logo, yazı tipi, simge, sembol, grafik içeren bir resim

Açıklama otomatik olarak oluşturuldu

**ANKARA SCIENCE UNIVERSITY**

**2024-2025 / SPRING**

**SENG 324 / Software Design Patterns**

**TERM PROJECT**

* You are given a detailed problem description for a software application that involves the use of multiple design patterns. Examine the descriptions carefully and implement the requested application in Java. You should submit the full source code as a zipped archive and a packaged application jar file (the so called ***fat jar file,*** that includes your implementation with all the external jar files packaged together) so that it can be executed easily using the command java -jar yourPackagedApp.jar.
* As indicated by the text below (in item 2) you are required to submit the UML class diagram describing the full set of patterns and other key classes in the application. The UML diagram will be submitted as a PDF or an image file.
* You are also given an example mock up GUI in Figure 1. Your application should be something similar but feel free to make it more elegant, plausible and feature rich. You will be given a JSON file that contains a list of city entities. Please use this file as an input to your application.

1. **Problem description:** You are developing a software for sorting different lists consisting of a sequence of **City** objects. Each City object has 5 Attributes These are name, population, area, currentTemperature, currentWeatherState. The CurrentWeatherState attribute can take one of (Sunny, Cloudy, rainy, snowy) as values. You will be provided by a list of cities in a Json file. Use this city list as a n input.
   * Your software should employ the **Singleton Design Pattern** to implement a repository of cities. In other words the singleton class should initialize itself with a list of cities reading a json file containing all the cities, and should make this list available to requesters from a single point.
   * You should design a Graphical User Interface that displays the list of cities in list box in different order depending on a sorting criteria selected by the user via a combo box. The combo box should present 3 sorting options: sort by name, sort by population, sort by area. Therefore, you should implement an extensible framework that uses the **Strategy Design Pattern**. As such, your software will include various sorting algorithms implemented as sort strategies.
   * The user interface should include another list box which should only list cities obeying a given weather condition such as rainy, sunny, cloudy or snowy. These criteria should be selectable from a combo box, and once selected the city list should change dynamically to reflect the selected criteria. You should implement this behaviour using **Iterator Design Pattern**. So you should have a 4 different iterators each for a different weather condition.
   * The user interface should include two charts: one for displaying the temperatures of all cities as a bar chart, and another one displaying a pie chart for percentage of cities with different weather conditions. IN other words the pie chart should contain the percentage of sunny, cloudy, rainy and snowy cities. Both of these charts should change dynamically upon receiving updates from a weather report provider object. This weather report provider object should be running on a separate thread and should update the weather information of cities randomly every 3 seconds. You should implement this behaviour using the **Observer Design Pattern** (also known as the **Publisher-Subscriber Design Pattern**).
   * Finally you should add a city activity planner to the application to plan activities such as visiting museums, shopping malls, parks, historic city center etc. You *don't* want to modify the City class directly.You *wrap* a City into new decorated versions offering extra "visit planning" features. So you should implement this functionality using the **Decorator Design Pattern**. You should add at least 4 decorators (MuseumVisit, ShoppingMallVisit, ParkVisit and CityCenterVisit). Each decorator should have its own cost and required time in hours. The planner should be available to the user via the GUI as a small pane where each activity can be added by clicking a check box when a city is selected from the sorted city list.
2. Please also provide a complete UML diagram representing all the design patterns involved in this problem.

A screenshot of a computer

AI-generated content may be incorrect.

Figure 1 An example GUI mock up