1. 자기소개

- 1.1 학번과 이름: 2019182038 조소현

- 1.2 컴퓨터는 잘 다루는지 (0~10): 3.5점 정도 인 거 같습니다. 완전 다루지 못하는 것은 아니지만, 컴퓨터에 대한 기본적인 배경지식이 넓고 깊게 있기 보다는 보통의 사람들 정도로 아주 기본적인 정도로만 알고 있다고 생각되어 10점 만점에 3.5점 정도로만 컴퓨터를 다룰 줄 안다고 생각합니다.

- 1.3 프로그래밍 경험은 어떤 것이 있는지? 어떤 언어, 개발툴, 플랫폼을 해봤는지: 1학년 1학기에 C언어를 배웠고, 2학기에는 C++언어를 배웠습니다. 아직 두 언어를 잘하지는 못하고 여전히 어렵고 헤매고 있으나, 2학년 1학기에 자료구조 강의를 들으면서 오목 인공지능을 만들어 봤으며, 윈도우 프로그래밍 시간에 숙제로 로직은 좀 엉망이지만 2048게임을 구현해봤으며, 텀프로젝트로 리듬게임을 만드는 정도로 프로그래밍 경험을 가지고 있습니다.

- 1.4 이 수업에서 바라는 점이 무엇인지? 게임 개발의 기본적인 것들을 익히고 2D 게임 프로그래밍을 통해 게임 개발에 조금 더 다가가고 싶습니다. 또한, 요즈음 프로그래밍에 있어서 파이썬이 많이 화제가 되는 것 같아, 파이썬 언어를 배워보고 싶었습니다. 파이썬 언어의 기본적인 것들을 익히며 게임 개발에 활용하며 언어를 좀 더 깊게 배우고 싶습니다. 앞으로 졸업한 이후에도 개발자로써 살아갈텐데 졸업 전이나 후나 계속 프로그래밍을 하기 위해서는 프로그래밍과 가까워지고 프로그래밍을 하는 데에 있어서 익숙해져야 한다고 생각합니다. 따라서, 프로그래밍에 더욱 익숙해지고 개발하는 것에 한발짝 더 나아가는 수업이 되었으면 합니다.

2. OOP 관련 다음 슬라이드를 보고 내용 및 느낀 점을 정리하라

(1) 내용정리

다른 동물과 구별되는 사람만이 가진 진정한 고유의 특징은 바로 "상징 체계"를 만들고 공유하는 능력이다.

사람은 상징체계를 만들고 발전시켜 다음세대에 물려주는데 이것을 가능하게 하는 것이 바로 "추상"화 하는 능력이다.

곧 "추상"들을 만들고, 공유하고, 발전시키는 능력이 사람만이 가진 고유한 능력이라고 할 수 있다.

(추상화란 대상에서 특징만을 뽑아낸 것)

객체 지향이란, 곧 객체 위주로 볼 수 있고 한마디로 객체를 갖고 노는 것으로 생각할 수 있다.

객체 지향 키워드 5가지: 클래스(class), 오브젝트(object), 캡슐화(Encapsulation), 상속성(Ingeritance), 다형성(Polymorphism)

\*클래스(class)=추상(abstract), 오브젝트(object)=실체(instance)

- 클래스는 표현 대상들의 공통적인 특징에 대한 서술을 말한다. (특징에 대한 서술=추상)

추상은 사용자가 구체적인 내용을 생각하지 않고도 사용할 수 있는 기능이다.

- 오브젝트는 클래스의 인스턴스 곧 클래스가 실체로 만들어진 것이다.

- 곧 클래스는 공통 특징, 서술 이고 오브젝트는 구체, 실체 존재, 고유성을 말한다.

(예| 클래스: 멍멍이, 오브젝트: 옆집 점박이, 외가댁 누렁이, 우리집 초롱이)

오브젝트는 데이터와 데이터에 대한 조작을 묶은 것이다. 오브젝트의 모양은 클래스에 서술 되어 있다.

객체지향 프로그래밍언어는 캡슐화, 상속성, 다형성 이 3가지 특성을 지원하는 언어이다.

- 캡슐화: 데이터와 데이터를 다루는 방법을 묶는 것, 관련있는 것을 묶어서 이름을 부여하는 것이 추상화이다.

- 상속성: 하나의 클래스가 가지고 있는 특징(데이터+조작)을 그대로 다른 클래스가 물려받는 것이다.

상속성은 계보의 의미도 있다. 새로운 개념이 등장하였을 때 기존 클래스와 구분되는 특징만 기술하여 추가한다.

- 다형성: 상속성의 계층을 따라서 각각의 클래스에 한가지 이름을 줄 수 있는 것이다. 즉 같은 명령을 각기 다른 오브젝트에 줄 수 있다는 뜻이다.

객체지향관점에서 프로그램은 오브젝트에 메시지를 보내는 것으로 실행한다.

객체 지향은 프로그래밍 패러다임(관점)중의 하나이다. ex. JAVA, C++

(대표적인 프로그래밍 패러다임 두가지: 구조적 프로그래밍, 객체지향 프로그래밍|구조적 프로그래밍은 기능,절차 중심이고 객체지향 프로그래밍은 객체,관계 중심의 프로그래밍이다.)

- 객체지향의 장점 : 자연스러운 모델링, 생산성 향상, 점진적 개발, 견고한 의미적 모델링, 명확한 인터페이스 기술, 보안, 유지보수의 용이성

- 객체지향은 객체와 객체간 역할과 관계 중심이고 객체 중심이다.

- 객체지향 분석/설계 방법론 요약: 식별하고 일반화하고(오브젝트, 클래스), 대상들 간의 관계를 설정(상속)하고, 상호작용하도록 한다.

=> 객체지향을 한마디로 정의하면, 사람이 세계를 보고 이해하는 방법을 흉내 낸 방법론이다.

(2) 느낀점

객체 지향 언어란 어떤 것인지 그 개념을 좀 더 분명히 할 수 있었습니다. 개체지향언어를 많이 들어보기는 하였으나 정작 객체지향 프로그래밍과는 같은 것인지 또 어떤 것인지 잘 알지 못하였으나, 쉬운 설명으로써 이전보다 잘 알 수 있었습니다. 또한 객체지향 프로그래밍의 대표적 특징 중 기억에 남는 것이 클래스, 오브젝트, 상속인데 설명과 ppt를 보면서 C++에서 배웠던 것들 사용했던 것들이 떠올랐습니다. 객체지향 프로그래밍의 대표 예가 C++프로그래밍이 있는데, 실제로 C++을 사용하다보면 클래스를 이용할 일이 많이 생기는 것 같습니다. 저번 학기에 윈도우 프로그래밍을 배우며 클래스를 조금 활용해보며 상속 등에 대한 개념을 익혔던 것, C++수업 시간에 배웠던 클래스에 대한 개념이 다시 한 번 떠올랐습니다. 비록 아직 완전히 알지 못하고 어려우나 상속이란 어떤 개념이고, 클래스와 오브젝트간의 관계, 메시지 전달 등 클래스를 사용함에 있어서 필요한 내용들을 쉽게 터득할 수 있었습니다. 객체지향 프로그래밍에 있어서 오브젝트, 클래스, 관계 중심의 내용이 얼마나 중요한 지 알 수 있었습니다.

3.

- 3.1 위 사이트의 설명을 따라하고 간단히 요약하라.

command line은 프롬프트라고도 불리는 텍스트 기반 응용 프로그램으로 파일 보기, 처리, 조작이 가능합니다.

윈도우를 이용하고 있으니, 윈도우 키를 눌러서 명령 프롬프트를 검색하면 명령 프롬프트를 사용할 수 있게 됩니다.

>이런 기호가 뜨면 명령어를 입력하여 작업을 수행하면 됩니다.

cd: 현재 디렉터리 확인이 가능하다. cd Desktop을 입력하면 현재 디렉터리가 cd Desktop으로 변경됩니다. 곧 명령어 cd는 디렉터리를 변경하는데 사용됩니다.

dir은 디렉터리 혹은 파일 목록을 보여주는 명령어이다.

mkdir은 새 디렉터리를 생성하는 명령어이다.

del은 디렉터리 혹은 파일을 지우는 명령어인데 이것을 사용하여 파일을 지우면 재복구가 불가하니 주의하여 사용해야 한다.

- 3.2 본인이 가진 Command Line 방식의 숙련도에 대한 평가를 추가하라.(0~10)

현재 제가 가진 Command Line 방식의 숙련도는 2점 정도인 것 같습니다. 위 사이트의 설명을 따라해봤을 때처럼 가이드라인이 있을 때는 쉽게 따라할 수 있지만, 아무래도 명령프롬프트의 사용이 익숙하지 않다보니 가이드라인이나 설명이 없이

명령프로픔트를 사용하기에는 아직 무리가 있는 것 같습니다. 따라서 명령프롬프트 사용법을 몇번 더 따라하고 익혀야 숙련이 될 것 같습니다. 아직은 어떤 명령어가 있었는지 온전히 외우지 못하였고 사용하기에 어색하고 어려운 부분이 많습니다. 특히 컴퓨터에 대해 잘 알지 못하다보니 섣불리 건들였다가 내부의 무언가가 바뀌어 어려움을 겪을까봐 명령프롬프트를 사용함에 있어서 조심스러움을 가지게 되었습니다. 이전까지는 사용할 일이 크게 없을 줄 알아서 깊이 익히지 않았으나 이번 기회를 통해 사용방법을 좀 더 익히고 여러 명령어에 대한 숙련이 이루어져야만 한다고 생각합니다.