

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки
Кафедра обчислювальної техніки

Програмування
Лабораторна робота №6
**«Основи об'єктно-орієнтованого програмування.
Модулі та пакети»**

Виконала:
студентка групи ІО-64
Бровченко А. В.
Залікова книжка №6403
Перевірів Новотарський М. А.

Київ
2016 р.

Мета: вивчити способи створення та підключення модулів та пакетів. Основи ООП. Методи і атрибути класів та робота з ними. Побудова програми у стилі ООП.

Завдання:

1. Вивчити матеріал лекцій 18, 19, 20 та 21.
2. Виконати індивідуальне завдання лабораторної роботи, вибране відповідно до варіанту.

Теоретичні відомості

Інструкція import

Імпортувати модуль дозволяє інструкція import. Інструкція import має наступний формат:

```
import <Назва модуля1> [<Псевдонім>][, ...,  
                        <Назва модуляn> [<Псевдонім>]]
```

Після імпортування модуля його назва стає ідентифікатором, через який можна одержати доступ до атрибутів, визначених усередині модуля.

Доступ до атрибутів модуля здійснюється за допомогою точкової нотації.

Модуль random

`random.choice(sequence)` – повертає випадковий елемент непустої послідовності

Визначення класу й створення екземпляра класу

Клас описують за допомогою ключового слова `class` за наступною схемою:

```
class <Назва класу>[(<Клас1>[, ... , <Класn>])]:  
    [""] Рядок документування ""  
    <Опис атрибутів і методів>
```

Інструкція створює новий об'єкт і присвоює посилання на нього ідентифікатору, зазначеному після ключового слова `class`.

Це означає, що назва класу повинна повністю відповідати правилам іменування змінних. Після назви класу в круглих дужках можна вказати один або кілька базових класів через кому. Якщо ж клас не успадковує базові класи, то круглі дужки можна не вказувати.

Всі вирази всередині інструкції `class` виконуються при створенні класу, а не його екземпляра.

Створення атрибута класу аналогічно створенню звичайної змінної.

Метод всередині класу створюється так само, як і звичайна функція, за допомогою інструкції `def`.

Методам класу в першому параметрі, який обов'язково слід вказати явно, автоматично передають посилання на екземпляр класу. Загальноприйнято цей параметр називати ім'ям `self`, хоча це й не обов'язково.

Доступ до атрибутів і методів класу всередині обумовленого методу проводиться через змінну `self` за допомогою точкової нотації – до атрибута `x` з методу класу можна звернутися так: `self.x`.

Щоб використовувати атрибути й методи класу, необхідно створити екземпляр класу згідно з наступним синтаксисом:

```
<Екземпляр класу> = <Назва класу>([<Параметри>])
```

При доступі до методів класу використовують такий формат:

```
<Екземпляр класу>.<Ім'я методу>([<Параметри>])
```

Метод `__init__()`

При створенні екземпляра класу інтерпретатор автоматично викликає метод ініціалізації `__init__()`. В інших мовах програмування такий метод прийнято називати конструктором класу.

Формат методу:

```
def __init__(self[, <Значення1>[, ... ,<Значення> ]]):  
    <Інструкції>
```

За допомогою методу `__init__()` можна присвоїти початкові значення атрибутам класу. При створенні екземпляра класу параметри цього методу вказують після імені класу в круглих дужках:

```
<Екземпляр класу> = <Ім'я класу>([<Значення1>[, ... ,  
<Значення> ]])
```

Індивідуальне завдання

Варіант 3

Завдання:

Створити клас, який описує місяць року і містить назву місяця, список днів даного місяця, довжина якого формується в залежності від номеру місяця. Кожен день може бути сонячним, хмарним або дощовим. Кожен день також характеризується середньодобовою температурою. Клас має містити методи визначення середньої кількості сонячних хмарних та дощових днів, методи визначення мінімальної, максимальної та середньої температури. Створити об'єкти даного класу, для кожного місяця року, задати кількості сонячних, хмарних, дощових днів та середньодобову температуру кожного дня випадковим чином з урахуванням пори року та номеру місяця. Програма повинна виводити за запитом середні кількості сонячних, хмарних, дощових днів та середню температуру для кожного місяця за запитом.

Алгоритм виконання:

При створенні екземпляра класу, вводиться 3 параметри: номер місяця, рік і умова (option).

В класі робиться перевірка чи вказаний рік є високосним. В залежності від номеру місяця і року формується список днів місяця (у високосному році в лютому 29 днів).

Далі проводиться опис кожного дня місяця. Якщо умова = 1, то опис робиться випадковим чином, з урахуванням пори року і номеру місяця. Якщо умова = 0, то опис проводиться вручну. По закінченню, кожен день описується середньодобовою температурою та погодою, яка може бути сонячною, хмарною, дощовою та сніжною.

У класі є методи для визначення кількості сонячних, хмарних, дощових та сніжних днів, мінімальної, максимальної та середньої температури.

Також у класі є метод для повного опису місяця, який виводить назву місяця, кількість днів у місяці, кількість сонячних, хмарних, дощових та сніжних днів, мінімальну, максимальну та середню температуру за місяць.

Код програми:

```
import random

class Month:
    """Class describes the month of the year"""

    months = {1: 'January', 2: 'February', 3: 'March', 4: 'April',
              5: 'May', 6: 'June', 7: 'July', 8: 'August',
              9: 'September', 10: 'October', 11: 'November', 12: 'December'}
```

```

def __init__(self, number, year, option):
    """Number - number of the month.
    Option = 0 if you want to input the description of days manually.
    Else: type 1"""
    if number not in range(1, 13):
        raise ValueError
    self.number = number

    # Назва місяця
    self.name = Month.months[number]

    # Кількість днів
    if number in [1, 3, 5, 7, 8, 10, 12]:
        ndays = list(range(1, 32))
    elif number in [4, 6, 9, 11]:
        ndays = list(range(1, 31))
    elif number == 2:
        if year in range(0, 5000, 4):
            ndays = list(range(1, 30))
        else:
            ndays = list(range(1, 29))
    else: ndays = []
    self.ndays = ndays

    # Опис днів: 1 - випадково, 0 - вручну
    if option == 1:
        description_of_days = ['snowy', 'sunny', 'cloudy', 'rainy']
        days = dict.fromkeys(ndays)

        if number == 1:
            temp = list(range(-15, -2))
            for key in days.keys():
                days[key] = (random.choice(temp),
random.choice(description_of_days[:3]))
        elif number == 2:
            temp = list(range(-14, 3))
            for key in days.keys():
                days[key] = (random.choice(temp),
random.choice(description_of_days[:3]))
        elif number == 3:
            temp = list(range(-7, 1))
            for key in days.keys():
                days[key] = (random.choice(temp),
random.choice(description_of_days[1:]))
        elif number == 4:
            temp = list(range(5, 15))
            for key in days.keys():
                days[key] = (random.choice(temp),
random.choice(description_of_days[1:]))
        elif number == 5:
            temp = list(range(12, 25))
            for key in days.keys():
                days[key] = (random.choice(temp),
random.choice(description_of_days[1:]))
        elif number == 6:
            temp = list(range(18, 28))
            for key in days.keys():
                days[key] = (random.choice(temp),
random.choice(description_of_days[1:]))
        elif number == 7:
            temp = list(range(20, 33))
            for key in days.keys():
                days[key] = (random.choice(temp),
random.choice(description_of_days[1:]))
        elif number == 8:
            temp = list(range(20, 33))

```

```

        for key in days.keys():
            days[key] = (random.choice(temp),
random.choice(description_of_days[1:]))
        elif number == 9:
            temp = list(range(12, 29))
            for key in days.keys():
                days[key] = (random.choice(temp),
random.choice(description_of_days[1:]))
            elif number == 10:
                temp = list(range(3, 16))
                for key in days.keys():
                    days[key] = (random.choice(temp),
random.choice(description_of_days[1:]))
                elif number == 11:
                    temp = list(range(-3, 8))
                    for key in days.keys():
                        days[key] = (random.choice(temp),
random.choice(description_of_days[1:]))
                    elif number == 12:
                        temp = list(range(-7, -2))
                        for key in days.keys():
                            days[key] = (random.choice(temp),
random.choice(description_of_days[:3]))

            elif option == 0:
                description_of_days = ['snowy', 'sunny', 'cloudy', 'rainy']
                days = dict.fromkeys(self.days)
                for key in days.keys():
                    print('Day', key)
                    temp = input('Average temperature = ')
                    desc = int(input('Choose the description: snowy = 0, sunny = 1, cloudy
= 2, rainy = 3'))
                    if desc not in range(0, 4):
                        raise ValueError
                    days[key] = (temp, description_of_days[desc])

            else: raise ValueError
            self.days = days

def sunny_days(self):
    """Counts number of sunny days"""
    s = []
    for key in self.days:
        s.append(self.days[key][1])
    return 'In {} were {} sunny days'.format(self.name, s.count('sunny'))

def cloudy_days(self):
    """Counts number of cloudy days"""
    s = []
    for key in self.days:
        s.append(self.days[key][1])
    return 'In {} were {} cloudy days'.format(self.name, s.count('cloudy'))

def rainy_days(self):
    """Counts number of rainy days"""
    s = []
    for key in self.days:
        s.append(self.days[key][1])
    return 'In {} were {} rainy days'.format(self.name, s.count('rainy'))

def snowy_days(self):
    """Counts number of snowy days"""
    s = []
    for key in self.days:
        s.append(self.days[key][1])
    return 'In {} were {} snowy days'.format(self.name, s.count('snowy'))

```

```

def min_temperature(self):
    """Returns the lowest temperature of the month"""
    s = []
    for key in self.days:
        s.append(self.days[key][0])
    return 'The lowest temperature in {} was {}'.format(self.name, min(s))

def max_temperature(self):
    """Returns the highest temperature of the month"""
    s = []
    for key in self.days:
        s.append(self.days[key][0])
    return 'The highest temperature in {} was {}'.format(self.name, max(s))

def average_temperature(self):
    """Returns average temperature of the month"""
    s = []
    for key in self.days:
        s.append(self.days[key][0])
    return 'Average temperature in {} was {}'.format(self.name,
round(sum(s)/len(s), 2))

def full_description(self):
    """Full description of the month: name, number and characteristic of days,
average temperature, highest and lowest temperature."""
    d = [] ; t = []
    for key in self.days:
        d.append(self.days[key][1])
        t.append(self.days[key][0])
    return 'In {} were {} days. Including {} sunny, {} cloudy, {} rainy, {}
snowy.\n' \
        'Average temperature of the month: {}\n' \
        'The lowest temperature: {}\n' \
        'The highest temperature: {}'.format(self.name, len(self.ndays),
d.count('sunny'), d.count('cloudy'),
d.count('rainy'), d.count('snowy'),
round(sum(t)/len(t), 2), min(t), max(t))

# Створюємо об'єкти класу і описуємо кожен місяць
a = Month(1, 2016, 1)
print(a.full_description())
b = Month(2, 2016, 1)
print(b.full_description())
c = Month(3, 2016, 1)
print(c.full_description())
d = Month(4, 2016, 1)
print(d.full_description())
e = Month(5, 2016, 1)
print(e.full_description())
f = Month(6, 2016, 1)
print(f.full_description())
g = Month(7, 2016, 1)
print(g.full_description())
h = Month(8, 2016, 1)
print(h.full_description())
i = Month(9, 2016, 1)
print(i.full_description())
j = Month(10, 2016, 1)
print(j.full_description())
k = Month(11, 2016, 1)
print(k.full_description())
l = Month(12, 2016, 1)
print(l.full_description())

```

Вивід програми з контрольним прикладом:

In January were 31 days. Including 14 sunny, 10 cloudy, 0 rainy, 7 snowy.

Average temperature of the month: -10.16

The lowest temperature: -15

The highest temperature: -3

In February were 29 days. Including 10 sunny, 11 cloudy, 0 rainy, 8 snowy.

Average temperature of the month: -6.41

The lowest temperature: -14

The highest temperature: 2

In March were 31 days. Including 10 sunny, 6 cloudy, 15 rainy, 0 snowy.

Average temperature of the month: -3.0

The lowest temperature: -7

The highest temperature: 0

In April were 30 days. Including 10 sunny, 8 cloudy, 12 rainy, 0 snowy.

Average temperature of the month: 9.53

The lowest temperature: 5

The highest temperature: 14

In May were 31 days. Including 11 sunny, 15 cloudy, 5 rainy, 0 snowy.

Average temperature of the month: 17.16

The lowest temperature: 12

The highest temperature: 24

In June were 30 days. Including 10 sunny, 10 cloudy, 10 rainy, 0 snowy.

Average temperature of the month: 22.8

The lowest temperature: 18

The highest temperature: 27

In July were 31 days. Including 6 sunny, 16 cloudy, 9 rainy, 0 snowy.

Average temperature of the month: 25.03

The lowest temperature: 20

The highest temperature: 32

In August were 31 days. Including 12 sunny, 12 cloudy, 7 rainy, 0 snowy.

Average temperature of the month: 25.87

The lowest temperature: 20

The highest temperature: 32

In September were 30 days. Including 12 sunny, 10 cloudy, 8 rainy, 0 snowy.

Average temperature of the month: 17.93

The lowest temperature: 12

The highest temperature: 27

In October were 31 days. Including 13 sunny, 9 cloudy, 9 rainy, 0 snowy.

Average temperature of the month: 8.48

The lowest temperature: 3

The highest temperature: 15

In November were 30 days. Including 11 sunny, 9 cloudy, 10 rainy, 0 snowy.

Average temperature of the month: 2.03

The lowest temperature: -3

The highest temperature: 7

In December were 31 days. Including 9 sunny, 12 cloudy, 0 rainy, 10 snowy.

Average temperature of the month: -4.55

The lowest temperature: -7

The highest temperature: -3

Process finished with exit code 0

Висновки:

1. Завдання досить цікаве; для його виконання потрібно не лише вивчити матеріал 18-21 лекції, а і добре засвоїти і вміти користуватись попереднім вивченим матеріалом.
2. Завдання виконано і трохи покращено. Програма працює згідно з умовою.
3. При виконанні завдання виникли складнощі із розумінням самого завдання, які згодом успішно були усунуті.