**Abstract 클래스에서 상속 받은 두 개의 클래스를 구현 하여 5번의 실행 결과가 나오도록**

**프로그램을 작성 한다.**

1. 사용 데이터

|  |  |  |
| --- | --- | --- |
| **class Type** | **planeName** | **fuelSize** |
| Airplane | L747 | 1000 |
| Cargoplane | C40 | 1000 |

2. 클래스 다이어그램



**Airplane**

+Airplane()

+Airplane(planeName:String, fuelSize:int)

+flight(distance:int):void

**Cargoplane**

+Cargoplane()

+Cargoplane(planeName:String, fuelSize:int)

+flight(distance:int):void

+flight(distance:int):void // 운항

3. 구현 클래스

|  |  |  |  |
| --- | --- | --- | --- |
| **패키지명** | **클래스명** | **메소드** | **설명** |
| work | Plane | +Plane() | 기본 생성자 |
| +Plane(planeName:String, fuelSize:int) | 2개의 데이터 값을 받는 생성자 |
| *+flight(distance:int):void* | 일정 거리 만큼 운항, 연료 감소 |
| +refuel(fuel:int):void | 일정 양의 연료 주입, 기존 연료가 증가됨 |
| Airplane | +Airplane() | 기본 생성자 |
| +Airplane(planeName:String,fuelSize:int) | 2개의 데이터 값을 받는 생성자 |
| +flight(distance:int):void | 일정 거리 만큼 운항, 연료 감소  10 운항 시 연료 30감소 |
| Cargoplane | +Cargoplane() | 기본 생성자 |
| +Cargoplane(planeName:String,  fuelSize:int) | 2개의 데이터 값을 받는 생성자 |
| +flight(distance:int):void | 일정 거리 만큼 운항, 연료 감소  10운항 시 연료 50감소 |
| PlaneTest | + main(args:String[]):void | main 메서드 |

4. 클래스 구조

|  |
| --- |
| public class PlaneTest {  public static void main(String args[]) {  ***// 2개의 원소를 갖는 Plane 타입의 배열 객체를 생성***  ***// Airplane과 Cargoplane 객체를 생성하여 각 원소로 저장***  *// 생성된 객체의 정보 출력*  *// Airplane과 Cargoplane 객체에 100씩 운항 후 객체 정보 출력*  *// Airplane과 Cargoplane 객체에 200 주유 후 객체 정보 출력*  }  public static void printInfo(Plane[] list) {  *// 타이틀 출력*  *// Plane 객체들의 데이터 출력(출력 결과 참조)*  }  } |

5. 실행 결과 예

**Plane fuelSize**

**--------------------------------**

**L747 1000**

**C40 1000**

**[100 운항]**

**Plane fuelSize**

**--------------------------------**

**L747 700**

**C40 500**

**[200 주유]**

**Plane fuelSize**

**--------------------------------**

**L747 900**

**C40 700**