

Programming Assignment 12



Element

Overview

As you know, Chemicalc is a large computational chemistry system. The design phase has been completed and the implementation phase is continuing. You must implement the `Element` classes for this system.

You will not be developing the main class, nor will you have access to it.

Textual Description of Components

The following textual description was written and used in the design of the class you must implement.

Element

An element is a substance composed of atoms with the same number of protons but different numbers of neutrons. Each such atom is referred to as an isotope of the element.

An element is defined by its symbol (e.g., "Cl" for chlorine), its atomic number (e.g., 17 for chlorine), the mass number of all of its isotopes (e.g., 35 and 37 for chlorine), and the abundance of all of its isotopes (75.78% and 24.22% for chlorine).

Given the mass number and abundance of the isotopes, one can calculate the average mass (e.g., for chlorine the calculated average mass is $(0.7578 \cdot 35) + (0.2422 \cdot 37)$ or 35.4844). Note that the calculated average mass is different from the average mass that would be measured by a mass spectrometer and reported in the periodic table of the elements.

An element is referred to by its symbol.

Specifications

Element

`Element` objects must be immutable.

In addition, the `Element` class must comply with the following specifications:

1. It must have a private attribute named `calculatedAverageMass` that is a `double` containing the calculated average mass.
2. It must have a private attribute named `atomicNumber` that is an `int` containing the atomic number.
3. It must have a private attribute named `symbol` that is a `String` containing the symbol.
4. It must have an explicit value constructor with the following signature:

```
public Element(String symbol, int atomicNumber,  
               int[] mass, double[] abundance)
```

where `mass` contains the `mass` for all of the isotopes of the element and `abundance` is a number in the interval `[0.0, 100.0]` containing the abundance/percentage of each isotope. The `mass` and `abundance` arrays will be conformal; the constructor need not test to assure that it is the case.

5. It must have a method with the following signature:

```
public static boolean areIsotopes(Atom a, Atom b)
```

that returns `true` if and only if `a` and `b` are isotopes (i.e., have the same number of protons and a different number of neutrons).
6. It must have a method with the following signature:

```
public String toString()
```

This method must return the `String` representation of the `Element` (i.e., its symbol).

7. It must have the following accessors:

```
public int getAtomicNumber()  
public double getCalculatedAverageMass()
```

The `Element` class may have other private attributes and/or methods.

Existing Components

Your `Element` class must work with your existing `Atom` class.

Recommended Process

1. Read and understand the entire assignment.
2. Create a directory/folder for this assignment.
3. Copy `Atom.java` into the directory you created for this assignment. You may use either your implementation or the solution that was provided to you.
4. Think about the order in which you should implement the the methods in the `Element` class.
5. Implement and test (and debug, if necessary) the `Element` class one method at a time.
6. Submit `Atom.java` and `Element.java` in a file named `pa12.zip` using Autolab. Note that your submission will not compile against the official tests if it does not include both classes, and every method in them. Hence, if you are unable to complete a method, you should create a “stub” for it.

Grading

Your code will first be graded by Autolab and then by the Professor. The grade you receive from Autolab is the maximum grade that you can receive on the assignment.

Autolab Grading

Your code must compile (in Autolab, this will be indicated in the section on “Does your code compile?”), and all class names and method signatures comply with the specifications (in Autolab, this will be indicated in the section on “Do your class names, method signatures, etc. comply with the specifications?”) for you to receive any points on this assignment.

Autolab will then grade your submission as follows:

Conformance to the Course Style Guide: **20 points** (All or Nothing)

Correctness: **80 points** (Partial Credit Possible)

Manual Grading

After the due date, the Professor may manually review your code. At this time, points may be deducted for inelegant code, inappropriate variable names, bad comments, etc.