**TA Session #6: Note**

Compile – 코드를 기계어로 바꾸어주는 단계

Link

Run – 프로그램이 실행되는 단계

**Dynamic Memory Allocation**

통상적으로 메모리의 크기는 컴파일할 때 고정됨. 따라서 배열의 경우 그 길이가 컴파일할 때 고정되므로, 프로그램을 실행하는 도중 크기를 바꿀 수 없음. (크기를 바꾸는 코드를 작성하면 컴파일 시 에러가 발생함)

이러한 불편함을 해소하기 위해 ‘메모리의 동적 할당’이라는 개념을 사용. 동적 할당을 이용할 경우 런타임에 메모리가 할당되므로, 프로그램을 실행하는 도중 메모리의 크기를 변경할 수 있는 장점이 있음. (Python에서 np.zeros(n)을 생각하면 편할 것임.)

**Virtual Function(가상 함수)**

상속을 이용한 프로그래밍을 할 때, Parent class에 대한 포인터 변수를 Child class에 대한 변수를 가리키게 해야 할 경우가 생긴다. 이 경우 가상 함수 지정을 하지 않으면 멤버 함수 호출을 할 경우 Parent class의 멤버 함수가 호출이 된다.

즉, 가상 함수를 이용하는 것 자체가 동적으로 메모리를 할당하는 것과 같은 개념으로 이해할 수 있다.

가상함수는 실제로 포인터가 가리키고 있는 객체를 기준으로 멤버함수를 호출하게 한다.

|  |  |
| --- | --- |
| **Vftable** | **라면 끓이는 법(method)** |
| 라면 | 순수가상함수 |
| 신라면 | 주소 1에 저장 |
| 짜왕 | 주소 2에 저장 |
| 스파게티 | 주소 3에 저장 |

가상함수를 설정하게 되면, 클래스 ~ 가상함수(method)의 mapping이 컴파일시 이루어짐.

**Pure Virtual Function(순수가상함수)**

부모 클래스에서 굳이 함수를 정의할 필요가 없을 때(추상화) 설정

**Virtual Inheritance(가상 상속)**

C++에서 상속을 다중으로 하게 될 경우, 충돌이 일어날 수 있음. 가령 D라는 클래스가 B와 C 클래스 모두를 부모 클래스로 갖는다면? 그리고 B와 C가 모두 A라는 클래스를 부모 클래스로 갖는다면? 먼저 초기화에서의 문제가 생길 것임. 클래스 D의 초기화에서는 A, B, C 클래스의 생성자가 호출될 것임. 이 때, B가 호출하는 A 생성자와, C가 호출하는 A의 생성자 간 충돌이 일어날 수 있다.

이를 방지하기 위해 가상 상속을 하게 되는데, 이 때 D의 생성에서 A 생성자를 명시적으로 호출한다면 그 생성자가 호출될 것이다.