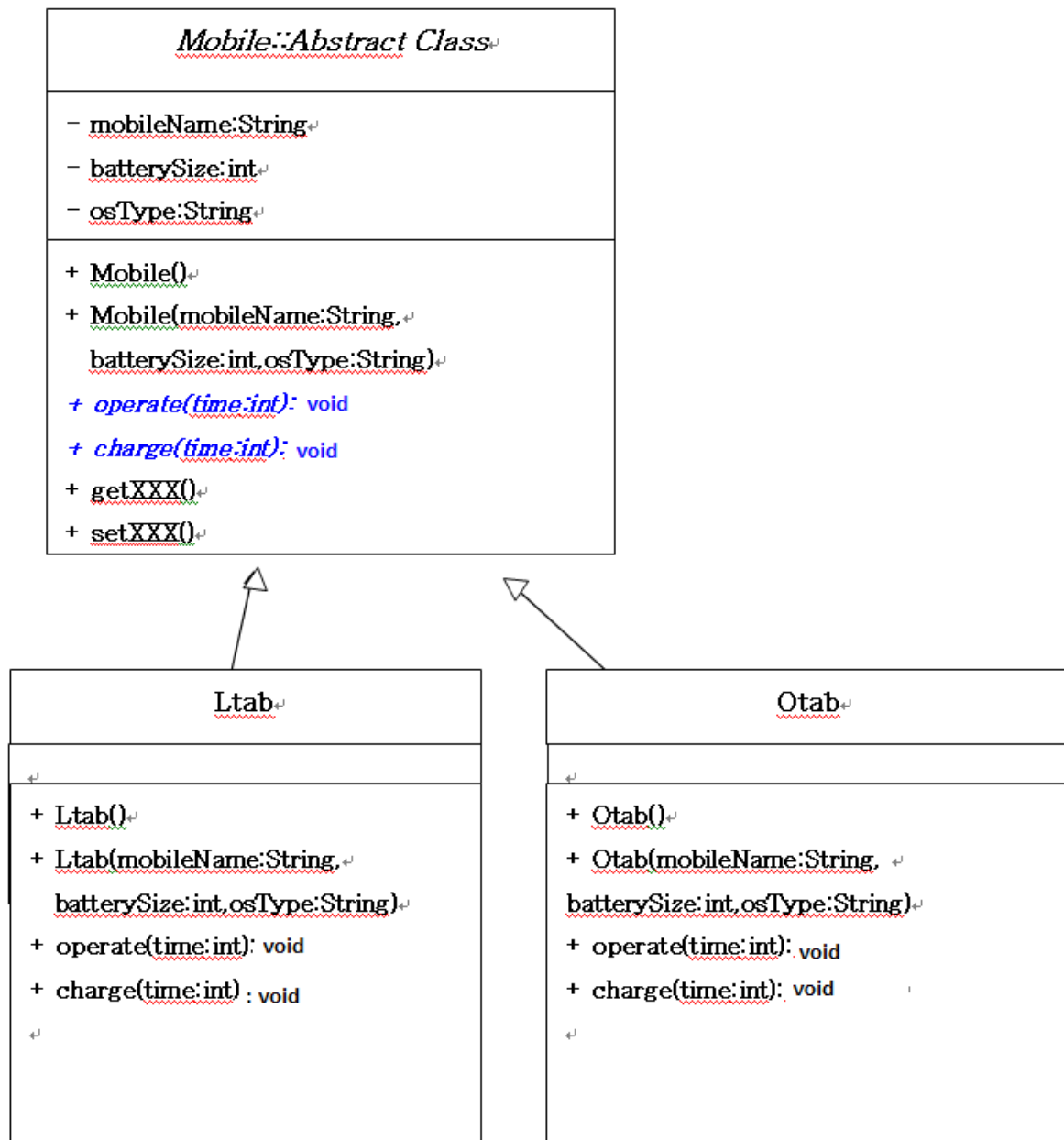


Abstract 클래스에서 상속 받은 두 개의 클래스를 구현 하는 프로그램을 작성 한다

1. 사용 데이터

mobileName	batterySize	osType
Ltab	500	ABC-01
Otab	1000	XYZ-20

2. 클래스 다이어그램



3. 구현 클래스

Package명	클래스명	메소드	설명
mobile	Mobile	+Mobile()	매개변수 없는 생성자
		+Mobile(mobileName:String, batterySize:int,osType:String)	3개의 멤버 변수의 값을 입력 받는 생성자
		<i>+operate(time:int):void</i>	abstract 메서드로 정의
		<i>+charge(time:int):void</i>	abstract 메서드로 정의
	Ltab	+Ltab()	매개변수 없는 생성자
		+Ltab(mobileName:String, batterySize:int,osType:String)	3개의 멤버 변수의 값을 입력 받는 생성자
		+operate(time:int):void	사용을 통해 배터리 감소 구현, 1분 사용 시 배터리 10감소
		+charge(time:int):void	충전을 통한 배터리 증가 구현 1분 충전 시 배터리 10증가
	Otab	+Otab()	매개변수 없는 생성자
		+Otab(mobileName:String, batterySize:int,osType:String)	3개의 멤버 변수의 값을 입력 받는 생성자
		+operate(time:int):void	사용을 통해 배터리 감소 구현 1분 사용 시 배터리 12감소
		+charge(time:int):void	충전을 통한 배터리 증가 구현 1분 충전 시 배터리 8증가

* 필요 시 getXXX와 setXXX는 자율적으로 구현한다

4. MobileTest 클래스 구조

```
public class MobileTest {  
    public static void main(String args[]) {  
        // 각각의 Mobile 객체를 생성하여 Mobile 배열에 저장한다.  
        // 생성된 객체의 정보를 출력한다.(printMobile() 호출)  
        // 각각의 Mobile 객체에 10분씩 충전을 한다.  
        // 10분 충전 후 객체 정보를 출력한다.(printMobile() 호출)  
        // 각각의 Mobile 객체에 5분씩 통화를 한다.  
        // 5분 통화 후 객체 정보를 출력한다.(printMobile() 호출)  
    }  
    public static void printMobile(Mobile[] mobile) {  
  
    }  
    public static void printTitle() {  
  
    }  
}
```

5. 실행 결과 예

Mobile	Battery	OS
Ltab	500	ABC-01
Otab	1000	XYZ-20
[10분 충전]		
Mobile	Battery	OS
Ltab	600	ABC-01
Otab	1080	XYZ-20
[5분 통화]		
Mobile	Battery	OS
Ltab	550	ABC-01
Otab	1020	XYZ-20