

## 복습 문제

### 객체지향 프로그래밍

#### < day 5 (조지수) >

##### 문제1 : 객체 지향 프로그래밍에서 **Getter**의 역할은 무엇인가요?

1. 객체의 인스턴스 변수 값을 반환하는 역할
2. 객체를 생성할 때 사용하는 메소드
3. 객체의 메소드를 실행하는 역할
4. 객체를 초기화하는 역할

##### 문제2 : **setter** 메소드의 특성 중 옳바르지 않은 것은?

1. 반환값이 있어야 함
2. 주로 인스턴스 변수의 값을 변경하는 역할
3. 메소드 이름은 'set'으로 시작해야 함
4. 메소드는 매개변수를 가져야 함

##### 문제3 : 클래스에서 **getter**를 사용하는 이유는?

##### 문제4 : (o,x) 문제

1. getter 메소드는 반환값이 있어야 한다.
2. setter 메소드는 일반적으로 'void'의 반환값을 가진다.
3. setter 는 변수값을 할당하는 목적의 함수이기에 인자를 받아야 한다.
4. setter 메소드는 반드시 최소한 하나 이상의 매개 변수를 가져야 합니다.
5. 메소드의 반환 타입이 'void'인 경우, 해당 메소드는 값을 반환하지 않는다.

##### 문제5 : **getter** 메소드의 인스턴스 변수에 접근 제한자로 사용되는 키워드는 무엇인가요?

#### < day 6 (조하연) >

##### 문제1. (객관식) 클래스 내부에서 사용 가능한 전역변수를 뜻하는 용어를 고르세요.

1. 객체(인스턴스) 필드
2. 참조형 변수
3. 지역 변수
4. 기본형 변수

##### 문제2. (단답식) 아래 코드에서 현재 자신의 객체를 나타내는 변수는?

```
public String getField1(){  
    return this.field1;  
}
```

### 문제3. (OX) O/X 문제

1. 하나의 클래스 파일에 기본 생성자와 커스텀 생성자를 모두 작성하는 것은 오버라이딩이다. (O/X)
2. 객체를 생성해야 정의된 static 메소드를 사용할 수 있다 (O/X)
3. 생성된 객체는 메모리 상에 접근할 수 있는 참조값을 가진다 (O/X)
4. 매개변수가 없는 생성자를 사용해 외부에서 전달받은 값으로 멤버 변수(필드)에 데이터 저장이 가능하다 (O/X)
5. 클래스 내부에 변수를 private으로 선언했다면 다른 패키지 또는 다른 클래스에서도 제한없이 사용 가능하다.(O/X)
6. 자동으로 생성되는 기본 생성자의 제한은 클래스의 접근 제한과 같다.

### 문제4. (코딩) 클래스와 객체 배열을 사용해 랜덤으로 산술연산 문제를 생성하고 사용자에게 답을 입력받는 프로그램을 만들기

요구 사항

- 1) 2자리 덧셈, 뺄셈, 곱셈, 나눗셈
- 2) 피연산자는 1~10자리의 랜덤한 정수
- 3) 문제 생성에 MathProblem.java 클래스와 객체 배열 사용
- 4) 문제 출력 및 정답 확인은 MathProblemMain.java 클래스에서 실행

### < Day 7 (차정호) >

문제1.Math.max ,Math.min 을 사용해서(100,200,300,400)의 최소값과 최대값을 구하시오.

문제2.Override의 사용 조건중 옳지 않은 것을 고르시오.

1. 부모 클래스와 자식 클래스 사이에서만 성립됩니다.
2. private 메소드는 상속 자체가 되지 않아 오버라이드도 성립되지 않습니다.
3. 리턴 타입, 메소드 명, 매개변수 패턴이 모두 같지 않아도 사용 가능 합니다.
4. 부모 클래스 메소드의 접근 제한자 범위보다 작아질 수 없고 확장은 가능합니다.

문제3.접근한정자 설명중 옳은 것을 모두 고르시오

1. private : 모두에게 비공개, 클래스 내부에서만 접근 가능합니다.
2. protect : 상속받은 클래스 혹은 같은 패키지 안에서만 접근이 가능합니다.
3. default : 같은 패키지 클래스만 접근이 가능합니다.
4. public : 모두에게 공개하지만 상황에 따라서 접근제한이 될 수 있습니다.

문제4.Wrapper 클래스 5가지를 적으시오.

문제5.빈칸을 채우시오.

```
public class WasingMachine {  
    private int size;  
    public WasingMachine (int size){  
        this.size = size;
```

```
    }  
}  
public class TrumWashingMachine extends WasingMachine{  
    private int sizeOfTrum;  
    public TrumWashingMachine(int sizeOfTrum, int size){  
        [          ?          ]  
        this.sizeOfTrum=sizeOfTrum;  
    }  
}
```

#### < day 8 (한진만) >

문제1. **StringBuilder**에 대해 설명하고 **String**과의 차이점을 서술하시오.

문제2. 추상 클래스에서 **new**를 이용해서 객체를 직접적으로 생성할 수 없는 이유를 서술하시오

문제3. 익명 내부 클래스의 설명으로 옳은 것을 고르시오

1. 익명 내부 클래스는 여러번 사용할 수 있다.
2. 추상 클래스를 상속 받지 않는다.
3. 생성자를 가질 수 있다.
4. 둘 이상의 인터페이스를 구현할 수 없다.

문제4. 다음 중 인터페이스의 선언된 구성요소와 생략된 키워드가 잘못 연결된 것을 모두 고르세요.

1. int calculate(int a, int b); : public abstract
2. String TYPE="ALL"; : public final
3. default temporary(String test){ } : public
4. static print() { } : protected

문제5. (O/X)

1. "implements" 키워드는 클래스가 인터페이스를 구현하는 데 사용된다. (O/X)
2. 클래스가 인터페이스를 구현할 때, 인터페이스의 모든 메서드를 구현해야함. (O/X)
3. 한 클래스가 여러 개의 인터페이스를 동시에 구현할 수 있다. (O/X)
4. 인터페이스는 멤버 변수를 가질 수 있다. (O/X)
5. 인터페이스에서 정의된 메서드의 본문을 가져야 한다. (O/X)

#### < Day 9 (황병훈) >

문제1. 빈칸을 채우시오.

@FunctionalInterface는 자바의 [ ]이며,  
[ ]를 정의할 때 사용된다.

문제2. (O/X)

1. Comparator는 <> 제너릭타입에 기본형을 사용할 수 있다.
2. '함수'의 형식으로 표현되는 인터페이스는 추상메소드가 여러 개 필요하다.
3. 메소드 정렬 알고리즘은 비교결과 리턴 값이 양수이면 교환한다.

문제3. 밑줄 친 빈 칸에 공통으로 들어갈 단어를 고르시오. 또한 해당 결과도 쓰시오.(2개 모두)

```
String na = "hi";    String you = "bye";  
System.out.println(na._____(you));  
System.out.println(you._____(na));
```

1. CompareTo
2. compareTo
3. Compareto
4. compareToo

문제4. 다음 주어진 배열을 sort와 Comparator를 사용해 내림차순으로 출력하시오.

```
Integer[] numbers = { 4, 1, 2, 5, 3};
```

문제5. 아래 코드를 람다식으로 바꿔쓰시오. (Student.java 클래스 필요)

```
Student[] students = new Student[4];  
students[0] = new Student("김태완", 30);  
students[1] = new Student("한진만", 23);  
students[2] = new Student("이광원", 31);  
students[3] = new Student("황병훈", 29);  
  
Comparator<Student> ageDecending = new Comparator<Student>() {  
    @Override  
    public int compare(Student o1, Student o2) {  
        return o2.getAge() - o1.getAge();  
    }  
};  
Arrays.sort(students, ageDecending);  
System.out.println(Arrays.toString(students));
```

< 배열 (한주영) >

문제1. 아래 설명은 배열에 대한 것으로 참,거짓을 판별하시오.

1. 배열은 인덱스를 이용해서 자료형이 같은 데이터를 관리한다 ( O , X )
2. System.out.println(배열);로 출력되는 정보는 객체의 이름과 참조값이다 ( O , X )
3. 정수 배열의 경우 배열값이 변경되어도 메모리의 위치는 동일하다 ( O , X )

4. 문자열 String 은 배열이 변경되어도 메모리 위치가 변경 안된다 ( O , X )
5. 배열 참조값은 10진수로 되어있어 '배열이름.hashCode'를 이용하면 변환한다 ( O , X )

**문제2. 다음 중 배열을 선언하는 올바른 방법을 고르시오.**

1. int arr = new int[5];
2. int[] arr = new int(5);
3. int arr[5];
4. int[] arr = {1, 2, 3, 4, 5};

**문제3. 다음과 같은 배열이 있을 때, 배열에 저장된 데이터를 연결하여 문자열로 리턴하는 메소드를 실행하는 코드를 작성하세요.**

```
int[] intel = {4,5,6,7,8,98,};  
System.out.println(_____);
```

**문제4. 다음은 평균을 구하기 위한 코드를 작성한 것이다. 오류가 나는 부분을 고치시오.**

```
public class Avg {  
    public static void main(String[] args) {  
        int score = { 93, 75, 95, 76, 70 };  
        int sum;  
        for (int i = 0; i < sum.length; i++) {  
            sum += score[i];  
        }  
        double avg = sum / score.length;  
        System.out.println("점수 합계 : " + sum); // 결과 : 409  
        System.out.println("점수 평균 : " + avg); // 결과 : 81.8  
    }  
}
```

**문제5. 문자열을 입력 받아 문자열을 구성하는 문자를 각각 배열에 저장하여 아래와 같이 검색하는 코드를 작성하세요.**

```
[실행 예시]  
문자열 입력하세요. test  
문자를 입력하세요. : e  
:: 실행 결과 ::  
test에 e가 존재하는 위치(인덱스) : 1  
e 개수 : 1
```