Vaccination Application

Your local Vaccination Center requires an application to manage clients and their vaccinations. Clients have a name, id, phone number and vaccination details.

The application is only currently required to add clients and to record their first vaccination. It should record the name and the date of the first vaccine dose received.

Depending on the type of vaccine given a second dose may or may not be needed.

The application should give some statistics about the percentage of people who have been given each vaccine type and also show which clients are due their second vaccine dose.

There are 4 different vaccines at the moment each with an associated id

1. Pfizer 2 dose. (4 weeks apart)
2. Astra-Zenica 2 dose (8 weeks apart)
3. Moderna 2 dose (4 weeks apart)
4. Johnson&Johnson 1 dose -

For all vaccines given to a client the following information is recorded:

- type (1-4)
- company name
- batch number
- completion status
  The completion status is set to true if no further doses are required. If a Johnson&Johnson vaccine is given the completion status is immediately set true.

For vaccines that require 2 doses the number of weeks required before the second does must also be recorded. For future application development there is also the potential to record the date of a second dose.

Each class written should have getters and setters for each field and a toString method.

Java Classes Required

1. A **Name** class.
   This class stores details of a persons information:

Attributes

```
firstname: String
lastName:  String
```

Methods

getters and setters
toString

2. A **Client** class.
   This is a class to keep track of the details of the people attending for a vaccine:

Attributes

```
name:       Name
id:         String
phone:      String
vaccine:    BasicVaccine
```

Methods

getters and setters
toString

3. A **VaccineCentre** class. This contains the name and location of the vaccine centre and a list of Clients.

Attributes

```
name_of_centre:       String
eircode:              String
client_list:          List of Clients
```

Methods

getters and setters

```
add_client:           Add a Client to the list
remove_client:        Remove a Client based on id
find_client:          Find a Client based on an id
show_all_clients:     List the name and vaccine details for all
                      clients
show_stats:           Show the % of clients that have been given each
                      of the vaccine types
show_clients_due:     List the names of clients that are due a second
                      vaccine – showing the type of vaccine and date it
                      is due.
load_client_data:     This loads a list of clients including vaccine
                      information for each client from a text file
save_client_data:     This saves the data relating to the  list of
                      clients to a text file.
```

4. A **BasicVaccine** class. This contains the information that is common to all vaccine types.

Attributes

```
company_name :        String
company_id:           int 1..4
batch_number:         int
vaccine_date:         Date
completion_status:    boolean
```

Methods

```
getters and setter
toString
get_next_dose_date:   returns a date object or null if no other    does
                      is required.
```

5. A **TwoDoseVaccine** class.

This is used for any vaccine that needs to store information about a second dose. This class is a subclass of **BasicVaccine**. It stores additional details about the the second dose.

Attributes

```
min_weeks_between_dosed :        int
actual_second_dose_date:         date
```

Methods

```
getters and setter
toString
get_next_dose_date():  Returns the date that the next vaccine is
                     due.
```

Design

You are to create a UML diagram which details all the java classes used in your application and the relationship between them.

Test Code

As you write each class you should implement tests for that class in your test class. Once all the classes are written you should have a test class which (via hard coding) completes the following:

- Creates at least 3 Vaccine objects. (either of type BasicVaccine for 1 dose vaccines or of type TwoDoseVaccine for those requiring 2 doses)
- Creates a VaccineCentre class
- Creates clients and assigns them a vaccine
- Adds clients to the vaccine centre
- Display all clients followed by their vaccine info and the date the second vaccination is due.

Appliation Code

You should also add the following functionality through a menu to your Test class.

This will be a command line program that displays the following menu options:

Cork Vaccination Center

==========

1. Add a Client for this Vaccine Centre

2. Remove a Client at this Vaccine Center

3. Record a Clients First Vaccination

4. Display details of all Clients (including their vaccine details)

5. Display overall vaccine stats (the percent of clients given each vaccine type)

6. Show the names of clients due a second vaccine and the date due.

7. Load Client Information from a text file

8. Save Client information to a text file.

9. Quit

Enter choice:>>:

Samples Files

Here is a possible format for the files:

```
Hall,David,127,08612345,4,Johnson&Johnson,546123,2020-12-04
Wall,Ann,128,08687654,1,Pfizer,123456,2020-04-14,4,-
Dawes,Fred,129,08687654,2,AstraZenica,876123,2021-05-04,8,-
Dawes,Mary,130,086871234,2,AstraZenica,876987,2021-05-30,8,-
```

This contains the following comma separated fields for all vaccine types

```
surname
first name
client id
phone number
vaccine id
vaccine company name,
vaccing batch no.
the date the first dose was administered or a –
```

Another two fields are stored for two dose vaccines:

```
number of weeks required before the second dose or  a -
the date the second dose was administered or a -
```