

## EXERCISE

### INTRODUCTION TO JAVA PROGRAMMING

1. 다음과 같은 배열을 생성하는 문장을 작성하시오.

- ① 30개의 원소를 가지는 int형 배열로 studentNumbers 변수로 참조된다.
- ② 100개의 원소를 가지는 double형 배열로 profits 변수로 참조된다.
- ③ 2000개의 원소를 가지는 char형 배열로 proverbs 변수로 참조된다.
- ④ 1.2, 3.3, 6.7과 같은 값을 가지는 double형 배열을 생성하고 초기화한다.

2. 다음의 배열 선언에서 잘못된 점을 지적하시오.

- ① `int[] numbers = new numbers[100];`
- ② `double rainfalls[100] = new double();`

3. `int[] employees = new int[5];`와 같은 배열 선언이 있다고 하자.

- ① 유효한 인덱스의 범위는?
- ② 만약 `employees[5]`와 같이 원소를 접근하였다면 어떤 일이 발생하는가?

4. 다음 문장의 출력을 쓰시오.

```
int[] numbers = new int[5];
for(int i = 0; i < 5; i++)
    numbers[i] = i + 1;
for(int i: numbers)
    System.out.println(numbers[i]);
```

5. 프로그램이 다음과 같은 선언을 가지고 있다.

```
double[] readings;
```

사용자에게 배열의 크기를 물어보고 그 크기대로 배열을 생성하는 문장을 작성하라.

6. 만약 어떤 values라는 double형 배열이 init() 메소드의 매개 변수로 넘어왔다고 하자. 배열의 각 원소를 0으로 초기화하는 문장을 작성하라.

```
void init(double[] values)
{
    // 여기에 배열의 원소를 0으로 초기화하는 문장을 작성하라
}
```

7. 어떤 책방에서 60개의 책꽂이를 가지고 있다. 각 책꽂이에는 10개의 선반이 있고 각 선반은 30권의 책을 진열할 수 있다.

- ① 이 책방에 진열된 모든 책의 번호를 저장할 수 있는 3차원 배열을 생성하라.
- ② 3차원 배열의 모든 원소를 0으로 초기화하는 반복 루프를 작성하라.

8. 다음 코드는 배열 a를 배열 b로 복사하려는 의도로 작성되었다. 실제 배열이 복사 되는가? 만약 올바르게 없다면 올바르게 수정하라.

```
int[] a = { 1, 2, 3, 4, 5 };
int[] b = new int[5];
b = a; // 배열 a를 배열 b로 복사
```

9. 다음 코드는 배열 a와 배열 b의 내용이 일치하는지를 알아보기 위하여 작성되었다. 배열이 올바르게 비교되는가? 만약 올바르게 없다면 올바르게 수정하라.

```
int[] a = { 1, 2, 3, 4, 5 };
int[] b = { 6, 7, 8, 9, 10 };
if( a == b )
    System.out.println("배열의 내용이 일치합니다.");
else
    System.out.println("배열의 내용이 일치하지 않습니다.");
```

## PROGRAMMING

### INTRODUCTION TO JAVA PROGRAMMING

1. 12개월 동안의 카드 사용 금액을 `double`형의 배열에 저장하는 클래스 `Credit-Card`를 설계하라. 프로그램은 다음과 같은 정보를 반환하는 메소드를 가져야 한다. 완전한 프로그램을 작성하여 테스트하라. 사용자로부터 월별 사용 금액을 입력받을 때는 음수를 체크하도록 하라.

- 1년 동안의 전체 사용 금액
- 월별 평균 사용 금액
- 가장 지출이 많았던 월
- 가장 지출이 적었던 월

2. 객관식 문제의 시험 점수를 채점하는 클래스 `Exam`를 작성하여 보자. `Exam`에는 10개의 객관식 문제와 답이 저장되어 있다. 예를 들면 다음과 같다.

① 문제: “캐나다의 수도는?”

(a) 밴쿠버      (b) 토론토      (c) 오타와      (d) 재스퍼

답: (c) 오타와

...

학생들은 시험을 통과하기 위하여 7문제 이상을 맞추어야 한다. `Exam` 클래스 안에 배열을 선언하고 문제와 답을 저장한 후에 학생들에게 문제를 제시하고 학생들의 답도 역시 배열에 저장한다. 다음과 같은 메소드를 제공하여야 한다.

- `isPassed()`: 학생이 시험에 통과했는지를 `true`와 `false`로 반환한다.
- `correctAnswers()`: 정답 횟수
- `incorrectAnswers()`: 오답 횟수

완전한 프로그램을 작성하여서 `Exam` 클래스를 테스트하도록 하고 학생들의 답변은 a, b, c, d만 입력이 가능하도록 프로그램을 작성하라.

3. 1차원 정수 배열을 입력으로 받아서 다음과 같은 계산을 하는 메소드를 작성하고 테스트하라.

- getTotal(): 1차원 배열 원소의 합계를 구한다.
- getAverage(): 1차원 배열 원소의 평균을 구한다.
- getMaximum(): 1차원 배열 원소 중에서 가장 높은 값을 반환한다.
- getMinimum(): 1차원 배열 원소 중에서 가장 낮은 값을 반환한다.

4. 어떤 회사의 분기별 매출액을 저장하는 클래스 Sales를 작성하여 보자. 회사에는 5개의 부서가 있고 각 부서의 매출액이 이차원 배열에 저장된다. 매출액은 사용자가 입력하도록 하고 이때 음수가 입력되면 안 된다. 다음과 같은 메소드를 제공하도록 하자.

- 분기별 부서별 매출액을 출력
- 분기별 회사 전체의 매출액을 출력
- 분기별 회사 전체의 평균 매출액을 출력
- 특정 분기에 최고의 매출을 올린 부서를 출력

## LAB

## INTRODUCTION TO JAVA PROGRAMMING

1. 사용자로부터 정수를 입력받아서 배열에 저장하고 출력하는 프로그램을 작성하여 보자.

```
class TestArray {
    public static void main(String args[])
    {
        Scanner scan = new Scanner(System.in);
        // 10개의 정수를 저장할 수 있는 배열을 선언하고 생성하라
        _____;
        // 사용자로부터 정수를 받아서 배열에 저장하는 반복 루프
        _____;
        _____;
        // 배열에 저장된 정수를 출력하는 반복 루프
        _____;
        _____;
    }
}
```

- ① 위의 프로그램의 빈칸을 채우고 실행하여 보라.
  - ② 배열에 저장된 정수를 출력할 때, for-each 반복 구조를 사용하여 보라.
  - ③ 배열에 저장된 정수를 역순으로 출력하여 보라.
2. 사용자가 입력한 문자열에서 각각의 문자들이 나타나는 횟수를 계산하는 프로그램을 작성하여 보자.

```
class CountLetter {
    public static void main(String args[])
    {
        Scanner scan = new Scanner(System.in);
        // 26개의 정수를 저장할 수 있는 배열 count를 선언하고 생성하라
        _____;

        System.out.println("문자열을 입력하시오: ");
    }
}
```



```

String buffer = scan.nextLine();

// 각 문자가 등장하는 회수를 계산한다
for(int i; i< buffer.length(); i++) {
    char ch = buffer.charAt(i);
    count[ch-'a']++;
}
// 배열에 저장된 횟수를 출력하는 반복 루프
_____
}
}

```

① 대문자도 카운트되도록 소스를 변경하라.

3. 배열에 대하여 실습하여 보자. 회사에서 직원들을 관리하는 프로그램을 작성하여 보자. 직원들의 이름, 주소, 연봉, 전화번호를 가지고 있다. 직원은 Employee 클래스로 표현된다. 회사 전체의 직원은 Employee의 배열 안에 저장된다.

① 먼저 Employee 클래스를 다음과 같이 작성하여 보자.

```

class Employee {
    String name; // 직원의 이름
    _____; // 주소
    _____; // 연봉
    _____; // 전화 번호
    // 각각의 필드에 대하여 getter와 setter 메소드를 작성하라
    ...
    // toString() 메소드를 작성하여 본다
    public void toString()
    {
    }
}
}

```

② Employee 클래스를 테스트하기 위하여 클래스를 하나 더 작성하여 보자.

```

class TestEmployee {
    public static void main(String args[])
    {
        Scanner scan = new Scanner(System.in);
        // 배열 참조 변수를 선언한다
    }
}

```

```
_____;  
// 배열을 생성한다  
_____;  
// 사용자로부터 데이터를 받아서 배열에 추가하여 본다  
...  
// ❶  
// 배열에 저장된 모든 데이터를 출력한다  
...  
}  
}
```

- ❸ 사용자로부터 배열의 내용을 수정할 수 있는 메뉴 시스템을 ❶ 위치에 추가하여 보라. 사용자로부터 직원 번호를 받은 후에 새로운 데이터를 입력받아서 배열의 내용을 수정한다.

