Classes Challenge 39:
Epidemic Outbreak Terminal App

Description:
You will be responsible for writing a program that simulates the spread of an infectious disease throughout a population. Using classes, you will model an individual person's and an entire population's attributes and behaviors. Your program will allow users to set various initial conditions in regards to the infection such as population size, infection rate, mortality rate, and infection duration. The program will then simulate the interaction of people within a population and spread the disease. Each iteration of spreading the disease will result in a summary displaying statistics of the population.

Step By Step Guide:
Defining your classes

- Define a class Simulation
- Define a class Person
- Define a class Population

Defining your class methods
The Simulation Class

- Define an __init__() method for the Simulation class, which takes one parameter, the required self parameter.
    - Initialize an attribute called day_number and set it equal to 1.

    - Print a message about needing to know population size.
    - Get user input for the population size.
    - Store the response in an attribute called population_size.

    - Print a message about needing to know the percentage of the population initially infected.
    - Get user input for the starting infection size.
        - Divide this value by 100 to turn it into a percentage.
    - Store the response in an attribute called infection_percent.

    - Print a message about needing to know the probability that a person will get infected if they come in contact with the disease.
    - Get user input for the infection probability.
    - Store the response in an attribute called infection_probability.

    - Print a message about needing to know how long the infection will last.
    - Get user input for the infection duration.
    - Store the response in an attribute called infection_duration.

    - Print a message about needing to know the mortality rate of those infected.
    - Get user input for the mortality rate.
    - Store the response in an attribute called mortality_rate.

    - Print a message about needing to know how long to run the simulation for.
    - Get user input for the number of days to simulate.
    - Store the response in an attribute called sim_days.

The Person Class

- Define an __init__() method for the Person class which takes one parameter, the required self parameter.
    - Initialize an attribute called is_infected and set it equal to False.
    - Initialize an attribute called is_dead and set it equal to False.
    - Initialize an attribute called days_infected and set it equal to zero.

- Define an infect() method for the Person class which takes two parameters, the required self parameter, and a simulation object.
    - Create a random integer from 0 to 100 to represent the chance this individual person becomes infected.
        - Type import random as the first line of code in your program.
    - If this value if less that the simulations infection_probability attribute:
        - Infect the person by setting their is_infected attribute to True.

- Define a heal() method for the Person class which takes one parameter, the required self parameter.
    - Set the persons is_infected attribute to False.
    - Set the persons days_infected attribute to 0.

- Define a die() method for the Person class which takes one parameter, the required self parameter.
    - Set the persons is_dead attribute to True.

- Define an update() method for the Person class which takes two parameters, the required self parameter and a simulation object.
    - First, check that the person is not dead. If the person is not dead:
        - Check if they are infected. If the person is infected:
            - Increase their days_infected attribute by 1.
            - Create a random integer from 0 to 100.
            - If this integer is less than the simulations mortality_rate attribute:
                - Call the persons die() method to kill the person.
            - Elif, the person didn't die, check to see if they can be healed. If the persons days_infected attribute is equal to the simulations infection_duration attribute:
                - Call the persons heal() method to heal the person.

The Population Class

- Define an __init__() method for the Population class, which takes two parameters, the required self parameter, and a simulation object.
    - Initialize an attribute called population and set it equal to a blank list.
    - Now, you must create the correct number of Person objects and store them in your Population object. To accomplish this use a for loop.
    - Loop until you have reached the population size, which is stored in the simulation objects population_size attribute. Each iteration of the loop you should:
        - Create a Person object.
        - Append the Person to the Populations population attribute.

- Define an initial_infection() method for the Population class, which takes two parameters, the required self parameter, and a simulation object.
    - Create a variable called infected_count. This will represent the number of people who must start infected based on the user's initial conditions.
    - To determine this value, multiply the simulation objects infection_percent and population_size attributes together.
        - Round this value to zero decimals.
        - Cast this value to an integer so we can use it in a for loop.
    - Use a for loop. Loop until you have infected infected_count people in the population. Each iteration of the loop you should:
        - Set the is_infected attribute to True for the current person in the population.
            - Recall the Population object has an attribute population which is a list. Each element in the list represents a Person object. Therefore, self.population[i] represents a Person object.
        - Set the days_infected attribute to 1 for the current Person in the population.
    - Now that you have initially infected the correct number of people, you need to spread them through the list. They are currently all located at the start of the list. This is bad because our simulation will spread the infection by looking at adjacent Person objects in the population list. Therefore, we need to shuffle the list.
        - Use the random module to shuffle the list.

- Define a spread_infection() method for the Population class which takes two parameters, the required self parameter, and a simulation object.
    - Use a for loop to loop through the length of the population list.
    - For example, for i in range(len(self.population)):
    - Recall that an individual Person will be represented as self.population[i].
        - If the is_dead attribute is False for the current person in the population, meaning they are currently alive, we will check to spread the infection.
            - If i is equal to 0:
                - This is the first Person in the population list, so we can only check to see if the Person to the right of them is infected.
                - If the Person object in the next index, i+1, is infected:

- Call the current Persons infect() method.
- Elif i is less than the length of the population list minus 1.
  - These Person objects are in the middle of the list so we will check both to the right and left of them.
  - If the Person object in the next index, i+1, is infected or if the Person object in the previous index, i-1, is infected:
    - Call the Persons infect() method.
- Elif i is equal to the length of the population list minus 1.
  - This Person is the last Person in the list, so we can only check to see if the Person to the left of them is infected.
  - If the Person object in the previous index, i-1, is infected:
    - Call the Persons infect() method.

- Define an update() method for the Population class which takes two parameters, the required self parameter, and a simulation object.
  - Increase the simulation objects day_number attribute by 1.
  - Loop through the population list. For each iteration you should:
    - Call the Persons update() method.

- Define a display_statistics() method which takes two parameters, the required self parameter, and a simulation object.
  - Create a variable total_infected_count and set it equal to 0.
  - Create a variable total_death_count and set it equal to 0.
  - Loop through the population list. For each iteration you should:
    - Check if the current Person is infected. If they are:
      - Increment total_infected_count by 1.
      - Check if the current Person is dead. If they are:
        - Increment total_death_count by 1.
  - Create a variable infected_percent and calculate the percentage of the population that is infected.
    - Round this value to 4 decimals.
  - Create a variable death_percent and calculate the percentage of the population that is dead.
    - Round this value to 4 decimals.
  - Print a summary of the population statistics for the current day of the simulation.
    - This should include the day number.
    - The percentage of the population infected.
    - The percentage of the population that is dead.
    - The total number of people infected.
    - The total number of deaths that have occured.
    - See example output for formatting.

- Define a graphics() method for the Population class which takes one parameter, the required self parameter.
  - Create a blank list called status.

- Loop through the population list. For each iteration you should:
  - Check if the person is dead. If they are:
    - Set a variable char equal to the letter X.
  - Else the person is alive.
    - If the person is infected:
      - Set a variable char equal to the letter I.
    - Else:
      - Set a variable char equal to the letter O
  - Append the current value of char to the list status.
- Loop through the list status. For each iteration of the loop:
  - Print the current value using the end=’-’ argument to keep the elements on one line separated by a ‘-’. Your list should look something like this: O-O-I-I-X-O-O-X-O-O-I.
    - O represents a healthy person.
    - I represents an infected person.
    - X represents a dead person.

The main code

- Create a Simulation object.
- Create a Population object.

- Call the Populations initial_infection() method.
- Call the Populations display_statistics() method.
- Call the Populations graphics() method.
- Prompt the user to press enter to being the simulation.

- Use a for loop to simulate each day of the simulation. For each iteration you should:
  - Spread the infection by calling the Populations spread_infection() method.
  - Update the population by calling the Populations update() method.
  - Display the statistics for the current day by calling the Populations display_statistics() method.
  - Show the graphics of the data by calling the Populations graphics() method.
  - If you are currently not on the last day of the simulation:
    - Prompt the user to press enter to advance to the next day of the simulation.

Example Output:

```
To simulate an epidemic outbreak, we must know the population size.
---Enter the population size: 100

We must first start by infecting a portion of the population.
--Enter the percentage (0-100) of the population to initially infect: 12
```

```
We must know the risk a person has to contract the disease when exposed.
--Enter the probability (0-100) that a person gets infected when exposed to the disease: 25

We must know how long the infection will last when exposed.
--Enter the duration (in days) of the infection: 4

we must know the mortality rate of those infected.
--Enter the mortality rate (0-100) of the infection: 35

We must know how long to run the simulation.
--Enter the number of days to simulate: 10

-----Day # 1-----
Percentage of Population Infected: 12.0%
Percentage of Population Dead: 0.0%
Total People Infected: 12 / 100
Total Deaths: 0 / 100
O-O-O-I-O-O-I-O-O-O-O-O-O-O-O-O-O-O-O-O-O-O-O-O-O-O-O-O-O-I-O-O-O-O-I-O-O-I-O-O-
O-O-O-O-O-O-O-O-I-O-I-O-O-O-O-O-O-O-O-O-O-O-O-O-O-O-O-O-O-O-O-O-O-I-O-I-O-O-O-I-
O-O-O-O-O-O-O-O-O-O-O-I-I-O-O-O-O-O-O-O-
Press enter to begin the simulation.

-----Day # 2-----
Percentage of Population Infected: 18.0%
Percentage of Population Dead: 6.0%
Total People Infected: 18 / 100
Total Deaths: 6 / 100
O-O-I-I-I-O-X-O-O-O-O-O-O-O-O-O-O-O-O-O-O-O-O-O-O-O-O-O-O-X-I-O-O-O-I-O-O-I-O-O-O-
O-O-O-O-O-O-O-I-X-X-O-O-O-O-O-O-O-O-O-O-O-O-O-O-O-O-O-O-O-O-O-I-I-O-I-O-O-O-I-O-O
-O-O-O-O-O-O-O-O-X-X-I-O-O-O-O-O-O-O-
Press enter to advance to the next day.

-----Day # 3-----
Percentage of Population Infected: 21.0%
Percentage of Population Dead: 11.0%
Total People Infected: 21 / 100
Total Deaths: 11 / 100
O-O-I-I-I-X-X-O-O-O-O-O-O-O-O-O-O-O-O-O-O-O-O-O-O-O-O-O-O-X-I-O-O-O-I-O-I-X-O-O-O-
O-O-O-O-O-O-O-I-X-X-O-O-O-O-O-O-O-O-O-O-O-O-O-O-O-O-O-O-O-O-O-I-I-O-I-O-O-O-X-X-
O-O-O-O-O-O-O-O-O-X-X-X-O-O-O-O-O-O-O-
Press enter to advance to the next day.

-----Day # 4-----
Percentage of Population Infected: 24.0%
Percentage of Population Dead: 17.0%
```

```
Total People Infected: 24 / 100
Total Deaths: 17 / 100
O-O-I-O-I-X-X-O-O-O-O-O-O-O-O-O-O-O-O-O-O-O-O-O-O-O-O-O-O-X-I-X-X-I-X-O-I-X-X-O-O-
O-O-O-O-O-O-O-O-X-X-I-O-O-O-O-O-O-O-O-O-O-O-O-O-O-O-O-O-O-O-O-X-X-O-O-O-O-O-X-
X-O-O-O-O-O-O-O-O-O-X-X-X-I-O-O-O-O-O-O-
Press enter to advance to the next day.

-----Day # 5-----
Percentage of Population Infected: 29.0%
Percentage of Population Dead: 21.0%
Total People Infected: 29 / 100
Total Deaths: 21 / 100
O-I-O-O-X-X-X-O-O-O-O-O-O-O-O-O-O-O-O-O-O-O-O-O-O-O-O-O-O-X-O-X-X-I-X-O-X-X-X-I-X
-X-O-O-O-O-O-O-O-X-X-I-I-O-O-O-O-O-O-O-O-O-O-O-O-O-O-O-O-O-O-I-X-X-O-O-O-O-I-X-X-
O-O-O-O-O-O-O-O-O-X-X-X-I-O-O-O-O-O-O-
Press enter to advance to the next day.

-----Day # 6-----
Percentage of Population Infected: 33.0%
Percentage of Population Dead: 26.0%
Total People Infected: 33 / 100
Total Deaths: 26 / 100
X-I-O-X-X-X-X-O-O-O-O-O-O-O-O-O-O-O-O-O-O-O-O-O-O-O-O-O-O-X-I-X-X-I-X-O-X-X-X-X-X-
X-O-O-O-O-O-O-X-X-X-I-I-O-O-O-O-O-O-O-O-O-O-O-O-O-O-O-O-O-O-X-X-X-O-O-O-O-I-X-X-
O-O-O-O-O-O-O-O-O-X-X-X-I-O-O-O-O-O-O-
Press enter to advance to the next day.

-----Day # 7-----
Percentage of Population Infected: 34.0%
Percentage of Population Dead: 29.0%
Total People Infected: 34 / 100
Total Deaths: 29 / 100
X-I-O-X-X-X-X-O-O-O-O-O-O-O-O-O-O-O-O-O-O-O-O-O-O-O-O-O-O-X-I-X-X-O-X-O-X-X-X-X-
X-X-O-O-O-O-O-I-X-X-X-X-I-O-O-O-O-O-O-O-O-O-O-O-O-O-O-O-O-O-O-X-X-X-O-O-O-O-X-X-
X-O-O-O-O-O-O-O-O-I-X-X-X-X-O-O-O-O-O-O-
Press enter to advance to the next day.

-----Day # 8-----
Percentage of Population Infected: 38.0%
Percentage of Population Dead: 32.0%
Total People Infected: 38 / 100
Total Deaths: 32 / 100
X-O-O-X-X-X-X-O-O-O-O-O-O-O-O-O-O-O-O-O-O-O-O-O-O-O-O-O-O-X-I-X-X-O-X-I-X-X-X-X-
X-X-X-I-O-O-O-X-X-X-X-X-X-O-O-O-O-O-O-O-O-O-O-O-O-O-O-O-O-O-I-X-X-X-O-O-O-O-X-X-X
-O-O-O-O-O-O-O-I-I-X-X-X-X-O-O-O-O-O-O-
```

```
Press enter to advance to the next day.

-----Day # 9-----
Percentage of Population Infected: 41.0%
Percentage of Population Dead: 37.0%
Total People Infected: 41 / 100
Total Deaths: 37 / 100
X-O-X-X-X-X-X-O-O-O-O-O-O-O-O-O-O-O-O-O-O-O-O-O-O-O-O-O-O-X-X-X-X-O-X-I-X-X-X-X-
X-X-X-X-O-O-O-X-X-X-X-X-X-O-O-O-O-O-O-O-O-O-O-O-O-O-O-O-O-X-I-X-X-X-O-O-O-O-X-X-
X-O-O-O-O-O-O-O-X-I-X-X-X-X-I-O-O-O-O-O-
Press enter to advance to the next day.

-----Day # 10-----
Percentage of Population Infected: 44.0%
Percentage of Population Dead: 40.0%
Total People Infected: 44 / 100
Total Deaths: 40 / 100
X-I-X-X-X-X-X-O-O-O-O-O-O-O-O-O-O-O-O-O-O-O-O-O-O-O-O-O-O-X-X-X-X-I-X-X-X-X-X-X-X
-X-X-X-O-O-O-X-X-X-X-X-X-O-O-O-O-O-O-O-O-O-O-O-O-O-O-O-O-X-X-X-X-X-X-O-O-O-X-X-X
-O-O-O-O-O-O-I-X-O-X-X-X-X-I-O-O-O-O-O-
```