Arrowhead Software Engineer Take Home Problem

Objective: Implement a validation engine that can validate values with the rules specified below. It should be able to accommodate future changes gracefully, especially the potential for many additional rules. You are free to create as many files and import libraries as you see fit. The values to test against are in the junit test file named “ValidationEngineTest”, which specifies the expected pass/fail results and messages.

The validation engine should implement the provided interface “IValidationEngine” and should return an object that implements “IValidationResult”. The rules are evaluated in the order of tags in the validation tag array. If there is a failure, the failure message that is returned should just be based on the first rule that fails.

Validation Tags:

1. **Money** – The value is a valid monetary value. Note that we’re not looking for commas or dollar signs. Just the number.
   - Failure example: 2.0101 (reason: >2 decimal places)
   - Failure message: “Value is not a valid monetary value”
2. **Factor –** The value is a valid decimal number.
   - Failure example: 2.01656F (reason: alphanumeric character F isn’t acceptable. It must be strictly numeric).
   - Failure message: “Value is not a valid decimal number”
3. **notEmpty** – The value is not null or empty
   - Failure Example: “”
   - Fail message: “Value cannot be empty”
4. **notNegative** – The value is not a negative number
   - Failure Example: -1
   - Fail message: “Value cannot be negative”
5. **withinModLimits** – The limits are greater than or equal to 1.0 and less than or equal to 3.5.
   - Failure Example: Value = 3.6
   - Fail message: “Value is not within limits for a mod factor”
6. **allAlpha** – The value contains all alphabetic letters
   - Failure Example: Value=Bob #1
   - Fail message: “Value has non alphabetic characters”
7. **firstLetterUpperOnly** – The value’s first letter is upper case only
   - Failure Example: Value=bob
   - Fail message: “Value’s first letter is not uppercase”

The following files are provided for you:

1. IValidationEngine – The interface to validate the values
2. IValidationResult – The interface to store validation results
3. ValidationEngineTest – To test your code