Projektowanie i Eksploatacja Sieci Komputerowych
Grafy w środowisku Python

1. Utwórz nowy katalog i plik na kod pythona na najbliższe zajęcia
2. Dictionary w pythonie to tablica asocjacyjna, której elementami mogą być inne elementy języka (listy, tablice, krotki etc.). Sprawdź, jak działa dictionary wykonując następujący kod:
```
graph = { "a" : ["c"],
    "b" : ["c", "e"],
    "c" : ["a", "b", "d", "e"],
    "d" : ["c"],
    "e" : ["c", "b"],
    "f" : []
  }

print(graph)
```
3. Spróbuj również wypisać jeden z elementów korzystając z konstrukcji graph["c"]
4. Wykorzystując wiedzę z ostatnich zajęć utwórz klasę Graph. Dodaj do niej funkcję __init__(self, g), która będzie przypisywać zawartość miennej g do wewnętrznej zmiennej klasowej self.graph
5. Do klasy Graph dodaj funkcję wypisującą o nazwe printAll(), która:
    1. W przypadku grafu pustego wypisze "Empty graph"
    2. W przeciwnym przypadku wypisze elementy poleceniem print lub pętlą for x in ...
    3. Skorzystaj z funkcji len(nazwa_zmiennej) do sprawdzenia długości słownika
6. Dodaj do klasy funkcję addVertex(self, x) dodającą wierzchołek o nazwie zdefiniowanej w zmiennej x do grafu. Przed dodaniem sprawdź poleceniem if x not in self.graph czy nowy wierzchołek jest już w grafie. Jeśli nie ma, dodaj nowy element słownika o podanej nazwie. Jeśli jest, nie rób nic.
7. Dodaj do klasy funkcję, dodającą krawędź pomiędzy dwoma wybranymi wierzchołkami. Funkcja ma przyjmować dwa argumenty łańcuchowe, będące identyfikatorami wierzchołków do połączenia. Pamiętaj o sprawdzeniu czy wierzchołki istnieją (dodaj nowe jeśli będzie taka potrzeba). Pamiętaj również o dodaniu informacji zwrotnej o istnieniu krawędzi dla drugiego jej końca (przy krawędzi a – c zarówno wierzchołek "a" ma otrzymać wpis że jest połączony z c jak i c, że jest powiązany z a)
8. Dodaj do klasy funkcję writeToDot(self, path), która korzystając z następującego przykładowego kodu:
```
h = open(path,"w")
write(zmienna_z_zawartoscia)
```
oraz wykorzystując konstrukcję for x in ..., wypisze stworzony graf do formatu Dot.
9. Przetestuj wygląd zapisanego grafu z kodem i sprawdź, czy wszystko się zgadza.
10. Uzupełnij kod klasy o reakcję na podanie pustego dictionary do __init__ (program ma działać dalej)
11. Przetestuj następujący kod (może być w innym pliku)
import random
print(random.randint(0,5))
12. Korzystając z randint spróbuj dodać do klasy funkcję addRandom(self, newKeyName), która doda wierzchołek w losowe miejsce w grafie. Do znalezienia identyfikatora wylosowanego wierzchołka użyj konstrukcji:
list(self.graph)[n]
która zwróci klucz, pod którym jest n-ty element dictionary
13. Kod programu zamieść w sprawozdaniu