**Lab: Exceptions and Error Handling Lab**

Problems for exercise and homework for the ["C# OOP" course @ SoftUni"](https://softuni.bg/trainings/3585/csharp-oop-february-2022).

You can check your solutions here: [https://judge.softuni.org/Contests/3324/Exceptions-and-E HYPERLINK "https://judge.softuni.org/Contests/3324/Exceptions-and-Error-Handling-Lab"r HYPERLINK "https://judge.softuni.org/Contests/3324/Exceptions-and-Error-Handling-Lab"ror-Handling-Lab](https://judge.softuni.org/Contests/3324/Exceptions-and-Error-Handling-Lab)

* **Cards**

Create a class **Card** to hold a card’s **face** and **suit**.

* Valid card faces are: 2, 3, 4, 5, 6, 7, 8, 9, 10, J, Q, K, A
* Valid card suits are: S (♠), H (♥), D (♦), C (♣)

Both face and suit are expected as an uppercase string. The class also needs to have a **toString()** method that prints the card’s face and suit as a string in the following format:

**"[{face}{suit}]" – example: [A♠] [5♣] [10♦]**

Use the following UTF code literals to represent the suits:

* \u2660 – Spades (♠)
* \u2665 – Hearts (♥)
* \u2666 – Diamonds (♦)
* \u2663 – Clubs (♣)

Write a program that takes a deck of cards as a string array and prints them as a sequence of cards (space separated). Print an exception message **"Invalid card!"** when an invalid card definition is passed as input.

**Input**

* A single line with the faces and suits of the cards in the format:

**"{face} {suit}, {face} {suit}, …"**

**Output**

* As output, print on the console the list of cards as strings, separated by space.

**Examples**

|  |  |
| --- | --- |
| **Input** | **Output** |
| A S, 10 D, K H, 2 C | [A♠] [10♦] [K♥] [2♣] |
| 5 C, 10 D, king C, K C, Q heart, Q H | Invalid card!  Invalid card!  [5♣] [10♦] [K♣] [Q♥] |

**Hints**

Write a method **CreateCard(face, suit)**, which creates a card face and card suit and returns a **Card** object. The method should throw an exception if invalid data are given in its arguments. Later, you can catch the exception and print an error message.

* **Sum of Integers**

You will receive a sequence of **elements of different types**, separated by **space**. Your task is to calculate the sum of all valid integer numbers in the input. Try to add each element of the array to the sum and **write** **messages** for the possible **exceptions** while processing the element:

* If you receive an **element**, which is **not in the correct format** **(FormatException)**:  
  **"The element '{element}' is in wrong format!"**
* If you receive an **element**, which is **out of the integer** **type range (OverflowException)**:  
  **"The element '{element}' is out of range!"**

After each processed element add the following message:

**"Element '{element}' processed - current sum: {sum}"**

At the end print the total sum of all integers:

**"The total sum of all integers is: {sum}"**

**Examples**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 2147483649 2 3.4 5 invalid 24 -4 | The element '2147483649' is out of range!  Element '2147483649' processed - current sum: 0  Element '2' processed - current sum: 2  The element '3.4' is in wrong format!  Element '3.4' processed - current sum: 2  Element '5' processed - current sum: 7  The element 'invalid' is in wrong format!  Element 'invalid' processed - current sum: 7  Element '24' processed - current sum: 31  Element '-4' processed - current sum: 27  The total sum of all integers is: 27 |
| 9876 string 10 -2147483649 -8 3 4.86555 8 | Element '9876' processed - current sum: 9876  The element 'string' is in wrong format!  Element 'string' processed - current sum: 9876  Element '10' processed - current sum: 9886  The element '-2147483649' is out of range!  Element '-2147483649' processed - current sum: 9886  Element '-8' processed - current sum: 9878  Element '3' processed - current sum: 9881  The element '4.86555' is in wrong format!  Element '4.86555' processed - current sum: 9881  Element '8' processed - current sum: 9889  The total sum of all integers is: 9889 |