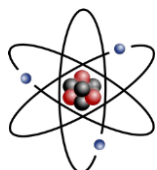




3ª Avaliação



O átomo é a unidade fundamental da matéria e a menor fração capaz de identificar um elemento químico. Em nível microscópico, pode-se afirmar que um elemento químico é o conjunto de átomos que possui o mesmo número atômico. Já uma molécula é um conjunto de átomos – iguais ou diferentes – unidos por ligações covalentes.

Com base nesses conceitos elementares da Química, desenvolva uma aplicação em Dart que implemente os seguintes requisitos e funcionalidades:

REQUISITOS E FUNCIONALIDADES:

1. Implemente a classe “Element” para representar um único elemento químico da Tabela Periódica e que armazene os seguintes dados do elemento: símbolo, nome em português, nome em latim e peso atômico.
2. Implemente a classe “Elements” que deverá armazenar os dados de todos os elementos químicos da Tabela Periódica. Os dados dos elementos deverão ser carregados de um arquivo no momento da instanciação da classe.

Na seção “Recursos do Projeto” apresentada adiante, será disponibilizado um repositório que contém um arquivo CSV com os dados de todos os elementos químicos.

3. A classe “Elements” deverá ser um *singleton* utilizado internamente pelas classes do projeto.
O padrão “singleton” determina que uma classe tenha apenas uma instância de si mesma em toda a aplicação.
4. Implemente a classe “Atom” que representa um único átomo, sabendo que este é definido pelo seu símbolo na Tabela Periódica. Por exemplo: “H”, “O”, “C”, “Cl”, “Ag”, etc.
5. O símbolo do átomo deverá ser o único parâmetro do construtor da classe “Atom”. Se o símbolo for inválido, a classe deverá levantar uma exceção.
6. O símbolo deverá ser a string de representação de uma instância da classe “Atom”.
7. Defina a classe “Molecule” que representa uma molécula, sabendo que esta é definida por sua fórmula molecular. Por exemplo, a fórmula molecular da água é “H2O”, composta por 2 átomos de hidrogênio (“H”) e 1 átomo de oxigênio (“O”).

Para simplificar o problema, serão consideradas apenas fórmulas com letras e dígitos numéricos. Não serão avaliadas fórmulas complexas como “(NH4)2SO4” ou “Na2[B4O5(OH)4]8H2O”.

8. A fórmula molecular deverá ser o único parâmetro do construtor da classe “Molecule”.
9. A classe “Molecule” deverá implementar o *getter* “formula” que retorna uma string com a fórmula da molécula representada pelo objeto.

Exemplos de fórmulas: “O2”, “H2O”, “NaCl”, “H2SO4”, “C6H12O6”, etc.

10. A classe “Molecule” deverá implementar um *setter* “formula” que recebe uma string com a fórmula da molécula a ser representada pela instância da classe. Se a fórmula for inválida a classe deverá levantar uma exceção.
11. A classe “Molecule” deverá implementar um *getter* “weight” que retorna o peso atômico da molécula. O peso atômico da molécula é dado pela soma dos pesos atômicos de todos os átomos que compõem a molécula.
12. A classe “Molecule” deverá implementar a interface genérica “Comparable<T>”, que permitirá que instâncias de “Molecule” possam ser comparadas umas com as outras. O parâmetro de comparação utilizado deverá ser o peso atômico da molécula representada pela instância.

A interface `Comparable<T>` que é uma classe abstrata já existente no SDK do Dart. Veja a documentação da interface `Comparable<T>` em api.dart.dev.

13. Seu código será testado com a função `main()` existente no arquivo “main.dart”, que poderá ser baixado do repositório GitHub disponibilizado na seção “Recursos do Projeto” apresentada adiante.

O código do arquivo “main.dart” não deverá ser alterado. Seu código deve se adaptar às chamadas lá existentes.

14. A codificação do projeto deverá observar os princípios de estilo do Effective Dart.

<https://dart.dev/guides/language/effective-dart/style>

15. O projeto deverá ser disponibilizado em um repositório com acesso público do GitHub, com “main” como seu *branch* principal (*default* do GitHub). Qualquer outro *branch* não será considerado.

CRITÉRIOS DE AVALIAÇÃO:

- Cada requisito/funcionalidade será avaliado individualmente e receberá uma das seguintes notas:

Nota	Grau de atendimento ao requisito/funcionalidade
0	Não atende
1 a 4	Atende parcialmente
5	Atende completamente

- A nota final será determinada pela média ponderada das notas de cada requisito/funcionalidade.