



강사 강태광

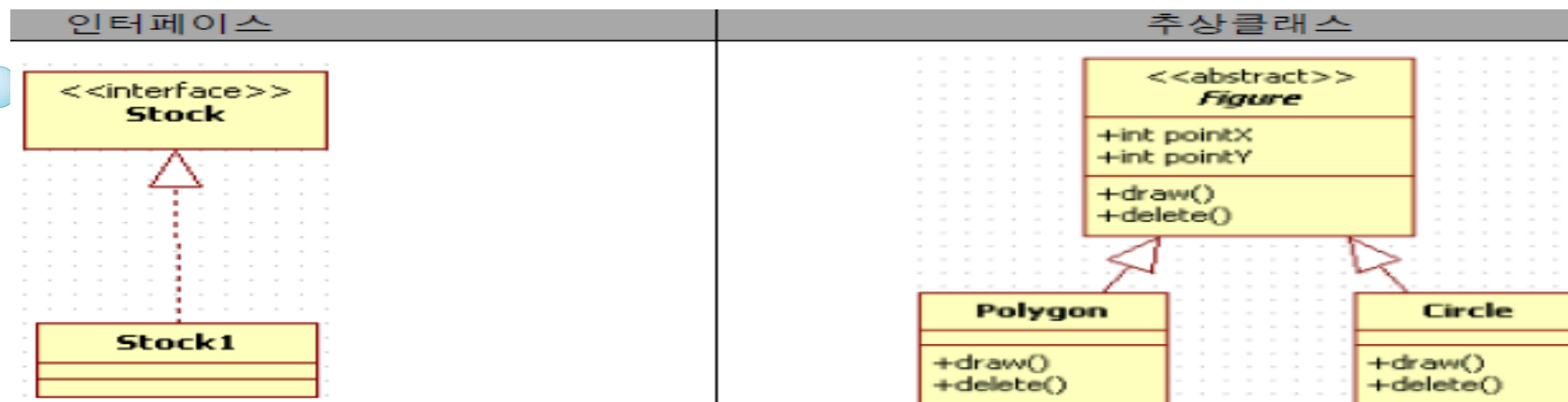
인터페이스 개념

1. 정의: -상수와 추상메서드의 집합체로 다른 클래스간의 공통 기능을 정의할 때 사용하는 클래스
 - 인터페이스를 사용하는 객체가 접근하는 구현되지 않은 메소드의 집합을 정의
2. 특징: 모든 멤버변수는 public static final(생략가능), 모든 메소드는 public abstract(생략가능)
Interface 인터페이스명으로 표기
3. 상속: 자바에서 다중 인터페이스 상속 가능, 상속받을 때 implement 사용

추상클래스 개념

1. 정의: 객체를 생성할 수 있는 공통 개념만을 표현하기 위한 인스턴스를 만들 수 없는 클래스
 - 하위 클래스를 만들기 위한 상위 클래스로서의 역할을 주목적으로 함
2. 특징:
 - 하나 이상의 추상 메소드 포함, 일반적인 메소드와 추상메소드 포함 가능, Abstract 클래스명으로 표기
3. 상속: 자바에서 단일 클래스에서만 상속 가능, 상속받을 때 extends 사용
 - 인터페이스와 추상클래스는 하위클래스에서 모든 추상 메소드를 구현하고 생성자를 갖지 못한다는 점에서는 동일하지만 추상클래스는 추상메소드 이외에 일반변수와 메소드를 갖는 점에서 차이점 있음

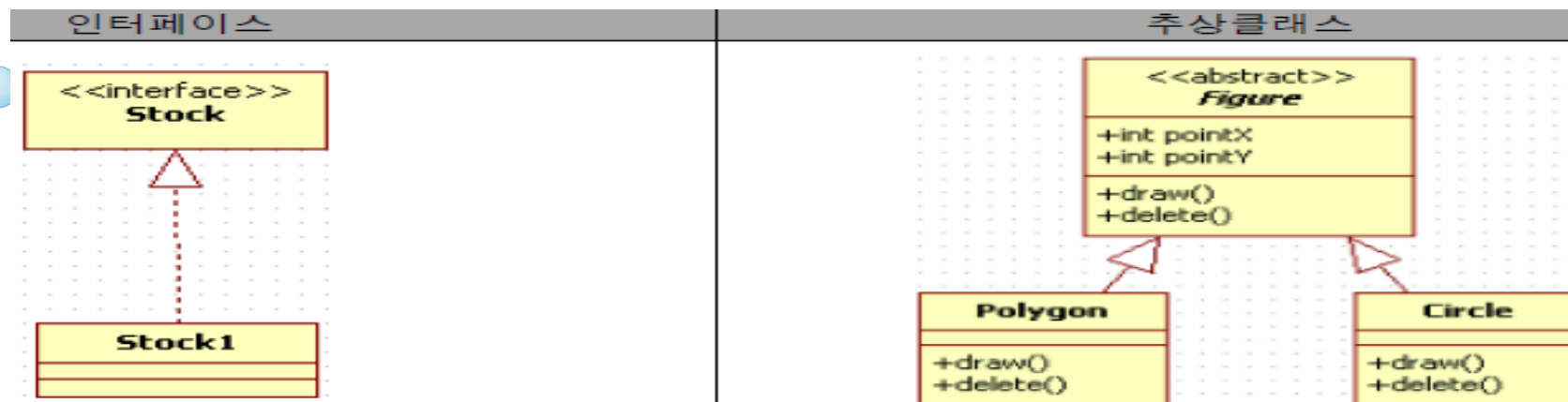
인터페이스와 추상클래스 도식 비교



나. 인터페이스와 추상 클래스 구현방법(코딩) 비교

인터페이스	추상클래스
<pre> public interface Stock { public void sell(); public void buy(); } public class Stock1 implements Stock { public void sell(); public void buy(); } </pre>	<pre> public abstract class Figure { public Object int pointX; public Object int pointY; public void draw() { } public void delete() { } } public class Polygon extends Figure { public void draw() { } public void delete() { } } public class Circle extends Figure { public void draw() { } public void delete() { } } </pre>
Interface '이름' 으로 정의 Implements로 하위클래스에서 상속	abstract class '이름' 으로 정의 extends로 하위클래스에서 상속

인터페이스와 추상클래스 도식 비교



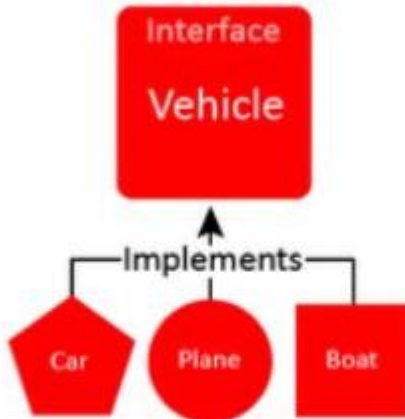
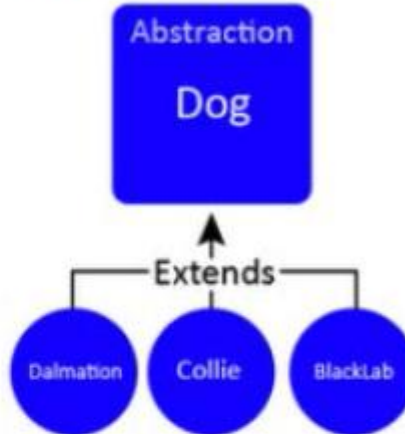
나. 인터페이스와 추상 클래스 구현방법(코딩) 비교

인터페이스	추상클래스
<pre> public interface Stock { public void sell(); public void buy(); } public class Stock1 implements Stock { public void sell(); public void buy(); } </pre>	<pre> public abstract class Figure { public Object int pointX; public Object int pointY; public void draw() { } public void delete() { } } public class Polygon extends Figure { public void draw() { } public void delete() { } } public class Circle extends Figure { public void draw() { } public void delete() { } } </pre>
Interface '이름' 으로 정의 Implements로 하위클래스에서 상속	abstract class '이름' 으로 정의 extends로 하위클래스에서 상속

인터페이스와 추상 클래스 사용 가이드라인

- ① 인터페이스: 추상 클래스와 달리 몸통을 갖춘 일반 메소드 또는 멤버변수를 구성원으로 가질 수 없음
오직 추상 메소드와 상수만을 멤버로 가짐 [개발시간을 단축하고 표준화 가능, 서로 관련없는 클래스들의 관계형성 가능, 독립적 프로그래밍 가능]
- ② 추상클래스: 추상 클래스를 이용해 객체 생성 불가능, 추상 클래스를 상속한 클래스는 반드시 추상 메소드를 구현해야 함
[추상 클래스는 인터페이스보다 느슨한 추상화를 제공하며 일반적인 클래스의 멤버를 가질 수 있음]

인터페이스와 추상클래스 사용 가이드라인

구분	인터페이스	추상클래스
개념도		
변수	상수(Static Final)만 가능	일반변수 가능
추상메소드	보유 가능	보유 가능
일반메소드	보유 불가능	보유 가능
다중상속	가능	불가능
장점	<ul style="list-style-type: none"> - 동시 개발로 개발기간 단축 - 표준화 가능 	<ul style="list-style-type: none"> - 재사용 부품을 이용하여 확장 가능 - 점진적 개발 용이