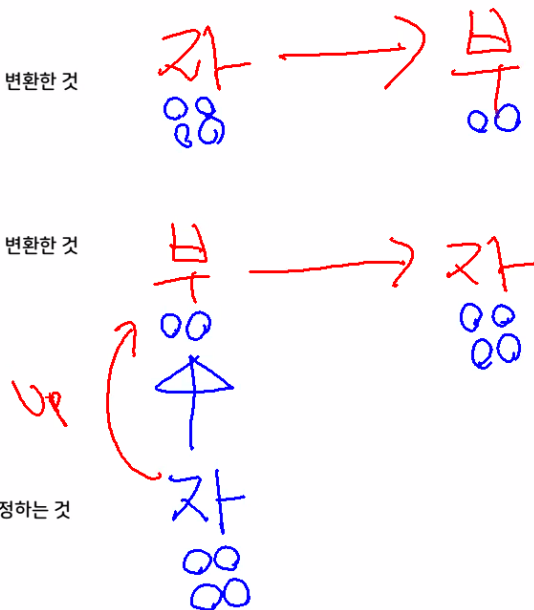


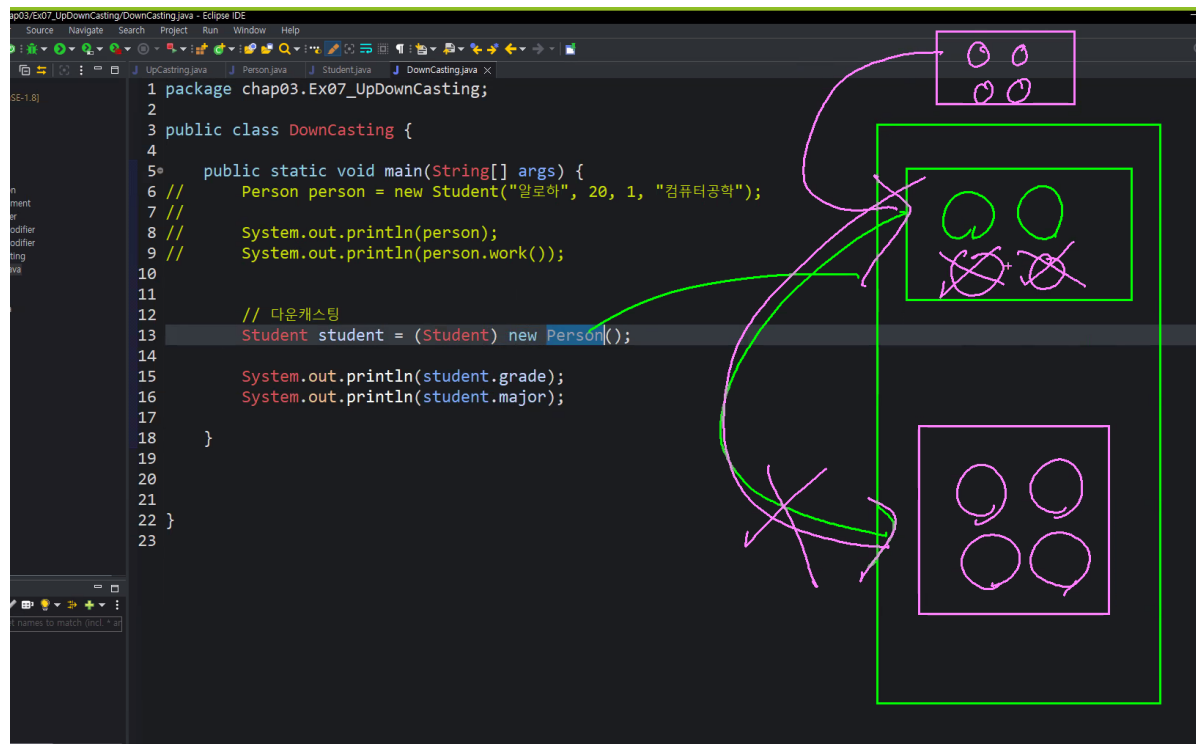
- fullcalendar api
- 카톡 알림 API
- 링크 채널로 연결해주거나
- 스크롤 /
- 알림톡 => 메일
- 업캐스팅 & 다운캐스팅
  - 작은 데이터 타입 -> 큰 데이터 타입 : 자동 형변환
  - 큰 데이터 타입 -> 작은 데이터 타입: 강제 형변환
- 클래스
  - 부모클래스 > 자식클래스
  - 자식클래스로 생성한 객체를 부모클래스 타입에 대입하는 것 => 업캐스팅
    - 업캐스팅한 객체로는 슈퍼클래스의 멤버(부모)만 접근 가능
  - 그 반대: 다운클래스
    - 슈퍼 클래스 객체의 레퍼런스를 서브 클래스 타입으로 변환한 것
  - 키워드: instanceof => a라는 객체가 부모로부터 왔는가, 자식으로부터 왔는지 확인 가능한 키워드
- 동적바인딩
  - 실행할 메소드를 컴파일 시 결정하지 않고 실행 시 결정하는 것
  - 오버라이딩된 메소드가 항상 실행

Chap.02 객체지향

## | 업캐스팅 & 다운캐스팅

- 업 캐스팅
  - 서브 클래스 객체의 레퍼런스를 슈퍼클래스 타입으로 변환한 것
  - 업캐스팅한 객체로는 슈퍼클래스의 멤버만 접근 가능
- 다운 캐스팅
  - 슈퍼 클래스 객체의 레퍼런스를 서브클래스 타입으로 변환한 것
  - 명시적으로 타입변환 필요
- 키워드 : instanceof
- 동적바인딩
  - 실행할 메소드를 컴파일 시 결정하지 않고 실행 시 결정하는 것
  - 오버라이딩된 메소드가 항상 실행
  - ⇄정적 바인딩





// Super Object

Sub Object

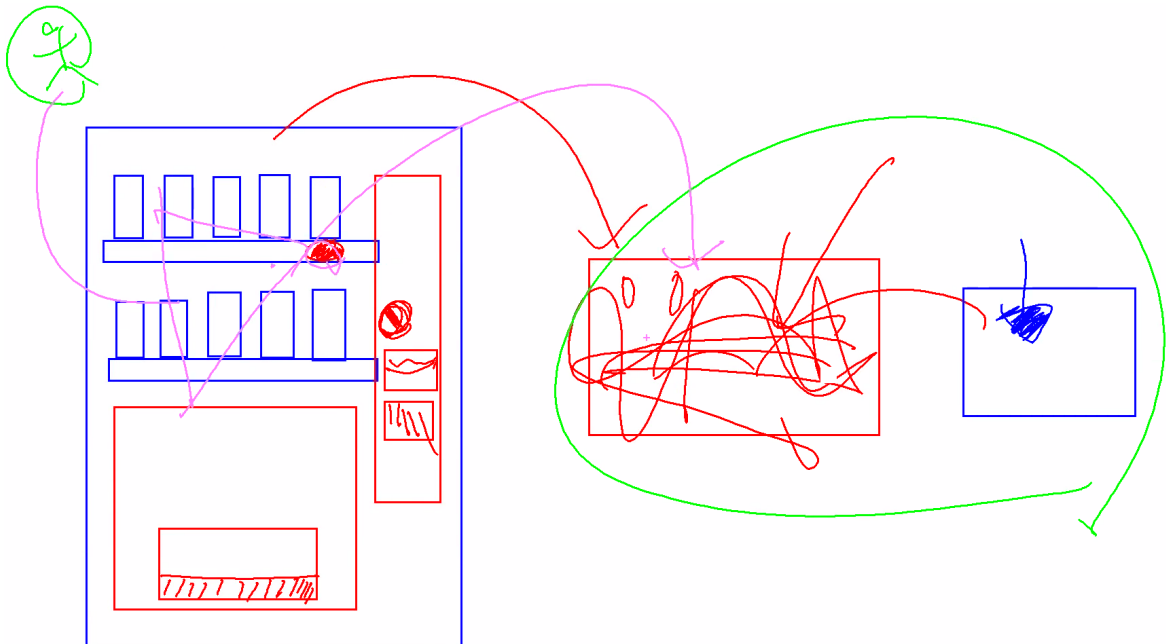
// 10

```

SubObject.java
3 class SuperObject {
4     int a = 10;
5
6     public void paint() {
7         draw();
8     }
9
10    public void draw() {
11        System.out.println("Super Object");
12    }
13 }
14
15 public class SubObject extends SuperObject {
16     int a = 20;
17
18    public void paint() {
19        draw();
20    }
21
22    public void draw() {
23        super.draw();
24        System.out.println("Sub Object");
25    }
26
27    public static void main(String[] args) {
28        SuperObject obj = new SubObject();
29
30        obj.paint();
31        System.out.println(obj.a);
32    }
33 }

```

추상 클래스를 상속받았다 하면 추상클래스의 메서드를 반드시 사용해야함



# | 인터페이스 (Interface)

\*서로 다른 장치들이 상호 데이터를 주고 받는 규격

- "abstract 메소드와 final 변수로만 이루어진 클래스"

- 키워드 : interface / implements

## • (특징)

- 인터페이스는 객체를 생성할 수 없다.
- 인터페이스 타입의 레퍼런스 변수는 선언 가능하다.
- 인터페이스끼리 상속 가능하다.
- 인터페이스를 상속받아 클래스를 작성하면, 반드시 모든 추상 메소드를 구현 해야한다.

\* (public static final) 변수

\* (public abstract) 메소드

