A Class for Chemical Elements

https://csci-1301.github.io/about#authors

January 17, 2023 (03:24:19 PM)

Contents

1 A Class for Chemical Elements **1**
- 1.1 Reading . . . 1
- 1.2 Modifying . . . 2
- 1.3 Enhancing . . . 2

This lab serves multiple goals:

- To understand how to represent information in a class,
- To understand the design of a class involving static members,
- To convert between different representations without changing the stored values in the attributes.

1 A Class for Chemical Elements

In this lab you will study and modify a class for chemical elements[1]. Consult https://en.wikipedia.org/wiki/List_of_chemical_elements#List for a complete list of all elements.

1.1 Reading

Download ChemElemProject[2] and extract the project. Open it in in your IDE, compile and execute it. Now read the code in "ChemElem.cs" and "Program.cs".

The class definition "ChemElem.cs" contains:

- Three attributes,
- One constructor,
- One static method,
- One method that returns the melting point in Celsius,
- One `ToString` method.

The application program "Program.cs" performs one simple conversion from Kelvin to Celsius. It uses data given by the user to create and display a ChemElem object (implicitly calling the `ToString` method).

[1] https://en.wikipedia.org/wiki/Chemical_element
[2] labs/ChemicalElements/ChemicalElements.zip

1.2 Modifying

Do the following:

1. In "ChemElem.cs", write getters and setters for all attributes
2. In "ChemElem.cs", write a constructor that requires no arguments.
3. In "Program.cs", create a second object using the custom constructor that takes 3 arguments, then display the value of its attributes using the getters you previously defined.
4. In "Program.cs", create an object using the no-args constructor, and set its values using the setters your previously defined.
5. In "Program.cs", display on the screen the string returned by the `ToString` method when it is called by the object you created in the previous step.
6. In "Program.cs", try calling the `FromKelvinToCelsius` method with one of your objects, for instance using `hydrogen.FromKelvinToCelsius(34)`. What happens?
7. Still in "Program.cs", try calling the `MeltingInCelsius` method with the class, for instance using `ChemElem.MeltingInCelsius();`. What happens?

1.3 Enhancing

We now want to significantly improve this class, by adding:

1. An attribute for the boiling point,
2. All the tools needed to display the information in Fahrenheit degrees, in addition to Celsius and Kelvin.

You may want to comment out part or all of your "Program.cs" file, before starting to change your class.

- Add an attribute for the boiling point (in Kelvin).
- Modify the constructor, so that it takes a 4th argument, and sets its value to be the value of the boiling point attribute.
- Create a *static* `FromKelvinToFahrenheit` method, taking inspiration from the `FromKelvinToCelsius` method.
- Create a `MeltingInFahrenheit` method, that returns the melting point in Fahrenheit of the calling object. This method should use your `FromKelvinToFahrenheit` method.
- Create a `BoilingInFahrenheit` method, that return the boiling point in Fahrenheit of the calling object. This method should use your `FromKelvinToFahrenheit` method.
- Modify the `ToString` method, so that the string returned includes
  - The name of the chemical element and its atomic number,
  - The *melting point* in Kelvin and in Fahrenheit,
  - The *boiling point* in Kelvin and in Fahrenheit.

You should test all of those modifications in your "Program.cs" file as you implement them. Use relevant data, test your program, and make sure the behavior is the expected behavior.

Elements of solution

Typically, you need to replace

```
public ChemElem(int atomicNumberParam, string nameParam, decimal meltParam)
```

by

```
public ChemElem(int atomicNumberParam, string nameParam, decimal meltParam, decimal
↪  boilParam)
```

and to add

```
boil = boilParam;
```

to your constructor.

More subtle, the `FromKelvinToFahrenheit` method can be defined as follows and then re-used:

```
public static decimal FromKelvinToFahrenheit(decimal kelvinParam)
{
    return kelvinParam * 9/5 - 459.67M;
}
public decimal MeltingInFahrenheit()
{
    return FromKelvinToFahrenheit(melt);
}

public decimal BoilingInFahrenheit()
{
    return FromKelvinToFahrenheit(boil);
}
```