**현장실습 주간보고서[3주차]**

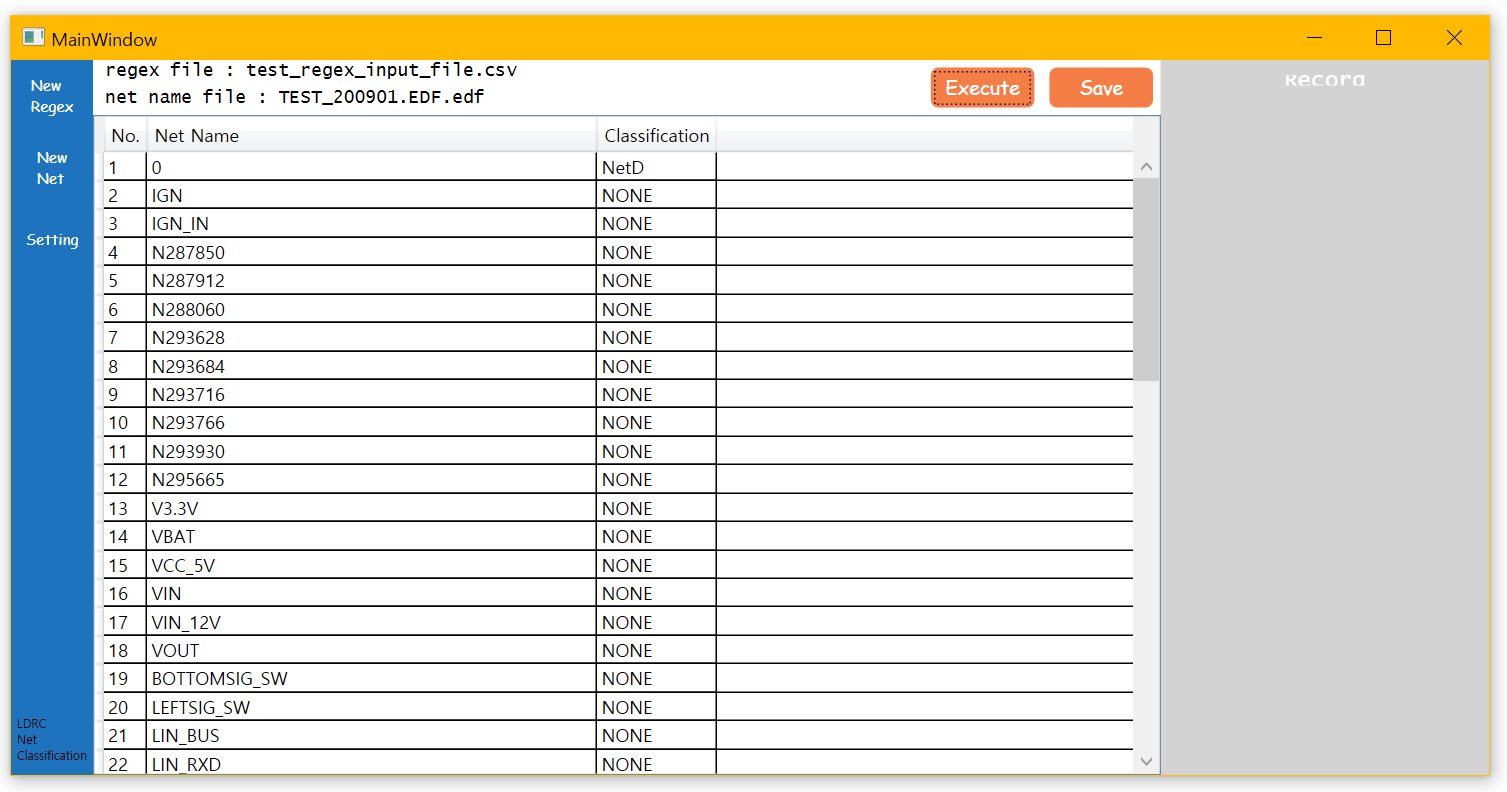
**21.1.18 ~ 21.1.22**

**아주대 미디어학부 황지훈**

**펜타큐브-개발**

**1. WPF를 이용한 LDRCNetClassification의 UI 부분 구현**

Net name과 해당 부류의 mapping결과를 UI에 보이기 위해 DataGrid를 추가했다. DataGrid에 보일 Collection(GridItems)을 정의했다. 그리고 DataGrid의 각 row를 표현할 객체(MappedRowData)를 정의하고 GridItem이 MappedRowData를 저장하는 Collection으로 선언했다. DataGrid에 데이터를 보여지게 하려면 Binding옵션이 필요했다. 따라서 DataGrid 테그에는 GridItems(Collection)을 binding하고, DataGridTextColumn 테그에는 MappedRowData의 각 property들을 binding했다. 추가적으로, Collection을 Binding할 때에는 ObservableCollection만 가능하다. 아래는 binding을 완료하고 testcase 파일을 실행한 결과이다.



**2. Nullable**

특정 data type이 null이 될 수도 있음을 명시하는 Nullable을 알게 됐다. 특정 type을 나타내는 T뒤에 ? 를 붙이면 해당 type이 가질 수 있는 값에 추가로 null값을 가질 수 있다. 예를 들어 int?[] arr = new int?[10] 과 같이 int형 배열을 선언하면 이 배열은 null값을 포함한 int형 값을 담을 수 있다. Nullable을 사용하면 해당 값 또는 객체가 존재하는지 쉽게 판단할 수 있다.

**3. File resource 접근**

File이나 Stream을 이용하여 파일에 접근하는 경우 원활하게 다루지 못한다면 메모리 누수가 발생할 수 있다. 따라서 이러한 경우에는 using을 사용하여 안전하게 스트림을 discard해주도록 한다. OpenFileDialog, SaveFileDialog등에 있는 InitialDirectory를 사용하는 경우, 해당 path가 존재하지 않을 가능성을 감안해야 한다. 따라서 이러한 경우 오류가 발생할 확률이 높으므로 보통 환경변수를 사용한 값을 넣거나, 또는 아예 지정하지 않는 것이 좋다.

**4. DI Pattern**

Prism의 unity, wpf 패키지를 이용하면 DI(Dependency Injection)패턴을 적용할 수 있다. DI를 이해하기 위해 다음 예시를 보자. A 객체는 B 객체를 생성하는 등의 dependency가 있다고 하자. 이때 두 객체 간에는 긴밀한 결합(tight coupling)이 생기고, tight coupling에 따라 B객체를 변경하면 A객체도 변경되는 사항이 많이 발생한다. 이러한 문제점이 발생할 가능성이 있는 곳에 DI pattern을 적용할 수 있다. DI 패턴을 적용하면 A라는 클래스가 B라는 클래스를 의존한다고 하여도, A클래스는 B클래스가 아닌, B의 인터페이스에 의존하게 만든다. 그리고 실제로 A클래스에게 B클래스를 설정해주는 일은 외부(Prism 패키지)에서 진행된다.

**5. Value Tuple**

MS에서 제공하는 ValueTuple 패키지를 사용하면 2개 이상의 pair로 이루어진 데이터를 tuple형태로 관리할 수 있다. 현재 프로젝트에서는 정규표현식을 담는 부분에 적용했다. 기존에는 Regex reference를 key로하는 Dictionary를 이용했지만, 객체 참조변수를 Dictionary의 key로 사용할 경우, 해당 객체에 변경사항이 발생할 때 다른 reference에 쓰일 가능성이 있다. 하지만 해당 Regex 객체를 Dictionary에 저장한 이후로 Regex 객체에 전혀 변경사항이 발생하지 않는다는 조건이 있다면, 제한적으로 객체 참조변수를 Dictionary의 key로 이용할 수 있다.

따라서 정규표현식을 담는 Dictionary의 Key, Value를 Value tuple로 구성하도록 변경했고, value tuple들은 List로 관리했다. 또한 value tuple을 저장하는 List로 구현하면 하나의 정규표현식이 의미하는 Net분류가 2개 이상일 때에도 List에 담아 둘 수 있다.

**6. Wrapper**

특정 Object에 가능을 추가적으로 제공하거나, 이 Object에 interface를 추가하기 위해 사용한다.

**7. 소감 및 차주계획**

이번주에는 펜타큐브에서 제공받은 dll파일들과 도면파일(.edf)을 사용하여 좀 더 의미 있는 데이터를 이용하여 테스트했다. 확장 메소드라는 개념도 알게 되었지만 좀 더 공부가 필요한 상태이다. 차주에는 MVVM모델에 대한 공부를 하고 프로젝트 코드에 적용해 본다. 또한 classification을 의미하는 8개의 Group List에 저장해보고 이를 하나의 collection에 따로 압축하여 UI의 DataGrid에서 보여질 수 있도록 한다.