутника з катетами x і y
```

```
# Програма обчислює кут між векторами (x1,y1) і (x2,y2)

import math

x1 = float(input('x1 = ')); y1 = float(input('y1 = '))
x2 = float(input('x2 = ')); y2 = float(input('y2 = '))

m1 = math.hypot(x1, y1)        # довжина першого вектора
m2 = math.hypot(x2, y2)        # довжина другого вектора
sp = x1*x2 + y1*y2             # скалярний добуток векторів
rad = math.acos(sp/m1/m2)      # кут у радіанах
gra = math.degrees(rad)        # кут у градусах

print('Кут між векторами у радіанах:', rad)
print('Кут між векторами у градусах:', gra)
```

Модуль random

```
randint(x, y)
# повертає випадкове ціле число з проміжку [x; y]

random()
# повертає випадкове дійсне число з проміжку [0; 1]

uniform(x, y)
# повертає випадкове дійсне число з проміжку [x, y]

shuffle(L)        # перемішує елементи списку L
```

Модуль time

```
time()
'''повертає кількість секунд, яка пройшла від 0 годин 0 хвилин
1 січня 1970 року (в середовищі програмістів цей момент часу
називається “початок Епохи”) до системного часу комп’ютера в
момент її виклику'''
```

```
#    Програма    обчислює    факторіал    випадкового    числа
# і виводить час виконання

import math
from random import randint as rnd
from time import time

start = time()
number = rnd(1,100)
fact = math.factorial(number)

print(number, '! = ', fact, sep = '')

print('time:', round(time()-start, 5))
```