**Day07 ||ヽ(\*￣▽￣\*)ノミ|Ю**

자바 상속의 특징

1. 코드의 재사용성: 추상화 ,공통코드를 한 곳에 두고 사용하려고
2. 제어용도: 모든 서브클래스를 슈퍼클래스로 제어하고 싶어서

함수의 재정의(Override): 슈퍼클래스의 함수를 재정의하면 슈퍼클래스의 함수를 보이지 않는 상태로 만든다.(not visible 상태로 변경됨). 그래서 재정의한 본인의 함수를 사용할 수 있는 것이다. 슈퍼클래스의 함수를 사용하려면 super를 사용해서 부를 수 있다. 본인 것은 this로

추상 메소드: 구현부 없이 머리만 틀만 작성하는 것

상속받은 서브클래스가 무조건 재정의 하도록 하기위해서 슈퍼클래스에서 추상 메소드를 작성한다.

🎳 추상 메소드

추상 메소드: 구현부 없이 머리만, 틀만 작성하는 것

상속받은 서브클래스가 무조건 재정의 하도록 하기 위해서 슈퍼클래스에서 추상 메소드를 작성한다. 이렇게 하면 강제성이 부여되기 때문에 모든 서브클래스가 본인만의 출력기능을 갖도록 할 수 있다. (아래 예시)

이렇게 하기 위해서는 클래스와 해당 메소드를 abstract를 작성해야 한다. 아래와 같이 Duck을 추상 클래스로 작성하고, display를 추상 메소드를 작성하였다.

🤔그렇다면 추상클래스를 왜 사용하는 것일까?

추상클래스는 상속을 통해 코드를 재사용하기보다는 제어를 하기위해서 사용하는 것이다.

🎹 자바의 배열

자바의 배열은 여러 개의 같은 종류의 데이터타입을 묶어서 관리하는 기술이다. 제약 조건이 같은 종류의 데이터만 묶어야 하는 것이다. 여러 변수를 하나로 묶어 하나의 이름으로 다룰 수 있게 물리적으로 연속된 메모리에 저장하는 구조이다. 배열명에는 전체 메모리 공간을 다루기 어려우니 시작주소만 저장해 사용하게 된다.

정적 배열: 실행 도중 배열의 크기를 변화시킬 수 없다. (사용자 입장에서 불편한 것

1. 배열의 값을 미리 알 때
2. 배열의 값이 미정일 때: 실행 시 결정되는 경우

* 선언과 동시에 값을 초기화해 생성
* New 연산자를 이용해 생성

동적 배열: 실행 도중 배열의 크기를 변화시킬 수 있다. (시스템 입장에서 불편한 것)

* Collection Framework에서 관련 기능을 제공한다.

🤔동적 배열이 시스템 입장에서 불편한 이유는?

왜냐하면 동적 배열은 크기가 정해져 있지 않기 때문에 우선 메모리를 할당해 놓고 값이 추가되면 다시 메모리 할당할 곳을 찾아 할당하고 이전 값을 모두 옮긴 후 사용할 수 있는 것이다. 즉 크기가 변화하면 계속해서 메모리 할당을 해야 하는 것이다.

정적 배열 : 초기값 미정 => int[] arr = new int[10];

int는 0으로, boolean은 false로 자동 초기화한다.

텍스트, 도표, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명텍스트, 스크린샷, 폰트, 도표이(가) 표시된 사진

자동 생성된 설명

🎹 final

final 변수, 함수, 클래스 앞에 사용가능. 보통 final키워드를 사용하면 public을 사용한다. 어차피 변경 불가능해 읽기 전용이 되었으니까.

변수: 값 변경 불가능

함수: 오버라이드 불가능

클래스: 상속 불가능, 서브클래스 생성 불가능