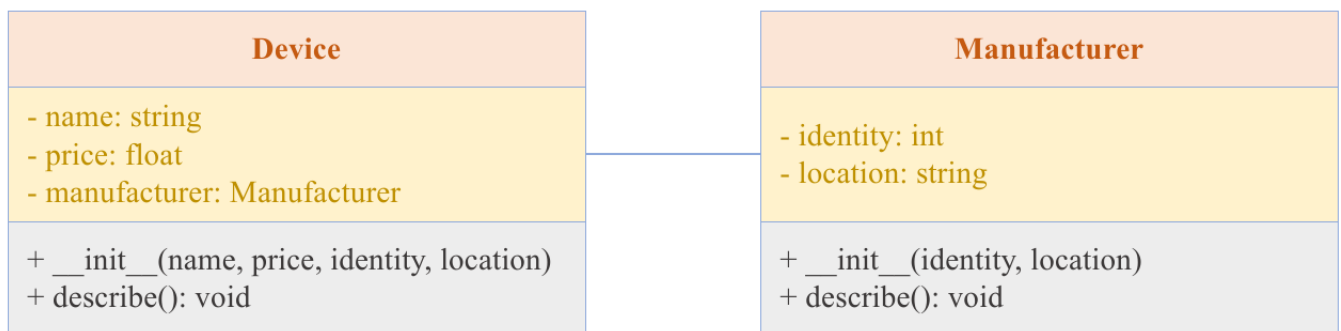


AI VIET NAM – COURSE 2022

Python OOP – Exercise

Ngày 13 tháng 6 năm 2022

1. Thực hiện theo yêu cầu như class diagram bên dưới.



Hình 1: Class Diagram

```
1 # Examples 1
2 device1 = Device(name="mouse", price=2.5, identity=9725, location="Vietnam")
3 device1.describe()
4 >> Name: mouse - Price: 2.5
5 Identity: 9725 - Location: Vietnam
6
7 device2 = Device(name="monitor", price=12.5, identity=11, location="Germany")
8 device2.describe()
9 >> Name: monitor - Price: 12.5
10 Identity: 11 - Location: Germany
```

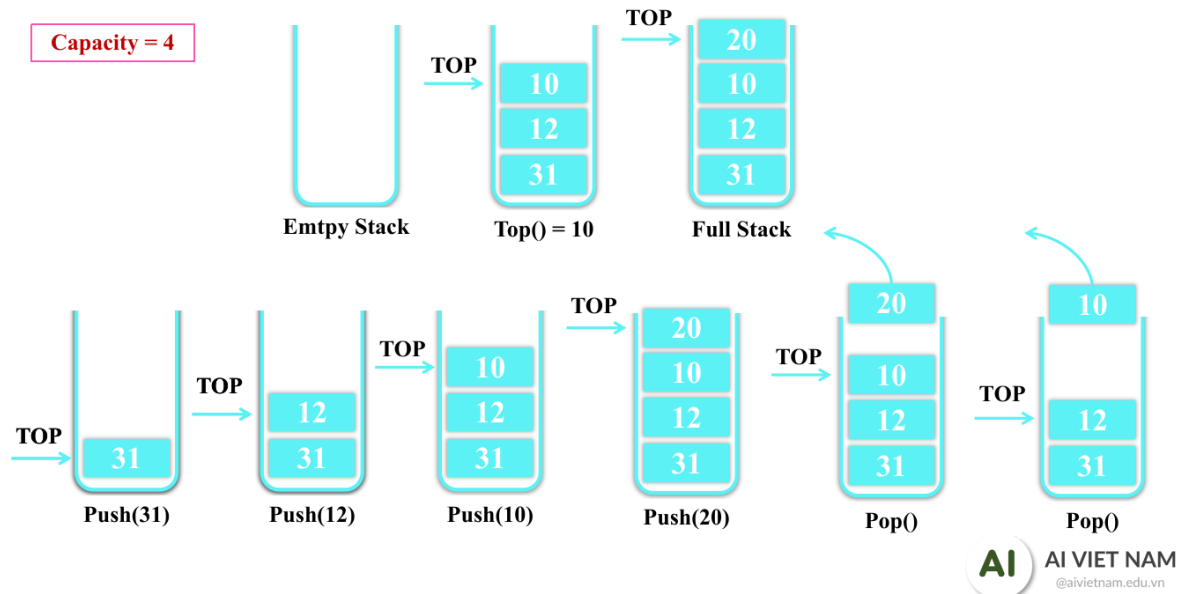
2. Một Ward gồm có name (string) và danh sách của mọi người trong Ward. Một người person có thể là student, doctor, hoặc teacher. Một student gồm có name, yob (int) (năm sinh), và grade (string). Một teacher gồm có name, yob, và subject (string). Một doctor gồm có name, yob, và specialist (string). Lưu ý cần sử dụng a list để chứa danh sách của mọi người trong Ward.
- Thực hiện các class student, doctor, và teacher theo mô tả trên. Thực hiện describe() method để print ra tất cả thông tin của các objects.
 - Viết addPerson(person) method trong Ward class để add thêm một người mới với nghề nghiệp bất kỳ (student, teacher, doctor) vào danh sách người của ward. Tạo ra một ward object, và thêm vào 1 student, 2 teacher, và 2 doctor. Thực hiện describe() method để in ra tên ward (name) và toàn bộ thông tin của từng người trong ward.
 - Viết countDoctor() method để đếm số lượng doctor trong ward.
 - Viết sortAge() method để sort mọi người trong ward theo tuổi của họ với thứ tự tăng dần. (hint: Có thể sử dụng sort của list hoặc viết thêm function đều được)
 - Viết aveTeacherYearOfBirth() method để tính trung bình năm sinh của các teachers trong ward.

```

1 # Examples
2 # 2(a)
3 student1 = Student(name="studentA", yob=2010, grade="7")
4 student1.describe()
5 #output
6 >> Student - Name: studentA - YoB: 2010 - Grade: 7
7
8 teacher1 = Teacher(name="teacherA", yob=1969, subject="Math")
9 teacher1.describe()
10 #output
11 >> Teacher - Name: teacherA - YoB: 1969 - Subject: Math
12
13 doctor1 = Doctor(name="doctorA", yob=1945, specialist="Endocrinologists")
14 doctor1.describe()
15 #output
16 >> Doctor - Name: doctorA - YoB: 1945 - Specialist: Endocrinologists
17
18
19 # 2(b)
20 print()
21 teacher2 = Teacher(name="teacherB", yob=1995, subject="History")
22 doctor2 = Doctor(name="doctorB", yob=1975, specialist="Cardiologists")
23 ward1 = Ward(name="Ward1")
24 ward1.addPerson(student1)
25 ward1.addPerson(teacher1)
26 ward1.addPerson(teacher2)
27 ward1.addPerson(doctor1)
28 ward1.addPerson(doctor2)
29 ward1.describe()
30
31 #output
32 >> Ward Name: Ward1
33 Student - Name: studentA - YoB: 2010 - Grade: 7
34 Teacher - Name: teacherA - YoB: 1969 - Subject: Math
35 Teacher - Name: teacherB - YoB: 1995 - Subject: History
36 Doctor - Name: doctorA - YoB: 1945 - Specialist: Endocrinologists
37 Doctor - Name: doctorB - YoB: 1975 - Specialist: Cardiologists
38

```

```
39 # 2(c)
40 print(f"\nNumber of doctors: {ward1.countDoctor()}")
41
42 #output
43 >> Number of doctors: 2
44
45 # 2(d)
46 print("\nAfter sorting Age of Ward1 people")
47 ward1.sortAge()
48 ward1.describe()
49
50 #output
51 >> After sorting Age of Ward1 people
52 Ward Name: Ward1
53 Student - Name: studentA - YoB: 2010 - Grade: 7
54 Teacher - Name: teacherB - YoB: 1995 - Subject: History
55 Doctor - Name: doctorB - YoB: 1975 - Specialist: Cardiologists
56 Teacher - Name: teacherA - YoB: 1969 - Subject: Math
57 Doctor - Name: doctorA - YoB: 1945 - Specialist: Endocrinologists
58
59 # 2(e)
60 print(f"\nAverage year of birth (teachers): {ward1.aveTeacherYearOfBirth()}")
61
62 #output
63 >> Average year of birth (teachers): 1982.0
```



Hình 2: Stack

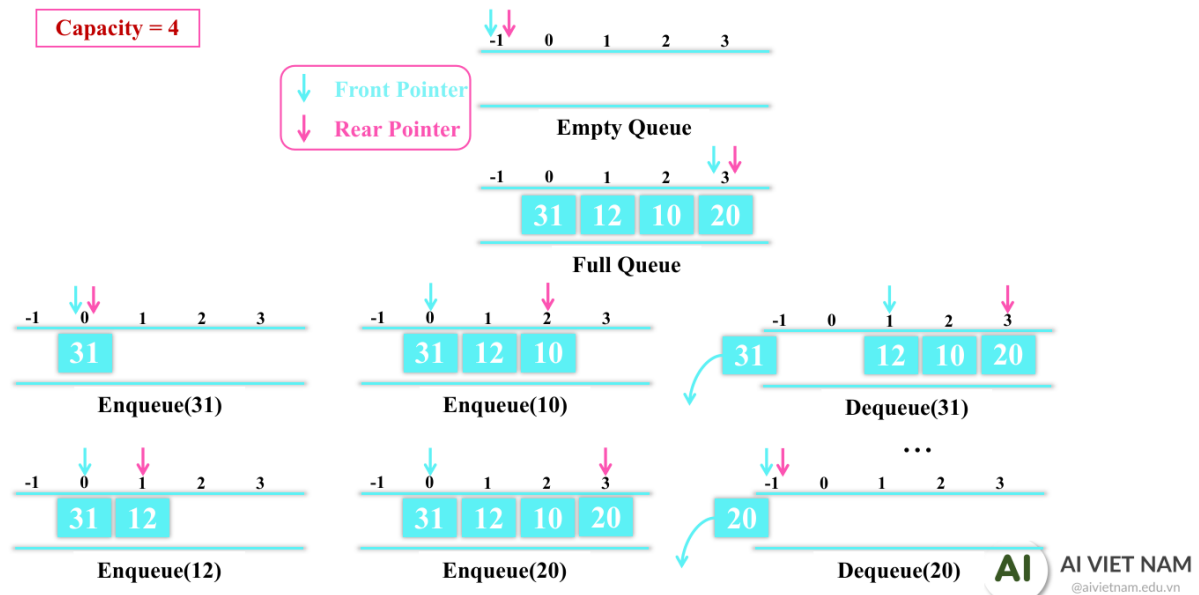
3. Thực hiện xây dựng class Stack với các chức năng (method) sau đây

- **initialization** method nhận một input "capacity": dùng để khởi tạo stack với capacity là số lượng element mà stack có thể chứa
- **.isEmpty()**: kiểm tra stack có đang rỗng
- **.isFull()**: kiểm tra stack đã full chưa
- **.pop()**: loại bỏ top element và trả về giá trị đó
- **.push(value)** add thêm value vào trong stack
- **.top()** lấy giá trị top element hiện tại của stack, nhưng không loại bỏ giá trị đó
- Không cần thiết phải thực hiện với pointer như trong hình minh họa

```

1 stack1 = MyStack(capacity=5)
2
3 stack1.push(1)
4
5 stack1.push(2)
6
7 print(stack1.isFull())
8 >> False
9
10 print(stack1.top())
11 >> 2
12
13 print(stack1.pop())
14 >> 2
15
16 print(stack1.top())
17 >> 1
18
19 print(stack1.pop())
20 >> 1
21
22 print(stack1.isEmpty())
23 >> True

```



Hình 3: Queue

4. Thực hiện xây dựng class Queue với các chức năng (method) sau đây

- **initialization** method nhận một input "capacity": dùng để khởi tạo queue với capacity là số lượng element mà queue có thể chứa
- **.isEmpty()**: kiểm tra queue có đang rỗng
- **.isFull()**: kiểm tra queue đã full chưa
- **.dequeue()**: loại bỏ first element và trả về giá trị đó
- **.enqueue(value)** add thêm value vào trong queue
- **.front()** lấy giá trị first element hiện tại của queue, nhưng không loại bỏ giá trị đó
- Không cần thiết phải thực hiện với các pointers như trong hình minh họa

```

1 queue1 = MyQueue(capacity=5)
2
3 queue1.enqueue(1)
4
5 queue1.enqueue(2)
6
7 print(queue1.isFull())
8 >> False
9
10 print(queue1.front())
11 >> 1
12
13 print(queue1.dequeue())
14 >> 1
15
16 print(queue1.front())
17 >> 2
18
19 print(queue1.dequeue())
20 >> 2
21
22 print(queue1.isEmpty())
23 >> True

```