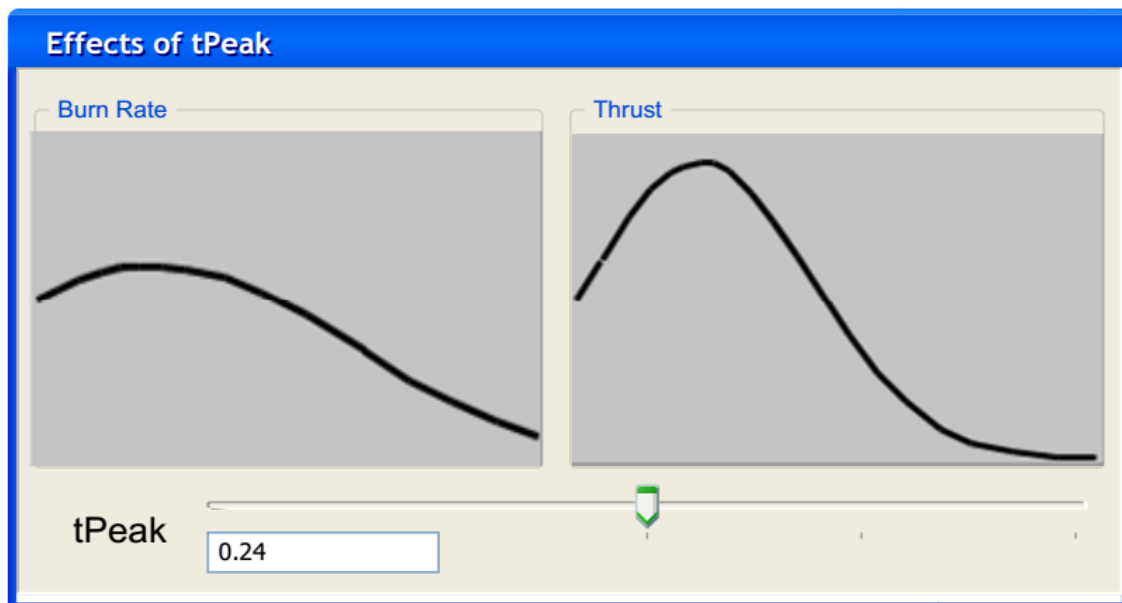


Practical Object oriented design (Mediator & Decorator)

Solve the following problems.

Problem1: Turbo Engine Ignition

You will need now to build for Audi an application (like the one shown in the figure below) that will give the possibility to the turbo ignition engine engineers to experiment visually with parameters that determine the relationships between the car thrust and the surface area of the ignited fuel.



In the new turbo engine that is under development when the engine ignites, the part of its fuel that is exposed to air burns producing thrust. From ignition to maximum burn rate, the burn area increases from the initial ignition area to the full surface area of the fuel. This maximum rate occurs at time t_{Peak} . As fuel burns off, the surface area reduces again until the fuel is consumed. Suppose that the burn rate and thrust equations are:

$$rate = 25^{-(t-t_{peak})^2} \quad thrust = 1.7 \left(\frac{rate}{0.6} \right)^{\frac{1}{0.3}}$$

The application from the figure above shows how t_{Peak} affects the burn rate and thrust of a car. As a user moves the slider, the value of t_{Peak} changes, and the curves take on new shapes. The value for t_{Peak} can be also entered numerically in the edit box below the slider. Design the application for above requirements?

Practical Object oriented design (Mediator & Decorator)

Problem2: War games

Imagine a war zone where armed units are moving into the enemy's territory. Armed units can include soldier, tank, grenadier, and sniper units. The strategy being employed is that whenever one unit attacks, other units should stop attacking and take cover. To do so, the unit that is currently attacking needs to notify the other units. Provide object oriented design to implement the above functionality.

Problem3: Clothing variations

The following program fragment implements variations in clothing by enumerating many possible combinations.

```
class Clothing { }  
class Dress extends Clothing { }  
class Costume extends Clothing { }  
class VestAndCostume extends Costume { }  
class TieAndCostume extends Costume { }  
class TieAndVestAndCostume extends Costume { }  
class BlouseAndDress extends Dress { }
```

The list is not exhaustive, many more combinations are possible, which gives an unmanageable and unmaintainable set of classes. How do you fix this by reorganising the class structure? Provide the modified design.

Problem4: HTML expressions

We wish to have a way to create an HTML expression composed of some plain text, and any combination (allows repeating) of following HTML tags: ``, `<a>`, `<u></u>`, `<i></i>`, ``. Both `<a>` and `` tags require extra information, namely a value for "href" and "style" parameters respectively. Design a simple application that allows us to create proper HTML expressions using the above tags only. Here is some sample expression that could be created:

```
<b><u><i><span href= "www.abc.com" style= "margin-left"><i>hello  
html</i></span></u></b>
```

Problem5: Logger

You have a simple logger class that is used as a utility class by various programmers that work on your project. This class logs a text string to a file. As your project grows you anticipate adding different types of behaviors to the logger class dynamically during runtime as different programmers require different types of additional functionality, e.g., they could add a timestamp to a log message, change the case of the message, translate

Practical Object oriented design (Mediator & Decorator)

the log message into a different language or any combinations of these behaviors. What design pattern is useful to model this requirement? Give the resulting design.

