Задание к модулю Exception Handling

# Задание 1

Напишите консольное приложение, которое выводит на экран первый из введенных символов каждой строки ввода. Опишите корректное поведение приложения, если пользователь ввел пустую строку.

Приложение использует кодировку UTF-8, если не оговорено другое. Возвращать null не разрешается.

1. Получить от пользователя набор строк
   1. Максимальная длина строки – Int32.MaxValue
   2. Строки могут быть null.
   3. Строки могут быть пустыми.
   4. Строки могут начинаться с пробельных или непечатных символов.
   5. Строки могут состоять из печатных символов. 🡨 нормальный режим работы
2. Получить первый символ от каждой строки
   1. Если строка null – бросить ArgumentNullException
   2. Если строка пустая ~~– вывести символ U+0000~~ бросить IndexOutOfRange
   3. Если строка начинается с пробельных или непечатных символов, не следует учитывать их, но следует вывести первый печатный. Если строка целиком состоит из пробельных или непечатных символов, то такая строка приравнивается к пустой.
   4. Если строка начинается с печатного символа – вывести его
3. Вывести полученные символы
   1. Каждый символ выводится отдельно
   2. ~~Символ U+0000 показывается как «U+0000»~~
   3. Вывод пробельных или непечатных символов не допускается согласно 1.d

# Задание 2

Напишите библиотеку классов, которая содержит метод преобразования строки в целое число. Использовать стандартные методы int.Parse() и int.TryParse(), равно как и другие стандартные методы парсинга, а так же сторонние библиотеки запрещено. Предусмотрите корректную обработку ошибок. Возвращать null не разрешается.

Приложение должно получать на вход строку в кодировке UTF-8 и возвращать число, которое попадает в диапазон Int32. Если полученное число выпадает за пределы данного диапазона, то бросается OferflowException. Далее подразумевается, что цифры – это символы 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, не-цифры – всё, что не входит в этот набор. Минус – символ U+002D, плюс - U+002B

Задачи:

1. Получить от пользователя строку
   1. Строка может быть null.
   2. Строка может быть пустой
   3. Строка может начинаться или заканчиваться последовательностью пробельных или непечатных символов
   4. В строке могут встречаться пробельные или непечатные символы между печатными
   5. Строка может состоять только из не-цифр
   6. Строка может состоять из не-цифр и цифр
   7. Строка может состоять из цифр
2. Преобразовать строку
   1. Если строка null – бросить ArgumentNullException
   2. Если строка пустая – бросить FormatException
   3. Если строка содержит в себе только пробльные или непечатные символы – она приравнивается к пустой строке.
   4. Если строка содержит пробельные или непечатные символы, но также содержит в себе ещё и печатные, то следует отбросить все, кроме печатных и далее работать только с ними.
   5. Если в строке встречаются не-цифры – бросить FormatException
   6. Если строка начинается с плюса или цифры – число считается положительным, если с минуса – отрицательным.
   7. В начале строки допускается только один знак плюса или минуса. Если оных больше – бросить FormatException
   8. Если строка состоит только из цифр – преобразовать строку в число
3. Преобразовать строку, полученную из предыдущего шага в число
4. Вернуть число пользователю

Примеры:

* “˽ ˽” -> “” -> FormatException
* “˽ ˽ ˽ ˽asd ˽ ˽” -> “asd” -> FormatException
* “˽ ˽123 ˽ ˽“-> “123” -> 123
* “˽ ˽1 ˽2 ˽3” -> FormatException
* “˽ ˽1a ˽ ˽” -> FormatException
* “˽ ˽ ˽1 ˽a ˽” -> “1 ˽a” -> FormatException
* “1” -> 1
* “-1” -> -1
* “˽- ˽1 ˽” –> “-1” -> -1
* “2147483647” -> 2147483647
* “-2147483646” -> -2147483646
* “---2147483646” -> FormatException
* “2147483648” -> Overflow