

1. 지불할 금액을 정수로 입력 받아 화폐단위가 각각 몇 개씩 필요한지 출력하는 프로그램을 작성하시오. 입력은 최소 천원 단위로 입력, 화폐단위는 50000, 10000, 5000, 1000 4가지이며, 가능한 큰 화폐단위로 지불, 입력이 236,000이면 50000원권 4개, 10000원권 3개, 5000원권 1개, 1000원권 1개로 출력하라.

2. 양의 정수를 입력하고 자리수를 출력하는 프로그램을 작성하시오. 예) “123”을 입력하면 “3자리수입니다”, “1234”를 입력하면 4자입니다.”라고 출력되도록 한다.

3. 세 값의 대소 관계의 가장 작은 값을 구하라.

4. 소문자 알파벳을 하나 입력받아서 다음과 같은 결과가 나타나도록 코드를 작성하시오.

소문자 알파벳 하나를 입력하시오>> e

abcde

abcd

abc

ab

a

5. 다음 그림과 같이 위쪽과 왼쪽 숫자를 곱한 곱셈표를 출력하는 프로그램을 작성하시오

```

  | 1  2  3  4  5  6  7  8  9
-----+-----
1 | 1  2  3  4  5  6  7  8  9
2 | 2  4  6  8 10 12 14 16 18
3 | 3  6  9 12 15 18 21 24 27
   ~
9 | 9 18 27 36 45 54 63 72 81

```

6. 다음과 같은 비정방 배열을 만든 후, 다음 배열 값을 출력하면서 각 행의 합과 평균도 함께 출력하는 프로그램을 작성하시오.

78	48	78	98
99	92		
29	64	83	
34	78	92	56

7. 배열을 사용하여 다음의 숫자를 정렬하라.

15, 7, 2, 9, 10, 12, 17, 11, 20, 5

1 : 2, 7, 15, 9, 10, 12, 17, 11, 20, 5

2 : 2, 5, 15, 9, 10, 12, 17, 11, 20, 7

3 : 2, 5, 7, 9, 10, 12, 17, 11, 20, 15

4 : 2, 5, 7, 9, 10, 12, 17, 11, 20, 15

5 : 2, 5, 7, 9, 10, 12, 17, 11, 20, 15

6 : 2, 5, 7, 9, 10, 11, 17, 12, 20, 15

7 : 2, 5, 7, 9, 10, 11, 12, 17, 20, 15

8 : 2, 5, 7, 9, 10, 11, 12, 15, 20, 17
 9 : 2, 5, 7, 9, 10, 11, 12, 15, 17, 20
 10: 2, 5, 7, 9, 10, 11, 12, 15, 17, 20

8. 다음 요구사항을 만족하는 Album 클래스를 정의하시오.

1. Album 클래스는 앨범명(title), 가수(artist), 발매연도(year), 장르(genre)를 나타내는 필드를 가진다.
2. 모든 필드는 private으로 선언한다.
3. 각 필드에 대해 접근자(getter)와 설정자(setter) 메소드를 정의하고, 이를 통해 필드 값을 읽고 수정할 수 있도록 한다.
4. 생성자는 default 생성자, 모든 필드를 초기화하는 생성자, 앨범명만 초기화하는 생성자를 작성한다.
5. toString 메소드를 작성한다.

main 메소드에서는 세 가지 생성자를 모두 사용하여 객체를 생성하고, 아래와 같이 출력되도록 작성한다.

앨범명	가수	발매연도	장르
Thriller	Michael Jackson	1982	Pop
Divide	Ed Sheeran	2017	Pop
The Dark Side	Pink Floyd	1973	Rock

9. 다음의 요구사항을 만족하는 전자지갑(EWallet) 클래스를 작성하시오.

1. 필드 정의

- 지갑ID(walletID) : 지갑 고유 번호 (private)
- 소유자(owner) : 지갑 소유자 이름 (private)
- 잔액(balance) : 지갑 잔액 (private)
- 거래 내역(transactionHistory) : 거래 기록을 저장하는 리스트 (private)
- 최소잔액(minBalance) : 출금이나 송금 시 유지해야 하는 최소 잔액 (private, 기본값 1000원)

2. 생성자 작성

- 기본 생성자 : 모든 필드를 기본값으로 초기화
- 모든 필드 초기화 생성자
- 소유자만 초기화하는 생성자

3. 다음 메소드를 작성한다.

- deposit(int amount) : 금액을 충전하고 거래 내역에 기록, 음수 금액은 예외 처리
- withdraw(int amount) : 금액을 출금하고 거래 내역에 기록, 잔액이 최소 잔액보다 적으면 출금 불가, 음수 금액은 예외 처리
- transfer(int amount, EWallet otherWallet) : 다른 지갑으로 송금, 송금 시 일정 수수료(fee)를 적용 (예: 1%), 잔액이 최소잔액 이하로 떨어지면 송금 불가, 거래 내역에 송금과 수신 기록 남김
- getBalance() : 현재 잔액 반환
- printTransactionHistory() : 거래 내역 출력

4. 예외 처리

음수 금액 예외, 잔액 부족 예외 (InsufficientBalanceException)

5. 메인 테스트

- 2개의 지갑 객체를 생성
- 입금, 출금, 잔액 조회, 송금 기능을 테스트
- 거래 내역을 출력하여 정상 동작 확인

예시 출력

지갑ID: W001, 소유자: Alice, 잔액: 5000

지갑ID: W002, 소유자: Bob, 잔액: 3000

Alice 지갑에서 Bob 지갑으로 2000 송금 완료 (수수료 20)

지갑ID: W001, 소유자: Alice, 잔액: 2980

지갑ID: W002, 소유자: Bob, 잔액: 5000

Alice 거래 내역:

입금 5000

송금 2000 -> Bob (수수료 20)

Bob 거래 내역:

입금 3000

수신 2000 <- Alice

10. Student & Professor 객체 관리 프로그램을 다음의 요구사항을 만족하는 프로그램을 작성하시오.

1. 클래스 정의

1) Person 클래스 (기본 클래스)

- 필드: name(이름), age(나이), gender
- 접근자(getter)와 설정자(setter) 메소드 작성
- 생성자: 기본 생성자, 모든 필드 초기화 생성자

2) Student 클래스 (Person 상속)

- 추가 필드: studentID(학번), major(전공)
- 생성자: 기본생성자, 모든 필드 초기화
- getter/setter 작성

3) Professor 클래스 (Person 상속)

- 추가 필드: professorID(교수번호), department(학과)
- 생성자: 모든 필드 초기화
- getter/setter 작성

2. 프로그램 요구사항

1) ArrayList<Person>과 HashMap<String, Person>을 사용하여 데이터를 저장

2) 사용자 메뉴 : 1. 데이터입력 | 2. 데이터조회 | 3. 전체데이터출력 | 4. 종료 >>

3) 메뉴 기능

- 데이터 입력

객체 종류 선택: 1. Student, 2. Professor

Student: 이름, 나이, 성별, 학번, 전공 입력

Professor: 이름, 나이, 성별, 교수번호, 학과 입력

"그만" 입력 시 입력 종료

- 데이터조회
 검색할 이름 입력
 동일한 이름의 Student/Professor 정보 출력
 "그만" 입력 시 조회 종료
- 전체데이터출력 : ArrayList 또는 HashMap에 저장된 모든 Student/Professor 데이터 출력
- 종료 : 프로그램 종료

3. 실행 예시

1. 데이터입력 | 2. 데이터조회 | 3. 전체데이터출력 | 4. 종료 >> 1

객체 종류 선택 (1. Student, 2. Professor) >> 1

이름, 나이, 성별, 학번, 전공 입력 >> 김철수, 20, 남, s001, 컴퓨터공학

객체 종류 선택 (1. Student, 2. Professor) >> 2

이름, 나이, 성별, 교수번호, 학과 입력 >> 이영희, 45, 여, p001, 전자공학

객체 종류 선택 (1. Student, 2. Professor) >> 그만

1. 데이터입력 | 2. 데이터조회 | 3. 전체데이터출력 | 4. 종료 >> 2

검색할 이름 입력 >> 김철수

김철수 20 남 s001 컴퓨터공학

검색할 이름 입력 >> 이영희

이영희 45 여 p001 전자공학

검색할 이름 입력 >> 그만

1. 데이터입력 | 2. 데이터조회 | 3. 전체데이터출력 | 4. 종료 >> 3

김철수 20 남 s001 컴퓨터공학

이영희 45 여 p001 전자공학

1. 데이터입력 | 2. 데이터조회 | 3. 전체데이터출력 | 4. 종료 >> 4

프로그램 종료