

Computer Programming 과제 #09

201402414 장수훈

소스코드 및 알고리즘	설명
<pre>package H1207; public abstract class Animal { protected int leg; public Animal(int leg) { this.leg = leg; } public abstract void eat(); public abstract void walk(); }</pre>	<p>필드엔 leg라는 정수형 변수를 생성하고 생성자를 통해 leg를 초기화 하였다. eat과 walk 추상메소드를 만든다.</p>

소스코드 및 알고리즘	설명
<pre>package H1207; public class Cat extends Animal implements Pet{ private String name; public Cat(String name) { super(4); this.name = name; } public Cat() { super(4); } public String getName() { return name; } public void setName(String name) { this.name = name; } public void play() { System.out.println("고양이가 공을 가지고 놀니다."); } public void eat() { System.out.println("고양이가 생선을 잘 먹습니다."); } public void walk() { System.out.println("이동물은 "+this.leg + "개의 다리로 걸습니다."); } }</pre>	<p>Animal 클래스와 Pet 인터페이스를 상속받는 Cat 클래스를 만들었다. 필드에는 name이라는 문자열 변수를 만들고 name을 가지는 생성자를 만들었다. leg는 슈퍼클래스를 호출해 4개로 저장하고 인터페이스들의 메소드들을 재정의 하였다.</p>

소스코드 및 알고리즘	설명
<pre>package H1207; public class Fish extends Animal implements Pet{ private String name; public Fish() { super(0); } public String getName() { return name; } public void setName(String name) { this.name = name; } public void eat() { System.out.println("물고기는 먹이를 먹습니다."); } public void walk() { System.out.println("이 동물은 헤엄칩니다."); } public void play() { System.out.println("물고기가 헤엄치며 놓니다."); } }</pre>	<p>이것또한 Animal과 Pet을 상속 받았다. leg를 초기화하기 위해 슈퍼클래스의 생성자를 호출하였 고 메소드 들을 구현 하였다.</p>

소스코드 및 알고리즘	설명
<pre>package H1207; public interface Pet { public abstract String getName(); public abstract void setName(String name); public abstract void play(); }</pre>	<p>과제 PPT에 있는 UML을 참고하여 getName과 setName과 play 추상 메소드를 정의한다.</p>

<pre>package H1207; public class Spider extends Animal{ public Spider () { super(8); } public void eat() { System.out.println("거미는 거미줄에 걸린 곤충을 먹습니다."); } public void walk() { System.out.println("이 동물은 "+this.leg+"개의 다리로 걸습니다."); } }</pre>	<p>Animal 클래스만 상속받는 클래스이다. 슈퍼클래스의 생성자를 호출하여 다리를 8개로 초기화 했다. eat, walk 메소드를 재정의 하였다.</p>
--	---

결과 값	결과 값 분석
<p>**FIsh의 메소드 호출</p> <p>물고기의 이름은 붕어싸만코입니다</p> <p>물고기가 헤엄치며 놀니다.</p> <p>물고기는 먹이를 먹습니다.</p> <p>이 동물은 헤엄칩니다.</p> <p>**Cat의 메소드 호출</p> <p>고양이의 이름은 나비입니다</p> <p>고양이의 이름을 페르시아(으)로 바꿨습니다.</p> <p>고양이가 공을 가지고 놀니다.</p> <p>고양이가 생선을 잘 먹습니다.</p> <p>이동물은 4개의 다리로 걷습니다.</p> <p>**Animal Fish의 메소드 호출</p> <p>물고기는 먹이를 먹습니다.</p> <p>이 동물은 헤엄칩니다.</p> <p>**Animal Spider의 메소드 호출</p> <p>거미는 거미줄에 걸린 곤충을 먹습니다.</p> <p>이 동물은 8개의 다리로 걷습니다.</p> <p>**Pet Cat의 메소드 호출</p> <p>고양이가 공을 가지고 놀니다.</p> <p>애완 동물의 이름을 맥킨토시(으)로 변경합니다.</p> <p>**객체에 대한 강제 변환</p> <p>물고기가 헤엄치며 놀니다.</p> <p>이 동물은 헤엄칩니다.</p> <p>물고기는 먹이를 먹습니다.</p> <p>**다형성 테스트</p> <p>이 동물은 8개의 다리로 걷습니다.</p> <p>거미는 거미줄에 걸린 곤충을 먹습니다.</p> <p>이동물은 4개의 다리로 걷습니다.</p> <p>고양이가 생선을 잘 먹습니다.</p> <p>이 동물은 헤엄칩니다.</p> <p>물고기는 먹이를 먹습니다.</p>	<p>첨부 파일에 있던 AnimalTest 클래스를 실행 시켜보았다.</p> <p>각가 클래스에 속하는 객체들을 만들고 메소드들을 있는 전부 실행 시키는 클래스 같다.</p> <p>물고기 이름과 고양이 이름을 접근자를 통해 붕어싸만코와 맥킨토시로 변경하였고</p> <p>다형성 테스트에서는 배열의 정의하고 for문을 이용해 walk메소드와 eat메소드를 하나씩 호출하였다.</p>