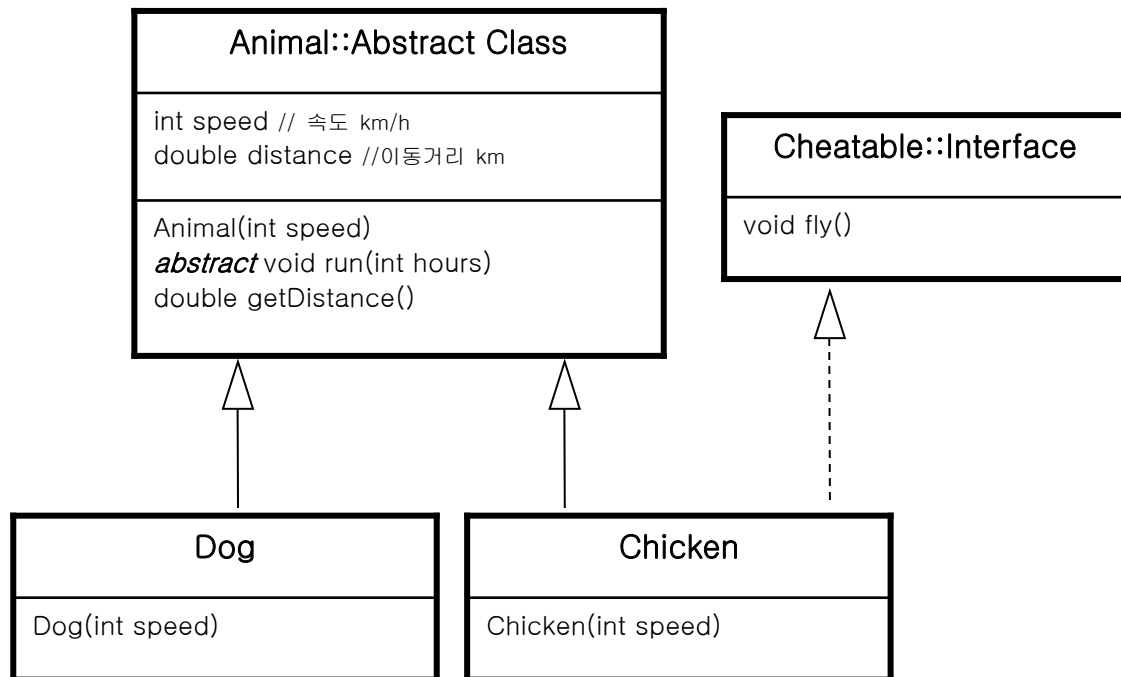


문제1. 그림과 같은 상속관계를 나타내고 있을 때, 아래 질문에 대한 클래스를 작성하십시오.



- I. 다음은 Dog 클래스와 Chicken 클래스의 구현에 필요한 요구사항들입니다.
- A. 이동거리는 run 함수가 호출될 때마다 누적되어 distance 변수에 저장되고, getDistance() 함수가 호출될 때 저장된 distance 값을 반환합니다.
 - B. fly 함수는 현재의 속도 값이 2배가 되도록 하는 기능을 수행합니다.
 - C. 또한, run 함수가 호출될 때 Chicken 객체는 입력받은 시간만큼 계속 이동하지만, Dog 객체는 입력인자로 넘겨받은 시간의 1/2 시간 동안만을 이동합니다. (즉, 속도가 10인 Dog object 가 1시간 동안 이동한 거리는 5가 되며, 속도가 3인 Chicken object가 1시간 동안 이동한 거리는 3이 됩니다.)

상기 설명을 참고하여 아래에 제시된 클래스 및 인터페이스를 작성해 주십시오.

- A. Animal 추상 클래스를 작성하십시오.
- B. Cheatable 인터페이스를 작성하십시오.
- C. Animal 추상클래스를 상속받은 Dog 클래스를 작성하십시오.
- D. Animal 추상클래스를 상속받고 Cheatable 인터페이스를 구현한 Chicken 클래스를 작성하십시오.

(참고: distance의 계산 시 이동거리=속도X시간 이므로 distance = speed X hours 등식을 사용하여 계산하면 됩니다.)

- I. 아래에 제시된 시나리오와 출력결과를 참조하여, AnimalTest 클래스를 완성하십시오. (10 점)
- A. 템플릿 코드에 선언된 dog 변수에 속도가 8인 Dog Object를 하나 생성하여 저장합니다.
 - B. 템플릿 코드에 선언된 chicken 변수에 속도가 3인 Chicken Object를 하나 생성하여 저장합니다.
 - C. 템플릿 코드에 선언된 cheatableChicken 변수에 속도가 5인 Chicken Object를 하나 생성하여 저장합니다.
 - D. 위에서 생성한 Chicken Object가 Cheatable 하다면 fly 함수를 호출합니다. 단, Cheatable 여부의 판단을 위해서는 반드시 생성된 Object의 타입 검사를 수행해야 합니다.
 - E. 3시간 동안 매 시간마다 각 객체들의 이동거리 정보가 아래와 같이 출력되도록 합니다.

화면 출력(AnimalTest 실행):

1시간 후

개의 이동거리=4.0

닭의 이동거리=3.0

날으는 닭의 이동거리=10.0

2시간 후

개의 이동거리=8.0

닭의 이동거리=6.0

날으는 닭의 이동거리=20.0

3시간 후

개의 이동거리=12.0

닭의 이동거리=9.0

날으는 닭의 이동거리=30.0