

다음과 같은 조건을 만족하는 프로그램을 작성 하시오

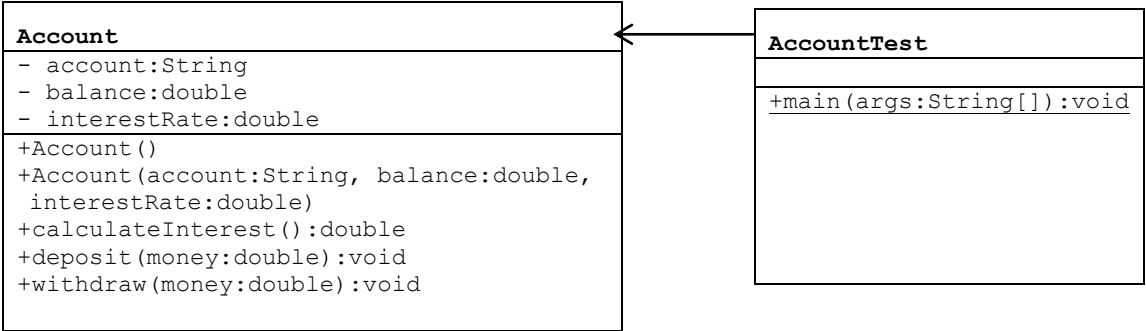
1. 조건

고정 금리의 계좌를 생성 하여 현재 잔액을 기준으로 이자를 계산하는 프로그램을 작성 한다.

사용 데이터

Account(계좌번호): 441-0290-1203, balance(잔액): 500000원, interestRate(이율): 7.3%

2. 구현 클래스



3. 구현 클래스

Package명	Class명	Method	설명
workshop6.exception	Account	+Account()	기본 생성자
		+Account(account:String, balance:double, interestRate:double)	3개의 클래스변수를 받는 생성자
		+calculateInterest():double	현재 잔액을 기준으로 이자를 계산 한다
		+deposit(money:double):void	입금을 통해 잔액정보를 증가 입금액이 0보다 적으면 Exception발생을 발생 시킨다.
		+withdraw(money:double):void	출금을 통해 잔액정보를 감소 출금액이 0보다 적으면 Exception을 발생 시킨다. 출금액이 현재 잔액 보다 많으면 Exception 을 발생 시킨다.
	AccountTest	+main(args:String[]): void	main 함수 안에서 Account객체 변수를 선언하고 생성 하여 동작 시킨다 Exception 발생 시 try, catch를 통해 처리

- * class 명과 method 명은 변경 하지 않는다
- * 위에 선언한 클래스 변수와 클래스 함수만을 이용한다
- * getXXX/setXXX는 필요 시 만들어서 사용한다

4. AccountTest 클래스 구조

```
public class AccountTest {
    public static void main(String args[]) {
        Account account;
        // account 객체 생성
        // account 기본 정보 출력
    }
}
```

```

        // account 예 -10원 입금 - Exception 처리
        // account 예 600000원 출금 - Exception 처리
        // 이자 출력 - 현재 잔고를 기준으로 고객에게 줄 이자 금액을 출력 한다
    }
}

```

```

public class Account {
    // 클래스 변수 선언
    // Constructor 구현
    // getter/setter 구현

    public double calculateInterest() {
        // 이자계산
    }
    public void deposit(double money) throws Exception {
        // 입금 처리 및 예외상황 처리
    }
    public void withdraw(double money) throws Exception {
        // 출금 처리 및 예외상황 처리
    }
}

```

5. 실행 결과

이자 계산: 현재잔고 x 이자율

실행 결과 예)

- 1) 입금 금액이 0보다 적을 때 "입금 금액이 0보다 적습니다."라고 출력
- 2) 출금 금액이 0보다 적거나 현재 잔액보다 많을 때 "출금 금액이 0보다 적거나 현재 잔액보다 많습니다."라고 출력

```

계좌정보: 441-0290-1203 500000.0 7.3
입금 금액이 0보다 적습니다.
금액이 0보다 적거나 현재 잔액보다 많습니다.
이자: 36500.0

```