Graded Project: PetCare Scheduler

Estimated Duration: 90-120 minutes

Prerequisites

To complete this project successfully, you must have fundamental and object-oriented Java skills. If you do not already have these skills, you can acquire them when you complete the first two courses for fundamental Java programming skills and object-oriented Java programming skills that are part of this program.

You must have also completed this course's EcoPoints Recycling Tracker guided practice project.

Learning Objectives

After completing this project, you will be able to demonstrate your ability to:

- Apply object-oriented programming principles to design Pet and Appointment classes using encapsulation
- Construct and manipulate ArrayList or HashMap collections to manage multiple pet records and appointments
- Use Java's Date and Time API to record and compare dates for pet registration and appointments
- Implement Java File I/O operations to persist and retrieve pet and appointment data from files
- Handle invalid inputs and runtime exceptions gracefully using Java's exception handling mechanisms
- Develop conditional logic and looping constructs to navigate menu options and display various appointment histories and reports
- Generate simple reports from stored data to show upcoming and overdue appointments

Your completed project will demonstrate that you can work with:

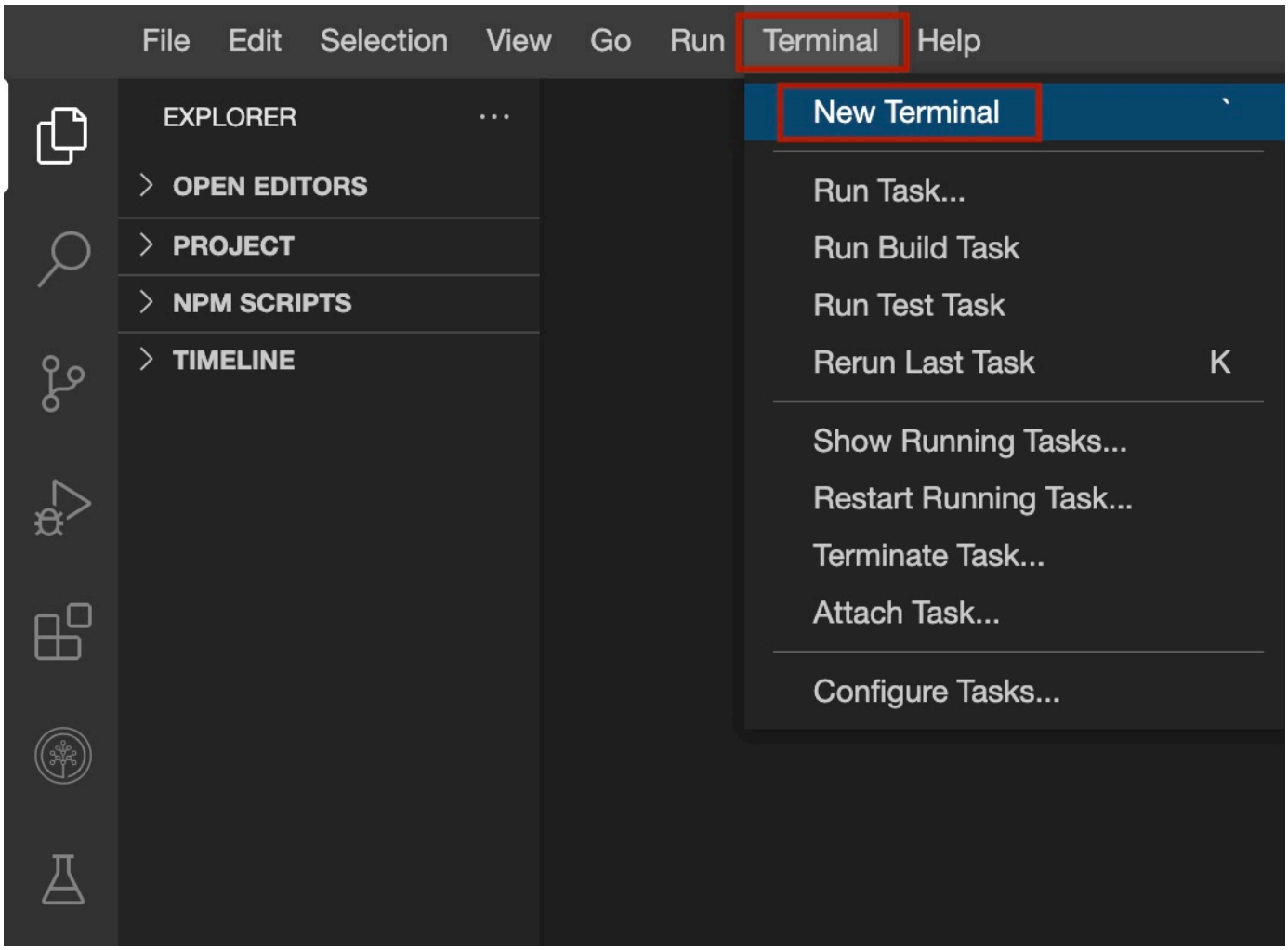
- Java programming basics
- Strings and string operations
- Operators and data types
- Exceptions
- `for` Loops and the `while` Loop
- Conditional statements
- Arrays, Sets, Maps
- Basic methods and functions
- Object-Oriented Programming (OOP) concepts
    - Encapsulation
    - Overriding
    - Polymorphism

You will complete this project in Cloud IDE, which has Java preinstalled. You can choose to complete this project in your own IDE if you have Java preinstalled. You will not be provided any code in this graded project.

Please apply the skills you learned in the prerequisite courses and this course's guided project, to complete this project. This project does not require any additional knowledge beyond the prior courses and the practice project for this course.

Initial setup

1. Open a new terminal.



2. Create a directory name `PetCareScheduler` under `/home/project` by running the following command in the terminal.

```
mkdir PetCareScheduler
```

3. Change to the project directory.

```
cd PetCareScheduler
```

The object blueprint classes

1. Create a new file named `Appointment.java`. You will create this first as the `Pet` class will include details of appointments.

```
touch Appointment.java
```

2. Click the following button to open the file for editing.

Open **Appointment.java** in IDE

3. The `Appointment` class should have the following attributes:

    - Appointment type (for example, vet visit, vaccination, grooming)
    - Date and time
    - Notes (optional)

4. Provide setters, getters as appropriate. Override the toString method to provide the string format of the `Appointment` object.

5. Create a new file named `Pet.java`.

```
touch Pet.java
```

6. Click the following button to open the file for editing.

Open **Pet.java** in IDE

7. The `Pet` class should have the following attributes.

    - Unique Pet ID
    - Name
    - Species/Breed
    - Age
    - Owner name
    - Contact info
    - Registration date
    - List of appointments

8. Provide setters, getters as appropriate.

The main application

1. Create a new file named `PetCareScheduler.java` where your console application is going to be.

```
touch PetCareScheduler.java
```

2. Open the file and paste the following content in the file to create the class.

Open **PetCareScheduler.java** in IDE

3. Add the required import statements.

4. Add the class structure inside, which you will be adding the code for the application in the main method.

5. Instantiate the appropriate class to read from console.

6. Create an appropriate collection object to collect all the `Pet` objects and all the `Appointment` objects.

7. Add the main method for the class.

    - Load the existing data from files, if any.
    - Obtain input from the user to perform the following operations as long as the user intends to:
        - Register pets
        - Schedule appointments
        - Store data
        - Display records
        - Generate reports
    - Verify that any possible exceptions and errors are handled diligently.

Add the methods to handle user input

1. Add a method to `PetCareScheduler.java` after the `main` method. This method should register a new pet and add the pet to the collection of pets maintained.

2. Add a method to `PetCareScheduler.java`. This method should allow you to add an appointment for a pet. The appointment should be added to the collection of appointments for the pet. Verify the following validations are completed:

    - Pet ID is validated and the appointment is only set for pets that exist.
    - The appointment is set for a valid future date and time.
    - Validate the appointment type.

3. Add a method to `PetCareScheduler.java` to store the data about the pets and the appointments to a file.

4. Add a method to `PetCareScheduler.java` to display the following details:

    - All registered pets
    - All appointments for a specific pet
    - Upcoming appointments for all pets
    - Past appointment history for each pet

5. Add a method to `PetCareScheduler.java` to generate a report regarding the following details:

    - Pets with upcoming appointments in the next week
    - Pets overdue for a vet visit (such as, no vet visit in the last 6 months)

6. Add code to main method to take user input and based on the input invoke one of the actions, by invoking the methods that have been created and implemented.

    Hint: Use switch-case

Compile and run

1. Now that you have assimilated all the code together, you can compile and run the files. Verify that you are still in the project directory.

    ```
    cd /home/project/PetCareScheduler
    ```

2. Compile all the classes by running the following method.

    ```
    javac *.java
    ```

3. Run the application, by running the following command.

    ```
    java PetCareScheduler
    ```

Checklist for tasks

At this point, all of these tasks should be accomplished.

Task 1: The Object blueprint classes
You created the Pet and Appointment Classes.

Task 2: You created the main application that loads all the data from the file, if any and takes the user input and does the appropriate action.

Task 3: You added all the methods to handle user input, that allows to:

- Register the pet
- Schedule an ppointment
- Display the records
- Store data
- Generate reports

Task 4: You compiled and run the code and ensured that the application is functioning as intended.

Evaluation

Now that you have completed the project, verify you provided the right code for it to be evaluated.

Author(s)

Lavanya