HackerRank For Work

CSYE7200-Implicit Convers... 35 minutes

Question - 1

SCORE: **50 points**

Implicit Conversion

Write an implicit conversion that achieves the following:

Rules:

1. For String: Convert to Int if possible, otherwise discard it
2. For Char: Convert to Int if possible, otherwise discard it (Do not convert to ASCII)
3. For Int: Keep it
4. For Double: Round it to Int
5. For all other types: simply discard it

Implement `def ingestInt(l:List[Any]): List[Int]` that using your implicit conversion and return a List of Int

Example:
Input: List(1," abc ",3,true,'c',"4",'6',7.6) Output: List(1, 3, 4, 6, 8)
Explain: 1 is int, keep it; "abc" is string and cannot convert to int, discard it; 3 is int, keep; 'c' is char and cannot convert to int, discard it; "4" is string but can convert to int 4; '6' is char but can convert to int 6; 7.6 is double round it to 8.

Hints:

- You only need one implicit conversion, use case match to achieve all the rules.
- Although the output is List of Int, the return type of implicit conversion probably not.
- After you convert all the elements in the list, you need to keep the int and throw others, this is what you need to implement in `ingestInt` method, there are many ways you can do, eg. filter, flatMap, for comprehension and etc.

Since it is difficult to extinguish String and Char and Boolean type from std input, you can't customize your input, however, you may find all the inputs from the head of the code.