This exercise you will be persisting data by writing text to file and serialization to and deserialization using json.

# You must follow the specifications exactly (To demo to instructor beginning of next class)

# The Atom class

This is the front end of the application. There are 12 members.

|  |
| --- |
| Atom  Class |
| Fields |
|  |
| Properties |
| **+** «c# property, set» Name : **string**  **+** «c# property, set» Symbol : **string**  **+** «c# property, set» Proton : **int**  **+** «c# property, set» Neutron : **int**  **+** «c# property, set» Weight : **double** |
| Methods |
| **+** «c# constructor» Atom()  **+** «c# constructor» Atom(  **string** name,   **int** proton,   **int** neutron,   **double** weight,   **string** symbol)  **$+** Parse(**string** objectData) : Atom  **$+** GetAtoms() : List<Atom>  **+** ToString() : **double** |

#### Description of class members

##### Fields:

There are no fields in this class.

##### Properties:

All the properties have public get and set

**Name** – this is a string representing the name of this atom. Both the getter and setter are public.

**Symbol** – this is a two-letter string representing the name of this atom that is used in chemistry. Both the getter and setter are public.

**Proton** – this is an int representing the number of protons in the nucleus of this atom. Both the getter and setter are public.

**Neutron** – this is a int representing the number of neutrons in the nucleus of this atom. Both the getter and setter are public.

**Weight** – this is a double representing the atomic weight of this atom. Both the getter and setter are public.

##### Constructor:

There are two constructors for this class: a default one and a user defined one

**public** **Atom()** – This is a default constructor that is necessary for serialization. This is an empty constructor that does not do anything.

**public** **Atom**(**string** name, **int** proton, **int** neutron, **double** weight, **string** symbol) – This constructor assigns the arguments to the appropriate fields.

##### Methods

**public static Atom Parse(string line)** – This is a public class method that takes a string and returns an Atom object. The argument is one string that is comprised of the five fields of this object. It does the following:

* It uses the Split() method to parse the argument into five parts.

If you do not have five parts, the correct thing to do is throw an exception.

* I~~f the argument does not yield five parts then throw an exception~~.
* Otherwise create an Atom object and initialize the fields with the appropriate parts.
* Return the above object

You will need to examine the arguments to decide what part will be assign to which field.

**public override string ToString()** – This is a public method overrides the corresponding method in the object class to return a stringify form of the object.

### Test Harness

Insert the following code statements in your Program.cs file:

Write create two methods that will be called from Main. The first method will serialize a collection of atoms to a file and the second will deserialized the contents of a file to a list of atom.

**static void WriteJson(**List<Atom> **atoms, string filename)** – This void class method that takes the list of atoms and a filename. You will write the necessary code to serialize the first argument to the specified file.

**static void ReadJson(string filename)** – This void class method that takes a filename. You will write the necessary code to deserialize the file contents to a List<Atom> and then display all atoms.

Appendix

Code for the GetAtoms method of the Atom class

public static List<Atom> GetAtoms()

{

List< Atom > elements = new List< Atom >();

elements.Add(Atom.Parse("Hydrogen 1 0 1.0079 H"));

elements.Add(Atom.Parse("Helium 2 2 4.0026 He")); ;

elements.Add(Atom.Parse("Lithium 3 4 6.941 Li")); ;

elements.Add(Atom.Parse("Beryllium 4 5 9.0122 Be"));

elements.Add(Atom.Parse("Boron 5 6 10.811 B"));

elements.Add(Atom.Parse("Carbon 6 6 12.0107 C"));

elements.Add(Atom.Parse("Nitrogen 7 7 14.0067 N"));

elements.Add(Atom.Parse("Oxygen 8 8 15.9994 O"));

elements.Add(Atom.Parse("Fluorine 9 10 18.9984 F"));

elements.Add(Atom.Parse("Neon 10 10 20.1797 Ne"));

elements.Add(Atom.Parse("Sodium 11 12 22.9897 Na"));

elements.Add(Atom.Parse("Magnesium 12 12 24.305 Mg"));

elements.Add(Atom.Parse("Aluminum 13 14 26.9815 Al"));

elements.Add(Atom.Parse("Silicon 14 14 28.0855 Si"));

elements.Add(Atom.Parse("Phosphorus 15 16 30.9738 P"));

elements.Add(Atom.Parse("Sulfur 16 16 32.065 S"));

elements.Add(Atom.Parse("Chlorine 17 18 35.453 Cl"));

elements.Add(Atom.Parse("Potassium 19 20 39.0983 K"));

elements.Add(Atom.Parse("Argon 18 22 39.948 Ar"));

elements.Add(Atom.Parse("Calcium 20 20 40.078 Ca"));

elements.Add(Atom.Parse("Scandium 21 24 44.9559 Sc"));

elements.Add(Atom.Parse("Titanium 22 26 47.867 Ti"));

elements.Add(Atom.Parse("Vanadium 23 28 50.9415 V"));

elements.Add(Atom.Parse("Chromium 24 28 51.9961 Cr"));

elements.Add(Atom.Parse("Manganese 25 30 54.938 Mn"));

elements.Add(Atom.Parse("Iron 26 30 55.845 Fe"));

elements.Add(Atom.Parse("Nickel 28 31 58.6934 Ni"));

elements.Add(Atom.Parse("Cobalt 27 32 58.9332 Co"));

elements.Add(Atom.Parse("Copper 29 35 63.546 Cu"));

elements.Add(Atom.Parse("Zinc 30 35 65.39 Zn"));

elements.Add(Atom.Parse("Gallium 31 39 69.723 Ga"));

elements.Add(Atom.Parse("Germanium 32 41 72.64 Ge"));

elements.Add(Atom.Parse("Arsenic 33 42 74.9216 As"));

elements.Add(Atom.Parse("Selenium 34 45 78.96 Se"));

elements.Add(Atom.Parse("Bromine 35 45 79.904 Br"));

elements.Add(Atom.Parse("Krypton 36 48 83.8 Kr"));

elements.Add(Atom.Parse("Rubidium 37 48 85.4678 Rb"));

elements.Add(Atom.Parse("Strontium 38 50 87.62 Sr"));

elements.Add(Atom.Parse("Yttrium 39 50 88.9059 Y"));

elements.Add(Atom.Parse("Zirconium 40 51 91.224 Zr"));

elements.Add(Atom.Parse("Niobium 41 52 92.9064 Nb"));

elements.Add(Atom.Parse("Molybdenum 42 54 95.94 Mo"));

elements.Add(Atom.Parse("Technetium 43 55 98 Tc"));

elements.Add(Atom.Parse("Ruthenium 44 57 101.07 Ru"));

elements.Add(Atom.Parse("Rhodium 45 58 102.9055 Rh"));

elements.Add(Atom.Parse("Palladium 46 60 106.42 Pd"));

elements.Add(Atom.Parse("Silver 47 61 107.8682 Ag"));

elements.Add(Atom.Parse("Cadmium 48 64 112.411 Cd"));

elements.Add(Atom.Parse("Indium 49 66 114.818 In"));

elements.Add(Atom.Parse("Tin 50 69 118.71 Sn"));

elements.Add(Atom.Parse("Antimony 51 71 121.76 Sb"));

elements.Add(Atom.Parse("Iodine 53 74 126.9045 I"));

elements.Add(Atom.Parse("Tellurium 52 76 127.6 Te"));

elements.Add(Atom.Parse("Xenon 54 77 131.293 Xe"));

elements.Add(Atom.Parse("Cesium 55 78 132.9055 Cs"));

elements.Add(Atom.Parse("Barium 56 81 137.327 Ba"));

elements.Add(Atom.Parse("Lanthanum 57 82 138.9055 La"));

elements.Add(Atom.Parse("Cerium 58 82 140.116 Ce"));

elements.Add(Atom.Parse("Praseodymium 59 82 140.9077 Pr"));

elements.Add(Atom.Parse("Neodymium 60 84 144.24 Nd"));

elements.Add(Atom.Parse("Promethium 61 84 145 Pm"));

elements.Add(Atom.Parse("Samarium 62 88 150.36 Sm"));

elements.Add(Atom.Parse("Europium 63 89 151.964 Eu"));

elements.Add(Atom.Parse("Gadolinium 64 93 157.25 Gd"));

elements.Add(Atom.Parse("Terbium 65 94 158.9253 Tb"));

elements.Add(Atom.Parse("Dysprosium 66 97 162.5 Dy"));

elements.Add(Atom.Parse("Holmium 67 98 164.9303 Ho"));

elements.Add(Atom.Parse("Erbium 68 99 167.259 Er"));

elements.Add(Atom.Parse("Thulium 69 100 168.9342 Tm"));

elements.Add(Atom.Parse("Ytterbium 70 103 173.04 Yb"));

elements.Add(Atom.Parse("Lutetium 71 104 174.967 Lu"));

elements.Add(Atom.Parse("Hafnium 72 106 178.49 Hf"));

elements.Add(Atom.Parse("Tantalum 73 108 180.9479 Ta"));

elements.Add(Atom.Parse("Tungsten 74 110 183.84 W"));

elements.Add(Atom.Parse("Rhenium 75 111 186.207 Re"));

elements.Add(Atom.Parse("Osmium 76 114 190.23 Os"));

elements.Add(Atom.Parse("Iridium 77 115 192.217 Ir"));

elements.Add(Atom.Parse("Platinum 78 117 195.078 Pt"));

elements.Add(Atom.Parse("Gold 79 118 196.9665 Au"));

elements.Add(Atom.Parse("Mercury 80 121 200.59 Hg"));

elements.Add(Atom.Parse("Thallium 81 123 204.3833 Tl"));

elements.Add(Atom.Parse("Lead 82 125 207.2 Pb"));

elements.Add(Atom.Parse("Bismuth 83 126 208.9804 Bi"));

elements.Add(Atom.Parse("Polonium 84 125 209 Po"));

elements.Add(Atom.Parse("Astatine 85 125 210 At"));

elements.Add(Atom.Parse("Radon 86 136 222 Rn"));

elements.Add(Atom.Parse("Francium 87 136 223 Fr"));

elements.Add(Atom.Parse("Radium 88 138 226 Ra"));

elements.Add(Atom.Parse("Actinium 89 138 227 Ac"));

elements.Add(Atom.Parse("Protactinium 91 140 231.0359 Pa"));

elements.Add(Atom.Parse("Thorium 90 142 232.0381 Th"));

elements.Add(Atom.Parse("Neptunium 93 144 237 Np"));

elements.Add(Atom.Parse("Uranium 92 146 238.0289 U"));

elements.Add(Atom.Parse("Americium 95 148 243 Am"));

elements.Add(Atom.Parse("Plutonium 94 150 244 Pu"));

elements.Add(Atom.Parse("Curium 96 151 247 Cm"));

elements.Add(Atom.Parse("Berkelium 97 150 247 Bk"));

elements.Add(Atom.Parse("Californium 98 153 251 Cf"));

elements.Add(Atom.Parse("Einsteinium 99 153 252 Es"));

elements.Add(Atom.Parse("Fermium 100 157 257 Fm"));

elements.Add(Atom.Parse("Mendelevium 101 157 258 Md"));

elements.Add(Atom.Parse("Nobelium 102 157 259 No"));

elements.Add(Atom.Parse("Rutherfordium 104 157 261 Rf"));

elements.Add(Atom.Parse("Lawrencium 103 159 262 Lr"));

elements.Add(Atom.Parse("Dubnium 105 157 262 Db"));

elements.Add(Atom.Parse("Bohrium 107 157 264 Bh"));

elements.Add(Atom.Parse("Seaborgium 106 160 266 Sg"));

elements.Add(Atom.Parse("Meitnerium 109 159 268 Mt"));

elements.Add(Atom.Parse("Roentgenium 111 161 272 Rg"));

elements.Add(Atom.Parse("Hassium 108 169 277 Hs"));

return elements;

}

Create static methods that does the following and then called them from your main

//Create class methods to do the following:

//1 - Display all the items in the collection elements

//2 - Serialize the first item using xml format and save to a suitable file

//3 - Read the above file and display the item

~~//4 - Serialize the entire collection using xml format and save to a suitable file~~

~~Reminder: To simplify access to the XmlSerializer class, you must add a using statement.~~

~~//5 - Read the above file and display the all the items~~

//6 - Serialize the first item using json format and save to a suitable file

Reminder: To gain access to the **JavaScriptSerializer** class, you must add a reference to the **System.Web.Extension.dll** library and to simplify access you need to insert a using statement.

//7 - Read the above file and display the item

//8 - Serialize the entire collection using json format and save to a suitable file

//9 - Read the above file and display all the items