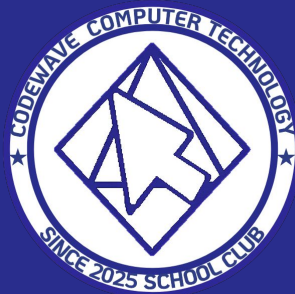


자료구조가 뭔데요

동아리 필수 전공과목

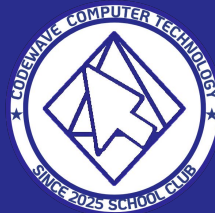


프로그래밍을 하면서 빠지지 않는 필수 개념?

네~ 맞습니다! 자료구조는 프로그래밍을 하면서 빠지지 않는 필수 개념이라서
우리 자랑스러운 **코드웨이브 동아리원**들은 알고 넘어가야겠죠?

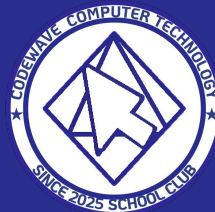


자료구조 이해하기



자료구조는 자료의 집합을 의미하며 각 원소들 사이의 관계가 **논리적으로** 정의된 일정한 규칙에 의하여 나열되며 자료에 대한 처리를 효율적으로 수행할 수 있도록 자료를 조직적, 체계적으로 구분하여 표현한 것을 의미합니다.

쉽게 말해 데이터들을 보기 좋고, 효율적으로 관리하기 위해서 **특정한 형태(구조)**로 만들어 둔다는 소리입니다.



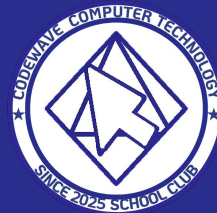
작업의 효율성, 추상화, 재사용성을 증가시키기 위하여 **상황에 따른 적절한 자료구조**를 선택해야 합니다.

자료의 처리를 좀 더 효율적으로 하기 위해서는 아래의 사항을 고려해야 합니다.

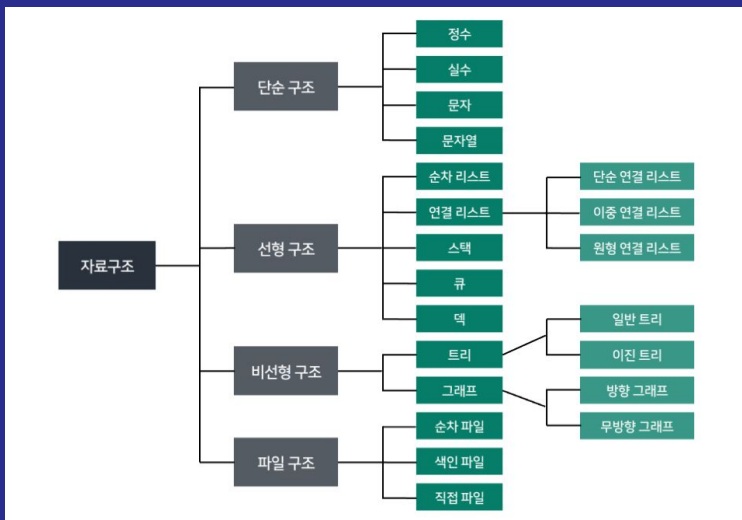
- 자료의 처리시간
- 자료의 크기
- 자료의 활용 빈도
- 자료의 갱신 정도
- 프로그램의 용이 성

한마디로 자료구조를 선택할때는 되도록 메모리가 적고, 재활용할 수 있는 자료를 쓰라는 소리다.

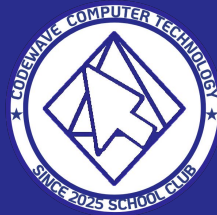
자료구조의 분류



자료구조에는 아래와 같은 다양한 구조들이 있습니다.



단순 구조



정수, 실수, 문자, 문자열 등의 기본 자료형을 의미합니다.

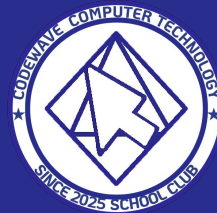
자료들 간의 앞뒤 관계가 1:1의 **선형 관계**를 이루는 구조입니다.
-리스트, 큐, 스택, 연결 리스트, 덱 등



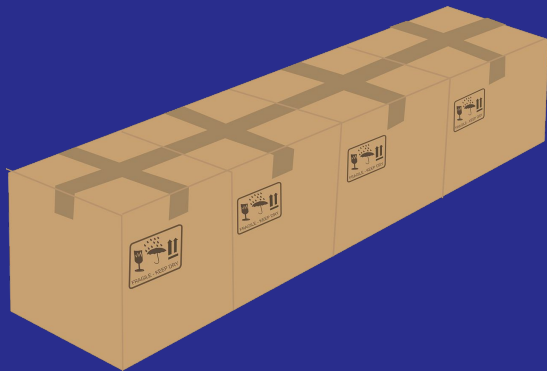
그럼 큐, 스택, 연결리스트가 뭐예요??

라고 물으실 것 같아서 준비했습니다.

리스트(list)

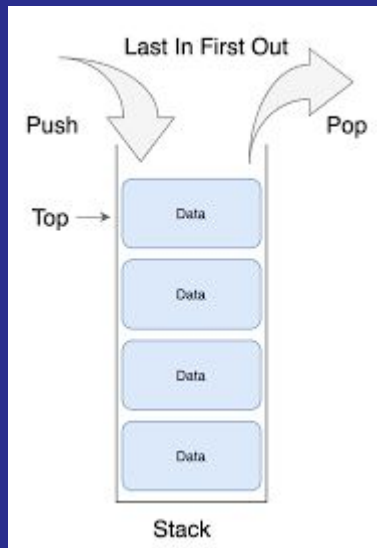


리스트는 간단히 말해 **순서대로 늘어선 박스**로 이해할 수 있습니다~
각 박스에는 다양한 데이터 타입을 저장할 수 있으며, 이 박스를 통틀어 리스트라고 합니다.



스택(stack)

스택은 **후입선출**의 개념을 가지는 자료구조중 하나로, Push라는 함수를 통해 Data를 저장시키고 계속해서 쌓아나간다.
Pop이라는 함수를 사용하면 Top의 위치에 있는 Data를 꺼낼수 있다.



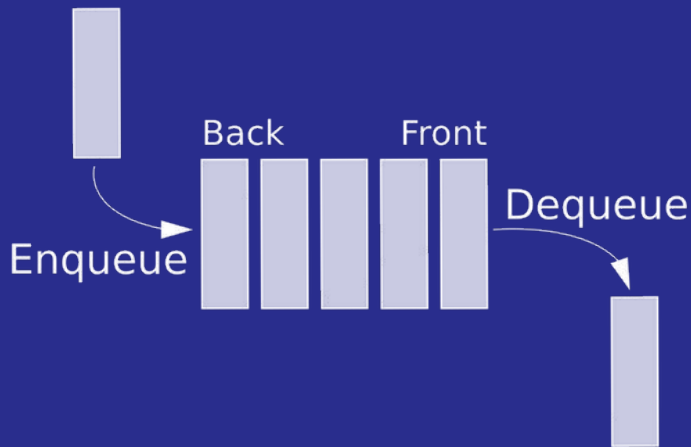
쉽게 이해를 해보자, 우리가 탄창에 총알을 넣으면
제일 먼저 삽입한 탄은 제일 아래로 가고, 마지막에 넣은 탄이
제일 먼저 발사된다.

그런 구조다~

따라서 탄창도 스택구조로 이루어져 있...

큐(queue)

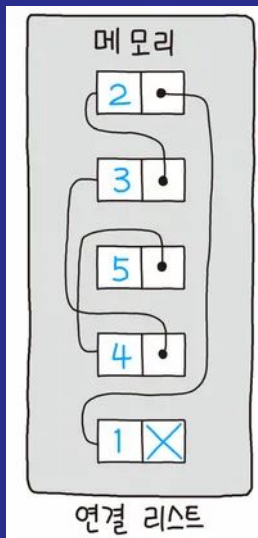
큐는 선입선출의 개념을 가지는 자료구조중 하나입니다.
쉽게 설명하자면 놀이기구를 타기위해 줄을 서는 행위를 Enqueue, 줄 맨 앞에 있는 사람이 놀이기구를 타게 되는 행위를 Dequeue라고 생각하면 쉽습니다.



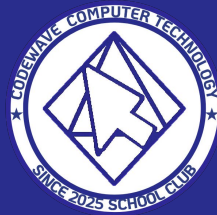
연결 리스트(linked list)

연결 리스트는 배열과 언뜻 비슷한 느낌을 내지만, 실제로는 굉장히 **복잡한 자료구조** 중 하나입니다. 쉽게 설명하자면 친구들이 줄을 서있는데, 각 친구가 자신 다음 친구의 손을 잡고 있는 구조라고 생각하시면 머리가 복잡해지는 일은 없으리라고 생각합니다.

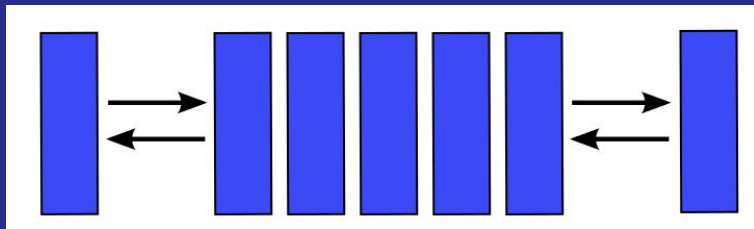
연결 리스트에서는 다음 친구(데이터)와 손을 잡는것을 '노드' 라고 표현합니다.



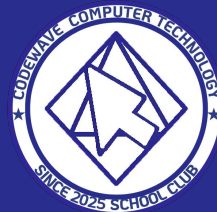
덱(deque)



덱은 한마디로 스택과 큐를 혼합한 자료구조라고 생각하시면 됩니다.
따라서 덱은 양쪽 끝에서 삽입과 삭제를 허용하는 자료구조 입니다.



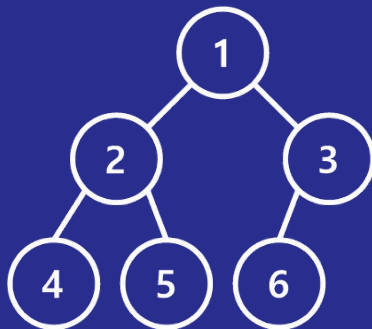
비선형 구조



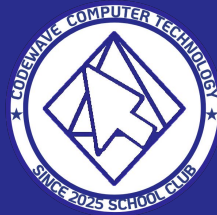
자료들 간의 앞뒤 관계가 1:다 또는 다:다의 관계를 가지고 있는 구조입니다.

트리(Tree) : 부모 노드 밑에 여러 자식 노드가 연결되고, 자식 노드가 부모가 되어 다시 각각의 자식 노드가 연결되는 재귀적 형태의 자료구조.

그래프(Graph) : 정점(Vertex)과 정점들을 연결하는 변(Edge)으로 구성된 자료구조.



파일 구조

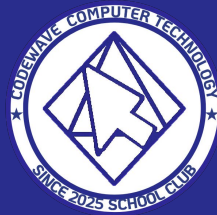


레코드의 집합인 파일에 대한 구조이다.
-순차 파일, 색인 파일, 직접 파일 등

네? 또 필수 개념이 하나 더 있다고요?

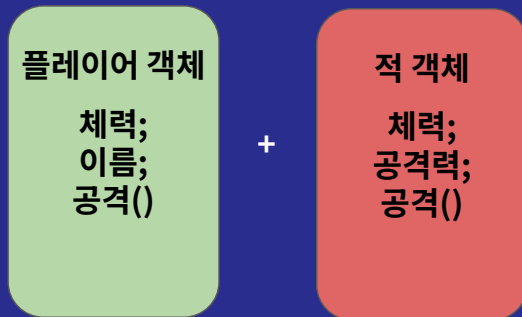
네~ 맞습니다! 객체지향 프로그래밍은 프로그래밍을 하면서 **또** 빠지지 않는 **필수 개념**이라서
다시 우리 자랑스러운 **코드웨이브 동아리원**들은 알고 넘어가야겠죠?





객체지향 프로그래밍 이해하기

객체 지향 프로그래밍 (Object-Oriented Programming, OOP)은 상태와 행위로 이루어진 객체들 간의 상호작용을 통해 프로그램을 구성하는 방식입니다.

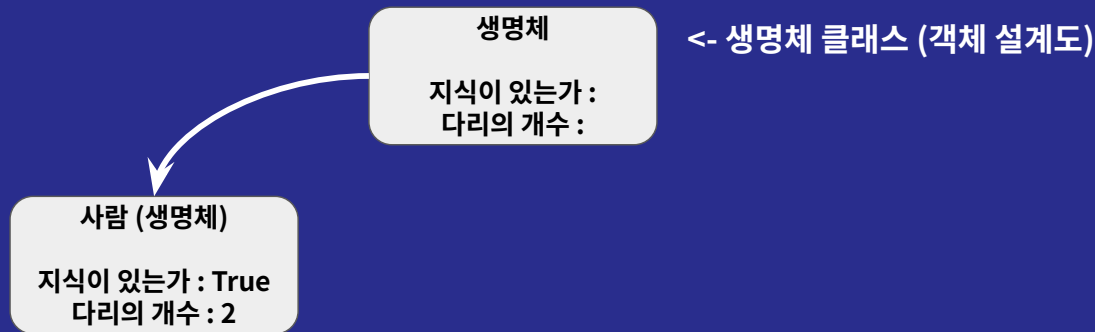


플레이어 객체만 있으면 프로그램이 진행되지 않지만 적 객체를 플레이어와 상호작용을 시켜주면 사냥을 하는 게임이 되겠죠?

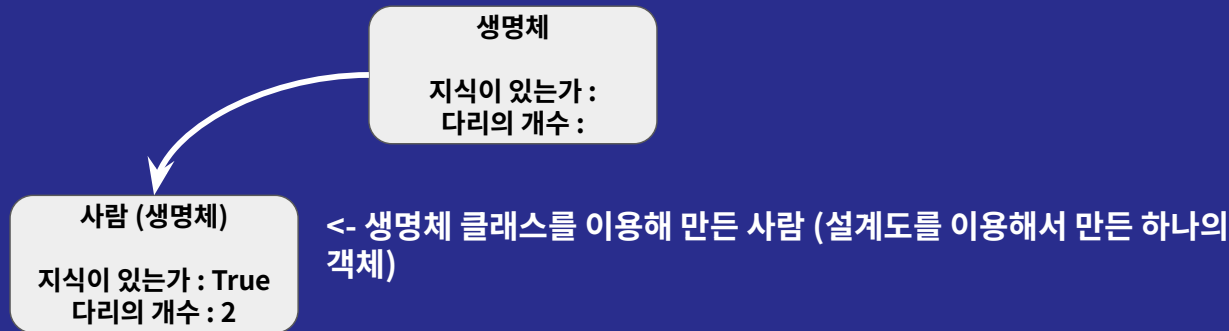
자 그럼 객체는 어떻게 생성하고 어떻게 사용하는 것일까요?

속성, 기능

클래스(Class)는 객체지향 프로그래밍에서 **특성**을 가진 객체를 생성해내는 마치 설계도같은 개념입니다.



객체(Object)는 속성과 기능을 가지는 물리적, 추상적 대상을 의미합니다.

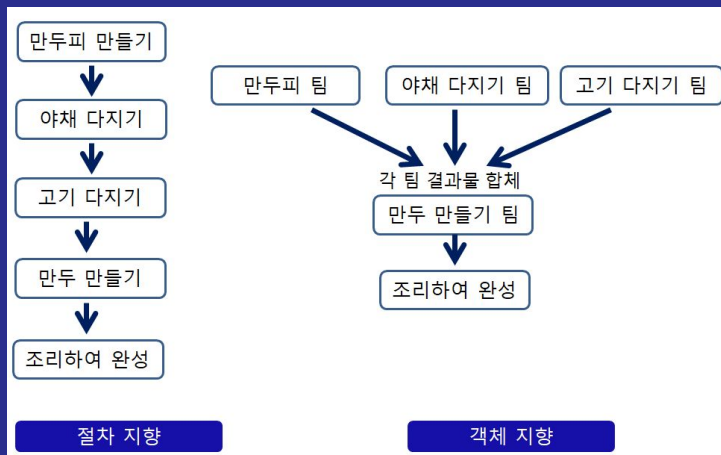


※ 객체는 참조한 클래스의 특성을 모두 가져옵니다.

절차지향 프로그래밍과의 차이점

객체지향과 **양대산맥**의 특성을 가지는 절차지향 프로그래밍과의 차이점은
아래 사진만 보셔도 아실거라고 생각합니다.

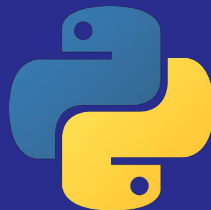
절차지향은 처음부터 **프로그램이 종료될 때까지** 순서대로 진행되는 특징이 있고
객체지향은 **여러 객체들이 상호작용**을 해서 프로그램을 이룬다는 특징이 있습니다.

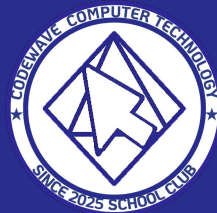


객체지향을 가지는 프로그래밍 언어

객체지향의 특성을 가지는 프로그래밍 언어로는

C++ , C# , Java , Python Javascript , Ruby , Swift 등이 있습니다.

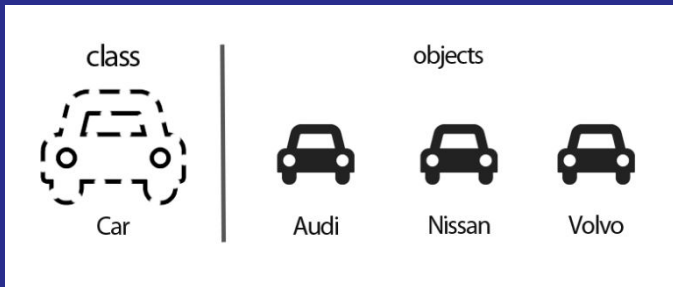




객체지향 프로그래밍은 마치 4원소 같은 추상화 , 캡슐화 , 상속 , 다형성의 네가지 특징을 가집니다.

지금부터 우리는 이 **4가지없는 4가지**의 특성을 알아보겠습니다.

추상화는 필요한 데이터 중에서도 가장 **중요한 데이터**만을 뽑아 클래스를 정의하는 과정을 뜻합니다.



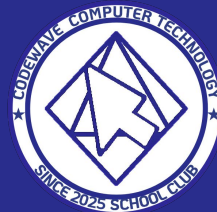
이때 Car라는 클래스가 정의되는 과정을 추상화 과정이라고 합니다.

캡슐화는 데이터 중에서도 **외부에 노출되면 안되는 데이터**를 숨기는 과정을 뜻합니다. (데이터 보호가 주 목적)



사진자료 찾다가 생각했는데 그냥 쉽게 모자이크라고 생각하세요.

상속

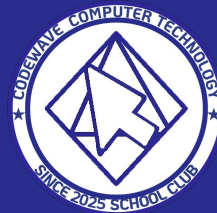


상속은 한마디로 부모 클래스의 속성과 기능을 물려받아서 새로운
자식 클래스를 정의하는 과정을 뜻합니다.

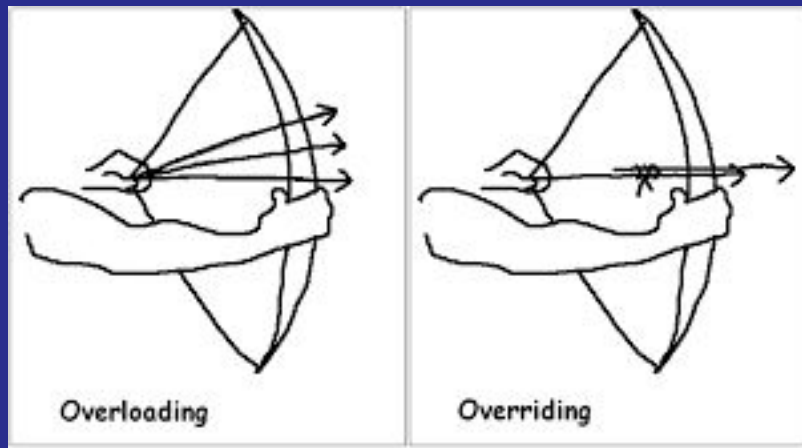


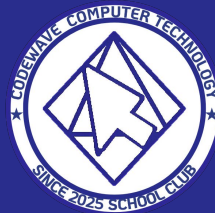
그냥 한마디로 족 ★ 보 ★ 형 ★ 성 이다 그 소리입니다.

다형성



다형성은 클래스 내부에 **같은 이름의 행위**를 여러개 정의하거나
상위 클래스의 행위를 **하위 클래스에서 재정의** 하여 사용하는 과정을
뜻합니다.





장점

클래스 단위로 개발하기 때문에 업무 분담이 편리하고 대규모 소프트웨어 개발에 적합하다.

클래스 단위로 수정이 가능하기 때문에 유지 보수가 편하다.

클래스를 재사용하거나 상속을 통해 확장하기 때문에 재사용이 용이하다.

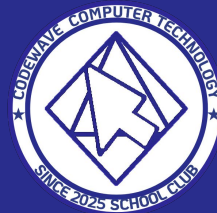
단점

처리 속도가 상대적으로 느리다.

객체의 수가 많아짐에 따라 용량이 커질 수 있다.

설계 시 많은 시간과 노력이 필요하게 될 수 있다.

수고하셨습니다



축하합니다! 당신은 프로그래머에 3% 가까워지셨습니다.

전진하세요.