**DataTable Validation**

MỤC LỤC

[1. Tổng quan 1](#_Toc61025273)

[1.1. Yêu cầu đặt ra: 1](#_Toc61025274)

[1.2. Nội dung thực hiện: 1](#_Toc61025275)

[2. Mô tả nội dung 1](#_Toc61025276)

[2.1. Lấy ra danh sách Argument 1](#_Toc61025277)

[2.2. Lấy ra danh sách RuleTable 2](#_Toc61025278)

[2.3. Đơn giản hóa việc tạo list RuleTable 3](#_Toc61025279)

[2.4. Kết quả chương trình 3](#_Toc61025280)

[3. Tài liệu tham khảo 3](#_Toc61025281)

# 1. Tổng quan

## 1.1. Yêu cầu đặt ra:

* Sửa lại thuật toán xử lý ListArgument, truyền đầu vào cho Expression

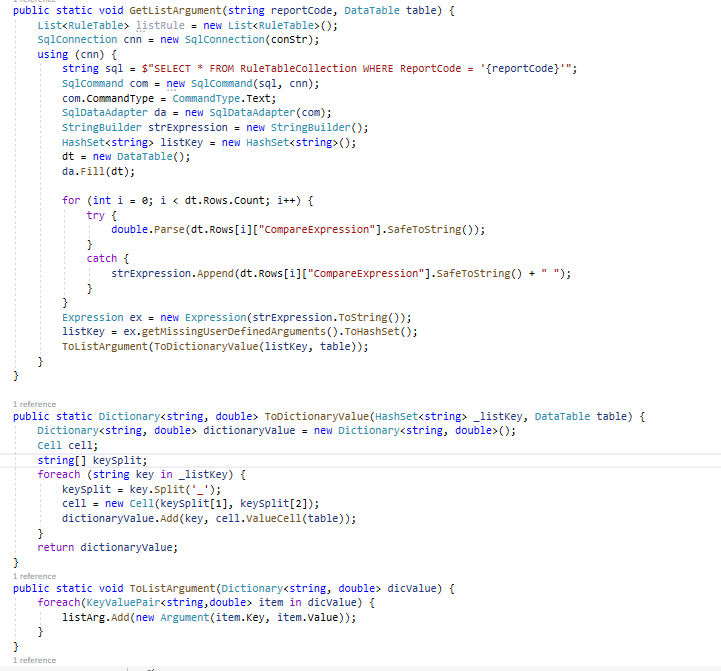
## 1.**2. Nội dung thực hiện:**

* Lấy danh sách key từ danh sách Expression của RuleTableCollection
* Mapping sang Dictionary
* Chuyển Dictionary thành ListArgument làm đầu vào cho các xử lý Expression

# 2. Mô tả nội dung

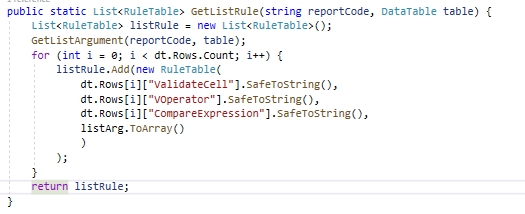
## 2.1. Lấy ra danh sách Argument

* **Mô tả:**
  + Đọc ra các Expression từ cơ sở dữ liệu
  + Cộng chuỗi tạo thành chuỗi các Expression tổng hợp
  + Dùng phương thức getMissingUserDefinedArguments() của Mxparser để lấy ra các key, lọc trùng sử dụng HashSet.
  + Gọi phương thức ToDictionaryValue để mapping Key và Value
  + Gọi Phương thức ToListArgument để lấy ra danh sách Argument
* **Code:**



## 2.2. Lấy ra danh sách RuleTable

* **Mô tả:**
  + Gọi phương thức GetListArgument để lấy ra danh sách Argument và khởi tạo RuleTable
* **Code:**

****

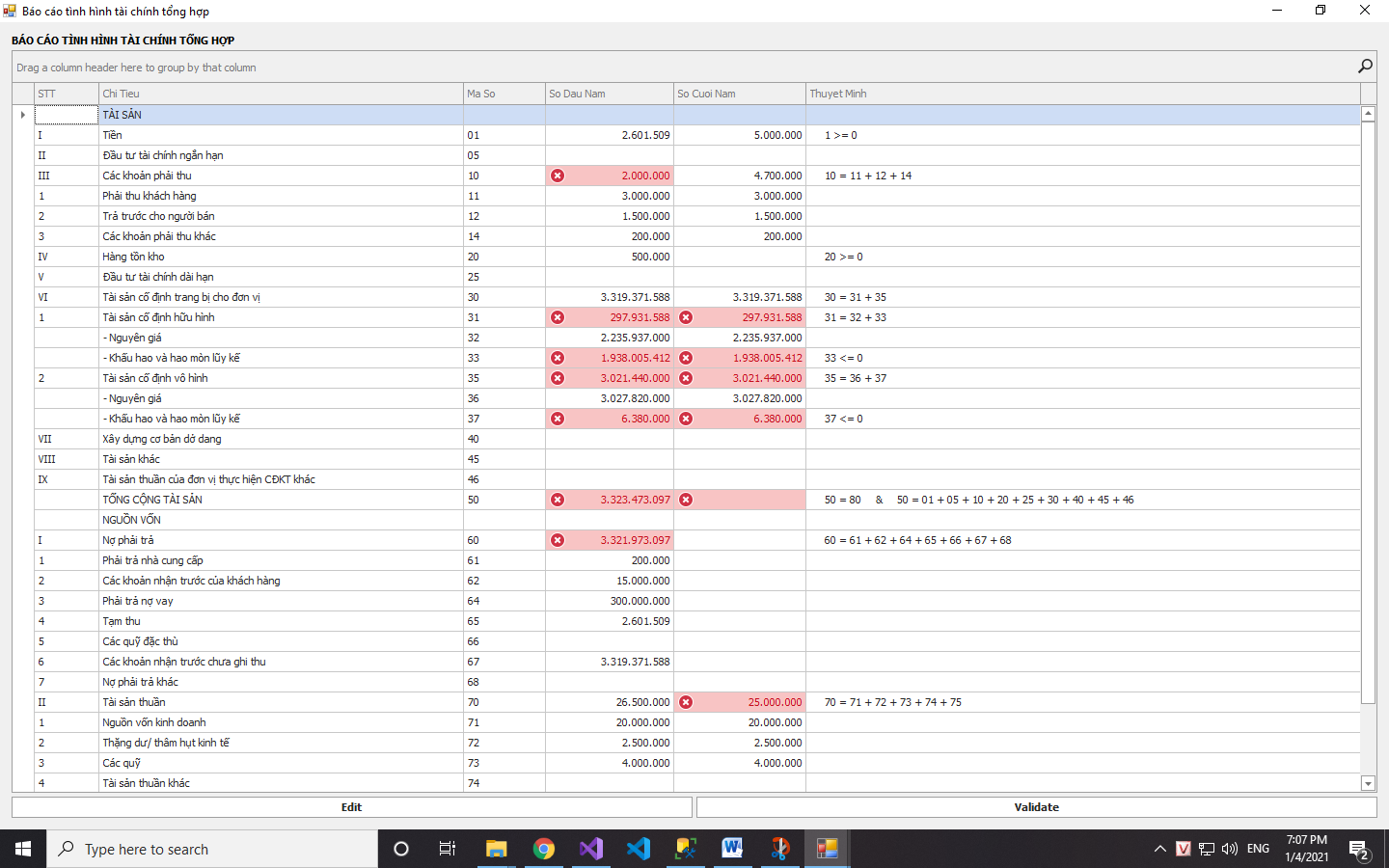
## 2.3. Đơn giản hóa việc tạo list RuleTable

public void CreateListRule() {

g\_listRule = DataProvider.GetListRule("B01/BCTC-TH", g\_table);

}

## 2.4. Kết quả chương trình



# 3. Tài liệu tham khảo

Link trang web:

* <https://fluentvalidation.net/>
* <https://en.wikipedia.org/wiki/Mxparser>
* <https://www.codeproject.com/Tips/381509/Math-Parser-NET-Csharp>
* <https://www.mathparsers.com/math-parser-for-csharp-dotnet/>