WIA1002 Data Structure

Lab Test 2 Set A (2 hours including 30mins grace period). Thursday 12pm-2pm Stack

Submission Link:

https://docs.google.com/forms/d/e/1FAIpQLSen-YZvmcKT9BLPQjCaUKj-4hSKnFwuKr5gpbISfNl6q9QxlQ/viewform?usp=sf_link

COVID-19 Vaccine Distribution System/Algorithm

This program is expected to create a stack to store and sort people for the COVID-19 vaccination program.

Part A: Create an unsorted list of people who have registered for the COVID-19 vaccine program.

1. Create a class called Person. This class stores information as follows: name, age, gender, occupation, occupation_list and frontliner. The last variable is of type Boolean to determine whether the object of class Person created is a frontliner or not. The list occupation_list is a list of occupations that are considered as frontliners which are doctocar, nurses, teacher and police.
2. An object of class person must be initialised with at least name and age. Create multiple constructors to cater for when the input has more parameters.
3. Create a class called Stack . You can re-use the codes from Question 1 of Lab6. Modify the codes so that Stack can store objects of type Person.
4. Class Stack has a variable called stack_name which is a required parameter at the point of the creation of Stack. Whenever we want to display the stack, the stack name is displayed first followed by the elements inside the stack (see Input and Expected Output).
5. Modify the push() method in Class Stack so that when parameters such as name and age are passed to the methods, the method automatically create an object person which stores such information. See Input for visualization
6. Overload push() method so that it can create and push a Person object that takes more parameters. For example stack.push(“adam”,25,”m”) and stack.push(“adam”,25,”m”,”fireman”) should both be possible. If occupation parameter is given, set frontliner to true if the occupation is found in the occupation_list. You should have a method called setFrontliner().
7. Overload push() method so that it can push a person object. See ‘katie’ example in Input.
8. Create a method inside Class Stack called displayStack() that display the elements of the stack as what stack should be i.e element pushed first should be at the bottom of the stack. For each element, print out the ArrayList index, name, age, gender and whether the person is a frontliner or not given his/her occupation. If the stack is empty, the stack display “Empty stack”. See Expected Output for visualization for Part A and Expected Output for part B.
9. Implement the input below in your main method and verify that you have successfully produced the expected output.

Input for Part A.

```
   public static void main(String[] args) {
 Stack raw = new Stack("Raw list");
   raw.push("ali",32,"m","doctor");
   raw.push("lisa",29,"f","nurse");
   raw.push("tanaka",41);
   raw.push("ahmad",18,"m","developer");
   raw.push("maria",20,"f","accountant");
   raw.push("chong",70,"m","lawyer");
   raw.push("raju",55,"m","police");
   raw.push("alex",16,"f","business man ");
   Person katie = new Person("katie",36,"f","teacher");
   raw.push(katie);
   raw.displayStack();}
```

Expected Output for Part A- An unsorted list of people who have registered for the vaccine program. Most left bottom column index 0 indicates bottom of the stack which be the last element taken out.

```
----Raw list----
8 katie 36 f Frontliner
7 alex 16 f Not frontliner
6 raju 55 m Frontliner
5 chong 70 m Not frontliner
4 maria 20 f Not frontliner
3 ahmad 18 m Not frontliner
2 tanaka 41 null Not frontliner
1 lisa 29 f Frontliner
0 ali 32 m Frontliner
```

Part B: Separate the elements of the unsorted list into two stacks, frontliners and others.

1. Create two other stacks called frontliners and others.
2. In the main program, create a loop to pop out all persons from raw list and push them into either frontliners or others, depending on whether they are frontliners or not. See Expected Output for Part B. Do not manually pop and push the persons.
3. At the end of Part B, raw list should be empty. Both frontliners and others list are unsorted.

Expected Output for Part B:

```
~~~~~~~Sort into frontliners and others~~~~~

----Raw list----

Empty stack

--------------------

----Frontliners----

3 ali 32 m Frontliner

2 lisa 29 f Frontliner

1 raju 55 m Frontliner

0 katie 36 f Frontliner

--------------------

----Others----

4 tanaka 41 null Not frontliner

3 ahmad 18 m Not frontliner

2 maria 20 f Not frontliner

1 chong 70 m Not frontliner

0 alex 16 f Not frontliner

--------------------
```

Part C: Create a new sorted list of persons to be Vaccinated.

1. Create a new stack called 'priority' with the name of the stack set as 'Vaccine Priority List'. Persons are pushed into this stack according to priority. Do not push and pop manually to complete Part C.
2. From the stack of frontliners and others in Part B, pop and push the elements so that young and non frontliners should be at the bottom of the stack as they will be called last, old people and work as frontliners should be at the top of the stack. See Expected Output for Part C for visualization.
3. To complete Part C, you may need to create some extra (static) methods.

Expected Output for Part C

```
~~~~~~~Sort into Vaccine list~~~~~
----Others----
Empty stack
-------------------
----Frontliners----
Empty stack
-------------------
----Vaccine Priority List----
8 raju 55 m Frontliner
7 katie 36 f Frontliner
6 ali 32 m Frontliner
5 lisa 29 f Frontliner
4 chong 70 m Not frontliner
3 tanaka 41 null Not frontliner
2 maria 20 f Not frontliner
1 ahmad 18 m Not frontliner
0 alex 16 f Not frontliner
```

Marks allocation

Completeness of program 3%

Meaningful comments 1%

First 50% of student submission 1%