**현장실습 주간보고서[1주차]**

**21.1.4 ~ 21.1.8**

**아주대 미디어학부 황지훈**

**펜타큐브-개발**

**1. C# 기본 지식 학습 및 기업 코딩 convention 습득**

C#의 기본문법을 습득하고 visual studio에서 .NET Framework를 설치했다. C#은 java와 매우 유사한 문법을 갖지만 java보다는 확장된 기능을 제공한다. 예를 들어, java에서 사용하던 getter, setter들은 property로 처리할 수 있으며 var 키워드를 통해 어떤 data든지 담을 수 있다. Java와는 다르게 C#에는 class보다 넓은 범위를 표현하는 namespace 키워드가 존재한다. C#은 C/C++과 달리, 컴포넌트(class, 함수 등) 기반 언어이기 때문에 서로 다른 컴포넌트에 존재하는 클래스명이 우연하게 일치하는 오류를 발생할 수 있다. 그래서 이러한 문제를 줄이고자 C#에서는 역할이나 의미가 유사한 클래스들을 그룹으로 관리할 수 있는 기술을 제공하는데 이것이 namespace이다. .NET Framework는 MS에서 개발한 윈도우 프로그램 개발 및 실행 환경이다. .NET Core는 특정 OS에 종속되지 않는 프로그램을 개발할 수 있도록 한다. Java에 JVM이 있듯이, .NET Framework에는 CLR이라는 가상머신이 있다. C# 컴파일러(ILASM)는 소스코드를 기계어가 아닌 중간 언어(IL)를 우선적으로 생성한다.

각 기업은 개발을 함에 있어서 자체적으로 정의한 규칙인 coding convention이 존재한다. 따라서 앞으로 펜타큐브에서 개발을 할 때 적용할 수 있도록 펜타큐브의 코딩 convention을 살펴봤다.

**2. git 사용법 이해**

로컬에서 작업을 하고 원격 서버에 push함으로써 git을 이용한 프로그램의 버전 관리를 경험했다. 새로운 프로젝트(LDRCNetClassification)를 생성하고 branch를 생성하여 로컬에서 작업한 뒤, 작업 완료 후 pull request를 하여 검토자의 피드백을 받는 방식을 경험했다. 작업 시작일에 맞춰 branch이름을 정했다. branch에서 작업함으로써 master branch의 내용을 보존할 수 있다. 특히, Branch는 하나의 master branch에 대해서 여러 명이 각자의 작업을 할 때 더욱 유용하다. Pull request를 하면 다른 팀원들로부터 자신의 코드에 대해 피드백을 받을 수 있다. 이러한 방식의 소통은 앞으로 개발자를 한다면 많이 사용될 방식이 될 것이라고 생각했다. Merge가 완료되면 로컬에서는 다시 pull하여 다음 작업을 수행하는 것이 좋다. gitignore파일은 원격 서버에 push하지 않을 파일(디렉토리)을 지정한다. 개인정보나 중요한 자료 등을 gitignore에 작성해 두는 것이 좋다.

Merge 방식으로 3가지를 배웠다. 첫번째로 기본 merge이다. 이 방식은 branch에서 commit한 내용들도 함께 merge되어 master의 commit에 기록된다. 두번째로 squash이다. Squash는 branch에서 등록한 commit은 master의 commit에 기록되지 않으며 PR을 하고 merge를 했다는 기록만 남게 된다. 마지막으로 rebase방식이다. 이 방식은 base를 바뀐다는 의미로 master를 해당 branch로 옮기는 방식이다. 이번에 진행된 merge에서는 branch에서 작업한 내용이 크게 중요한 내용이 아니어서 squash방식을 이용하여 merge했다.

**3. LDRCNetClassification 프로젝트 시작**

수많은 Net이름이 주어졌을 때 이들이 정규표현식에 의해 어떤 classification으로 분류되는지 구현하는 프로젝트이다. 입력으로 정규표현식 file과 net name file을 받고, 출력으로 net name과 classification 대응 결과를 출력했다. Classification은 8종류가 있으며 각각 class로 정의했다. 8개의 class는 모두 name field를 갖으며 같은 형식으로 출력되므로 polymorphism패턴을 사용했다. 즉, abstract class를 생성하고 8개의 class들이 이를 상속하도록 구현했으며 같은 출력 방식을 표현하기 위해 abstract class에 ToString 메소드를 override했다.

입력 받은 정규표현식 파일에는 “정규표현식, classification” pair형태로 구성되도록 했다. pair형태가 아닌 정규표현식만 작성된 파일은 입력으로 받을 수 없다. Net name이 입력으로 들어왔을 때, 이 net name이 특정 정규표현식에 통과가 되었다 하더라도 어떤 classification과 대응되는지는 알 수 없기 때문이다. 따라서 정규표현식 파일은 임의의 “정규표현식, classification” pair형태로 구성된 csv파일로 작성했으며, 입력 받은 후에는 Dictionary<Regex instance, classification string>에 저장했다.

csv파일을 읽어오는 과정에서 StreamReader 객체를 사용했고 해당 reference는 using 키워드를 이용하여 보호했다. using키워드를 사용하면 해당 구간에서만 파일 리소스를 사용하고 자동으로 해제(discard)되기 때문에 메모리 사용을 절약할 수 있다.