TESTING

Measure Converter

Unit Testing Exercise

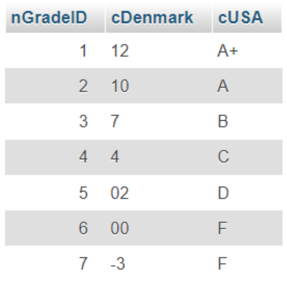
Feel free to choose your preferred programming language(s), unit testing framework(s) and libraries.

Part I

Create a backend application that performs conversions across several measures using object-oriented programming and according to the following specification:

- Length
  - Conversion between the Metric and Imperial systems
  - It only covers centimeters and inches, respectively
  - Length class. Methods:
    - Constructor
      - `@param     The numeric measure to convert with up to two decimals`
      - `@param     The system of said measure (Metric or Imperial)`
    - convert()
      - `It implements an if (if the system is Metric then … otherwise …)`
      - `@return    The value of the conversion with up to two decimals`
- Weight
  - Conversion between the Metric and Imperial systems
  - It only covers kilograms and pounds, respectively
  - Free class implementation
- Temperature
  - Conversion between the Celsius, Fahrenheit, and Kelvin scales
  - Temperature class. Methods:
    - Constructor
      - `@param     The numeric measure to convert with up to two decimals`
      - `@param     The temperature scale of said measure`
    - convert()
      - It implements a switch with the 6 possible conversions `(C to F, C to K, F to C, F to K, K to C, K to F)`
      - `Each switch calls a method that performs the specific conversion`
      - `@param     The destination temperature scale`
      - `@return    The value of the conversion with up to two decimals`

- Currency
  - Conversion between world currencies
  - Currency class. Methods:
    - Constructor
      - @param The base currency in 3-letter format (e.g., ‘DKK’)
    - convert()
      - It calls the API https://freecurrencyapi.net/
      - @param The numeric amount to convert with up to two decimals
      - @return The converted monetary amount with up to two decimals
- Grades
  - Conversion between the Danish and American grading systems
  - Grade class. Methods:
    - convert()
      - It queries a local database with the conversion information
      - Free choice of database model and DBMS
      - Possible implementation (MySQL):



nGradeID	cDenmark	cUSA
1	12	A+
2	10	A
3	7	B
4	4	C
5	02	D
6	00	F
7	-3	F

      - @param The grade to convert
      - @param The country to whose grading system the grade corresponds to
      - @return The converted grade

Part II

- Design and write unit tests for all the classes
  - Or maybe not (if you think that, in some case, they do not bring value)
- Design a comprehensive set of test cases
  - Apply black-box techniques
  - Look for extreme cases
- Write beautiful, efficient, maintainable unit tests
  - Use parameterised tests, data providers or similar
- Upload your code to Fronter