SE116 – LAB#6 2023-2024 SPRING

Aim: Introduction to the concept of run-time polymorphism (run-time binding); understanding the use of abstract classes with abstract methods.

1. Suppose that you are trying to complete your assignment, particularly for your SE116 course. You want to get a high grade from this assignment. If you complete the whole work before the deadline, you will have no problem. However, while studying, you may be exhausted and may fall asleep.

You have a total of 12 hours left to submit your assignment and you have limited energy. There are three certain actions which will affect your energy, time and completion percentage of your assignment:

- When you study, you complete a certain portion of your assignment on one side; on the other side, you get tired. If you get so tired that you have no energy left to study; you will fall asleep and cannot continue doing your assignment.
- If you get tired, you can get rest for a certain amount of time to increase your energy.
- When you are tired, you can also eat something to get energy.

Create an application that simulates the situation with the following steps:

Define an `abstract` base class `Activity`. This class will have three protected class variables; an `int completionPercentage`, a `double timeLeft` and an `int energy`. Initialize `completionPercentage` as 0, `timeLeft` as 12.0 and `energy` as 100 in your no-argument (non-parametrized) constructor. The class has also appropriate getters for data members and has appropriate constructor(s) to initialize the data members. Add an `abstract completed()` method to this class which will be defined later in concrete subclasses.

2. Derive three concrete classes from the base class `Activity`: `Study`, `Rest` and `Eat`. Implement `completed()` method for each derived class by defining the following actions:

- For the class `Study`, decrease `timeLeft` by 1.5 hours, decrease `energy` by 25, increase `completionPercentage` by 15.
- For the class `Rest`, decrease `timeLeft` by 1 hour and increase `energy` by 25. Resting doesn't affect the `completionPercentage`.
- For the class `Eat`, decrease `timeLeft` by 0.5 hours and increase `energy` by 10. Eating doesn't affect the `completionPercentage`.

In your `main` method of your `Test` class, create appropriate objects for your purpose. At first, you can display the current values of `completionPercentage`, `timeLeft` and `energy` to the user. Then, in a loop; ask the user to select an action (studying, resting, or eating). According to the action of the user, the program will call the corresponding `completed()` method on the corresponding object that you have created in the `main` method. The program will be terminated if `timeLeft` or `energy` values have been depleted, or if the user completes the assignment successfully. Finally, display `completionPercentage` of the assignment to the user.