Day 5 (조지수**)**

문제1 : 객체 지향 프로그래밍에서 Getter의 역할은 무엇인가요? 정답 :1번

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 올바르지 않은 것은? 정답 : 1번

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 getter를 사용하는 이유는?

-> 정답 **:** 클래스의 상태를 외부에서 안전하게 읽기 위해

문제4 : (o,x) 문제

1) getter 메소드는 반환값이 있어야 한다. **o**

2) setter 메소드는 일반적으로 'void'의 반환값을 가진다. **o**

3) set 은 변수값을 할당하는 목적의 함수이기에 인자를 받아야 한다. **o**

4) setter 메소드는 반드시 최소한 하나 이상의 매개 변수를 가져야 합니다. **o**

5) 메소드의 반환 타입이 'void'인 경우, 해당 메소드는 값을 반환하지 않는다.**o**

문제5 : getter 메소드의 인스턴스 변수에 접근 제한자로 사용되는 키워드는 무엇인가요? -> 정답 : **private**

Day 6 (조하연)
1. (객관식) 1)
2. (단답형) this
3. (OX문제) 1)X, 2)O, 3)O, 4)X, 5)X
4. (코딩)

```
import java.util.Random;
import java.util.Scanner;

class MathProblem {
    private char operator;
    private int operand1;
    private int operand2;

    public MathProblem(char operator, int operand1, int operand2) {
        this.operator = operator;
        this.operand1 = operand1;
        this.operand2 = operand2;
    }

    public String generateProblem() {
        return operand1 + " " + operator + " " + operand2 + " = ?";
    }

    public double getAnswer() {
        switch (operator) {
            case '+':
                return operand1 + operand2;
            case '-':
                return operand1 - operand2;
            case '*':
                return operand1 * operand2;
            case '/':
                return (double) operand1 / operand2;
            default:
                throw new IllegalArgumentException("올바르지 않은 연산자입니다.");
        }
    }
}

public class MathProblemMain {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        Random random = new Random();

        // 문제 생성 및 배열에 저장
        MathProblem[] problems = new MathProblem[5];
        for (int i = 0; i < problems.length; i++) {
```

```
            char operator = getRandomOperator();
            int operand1 = random.nextInt(10) + 1; // 1부터 10까지의 랜덤한 피연산자
            int operand2 = random.nextInt(10) + 1;
            problems[i] = new MathProblem(operator, operand1, operand2);
        }

        // 문제 출력 및 정답 확인
        for (int i = 0; i < problems.length; i++) {
            System.out.println("문제 " + (i + 1) + ": " + problems[i].generateProblem());
            System.out.print("정답을 입력하세요: ");
            double userAnswer = scanner.nextDouble();
            double correctAnswer = problems[i].getAnswer();

            if (userAnswer == correctAnswer) {
                System.out.println("정답입니다!\n");
            } else {
                System.out.println("틀렸습니다. 정답은 " + correctAnswer + " 입니다.\n");
            }
        }
    }

    private static char getRandomOperator() {
        char[] operators = {'+', '-', '*', '/'};
        return operators[new Random().nextInt(operators.length)];
    }
}
```

Day 7 (차정호**)**

1.(100,200,300,400)의 최소값과 최대값을 구하시오.

System.out.println("Math.max(100,200,300,400) =" + Math.max(Math.max(100, 200),Math.max(300, 400))); //오버로딩을 해서 작성합니다

System.out.println("Math.max(100,200,300,400) =" + Math.min(Math.min(100, 200),Math.min(300, 400)));

2.Override의 사용 조건중 옳지 않은 것을 고르시오. 3

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.

④부모 클래스 메소드의 접근 제한자 범위보다 작아질 수 없고 확장은 가능합니다.

3.접근한정자 설명중 옳은 것을 모두 고르시오 4

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.

4.Wrapper 클래스 5가지를 적으시오.

Integer, Long,Double,Character,Boolean

5.super

Day 8 (한진만**)**
1. StringBuilder에 대해 설명하고 String과의 차이점을 서술하시오 :

1번 해설 : StringBuilder은 문자열은 변경하고 조작하기 위한 클래스
String 클래스는 불변이기 때문에 문자열을 변경할 때 새로운 문자열을 생성하지만
StringBuilder는 가변이기 때문에 기존의 문자열에서 직접 수정 가능

2. 추상 클래스에서 new를 이용해서 객체를 직접적으로 생성할 수 없는 이유를 서술하시오

2번 해설 : 추상 클래스의 주요 목적은 다형성을 위해 만들어짐
추상 클래스는 상속을 통해 서브 클래스에 의해 확장되고, 다른 클래스들이 해당 클래스를 상속받아 자신의 특징을 더할 수 있는데
객체 생성은 그보다는 상속과 다형성을 통한 확장을 목적으로 하는 것이기 때문에 직접적인 인스턴스화를 허용하지 않음

3. 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.

3번 해설 : 익명 내부 클래스는 오로지 단 하나의 클래스를 상속받거나 단 하나의 인터페이스만을 구현 할 수 있기 때문에 정답 4

4. InterfaceA를 선언하려고 한다. 아래 보기를 통하여 InterfaceA 선언을 코딩하라.
보기 : 한개의 2개의 int 타입, 1개의 String 타입을 반드시 포함하여야 하며
추상 메소드와 default 키워드를 필히 사용해야 한다.
(메소드 명과 파라미터 명은 마음대로)

정답 :
```
// InterfaceA 선언
public interface InterfaceA {
    // 추상 메서드 선언
    void method1(int a); // int 타입 파라미터를 받는 추상 메서드
    void method2(int a, int b, String str); // int 타입 두 개와 String 타입 파라미터를 받는 추상 메서드

    // default 메서드 선언
    default void defaultMethod() {
        System.out.println("이것은 default 메서드입니다."); // 기본 구현을 가진 default 메서드
    }
}
```

5.
"implements" 키워드는 클래스가 인터페이스를 구현하는 데 사용된다. (O/X)
답: O
설명: "implements" 키워드는 클래스가 특정 인터페이스를 구현하기 위해 사용됨.

클래스가 인터페이스를 구현할 때, 인터페이스의 모든 메서드를 구현해야함. (O/X)
답: O
설명: 클래스가 인터페이스를 구현할 때는 인터페이스의 모든 추상 메서드를 반드시 구현해야함.

한 클래스가 여러 개의 인터페이스를 동시에 구현할 수 있다. (O/X)
답: O
설명: 자바에서는 한 클래스가 여러 개의 인터페이스를 동시에 구현할 수 있음. 이를 다중 인터페이스 구현이라고 함.

인터페이스는 멤버 변수를 가질 수 있다. (O/X)
답: X
설명: 인터페이스는 메서드 선언만을 갖고 있으며, 멤버 변수를 갖지 않음.

인터페이스에서 정의된 메서드의 본문을 가져야 한다. (O/X)
답: X
설명: 인터페이스는 추상 메서드의 집합으로, 메서드의 선언만 있고 본문은 없음. 구현은 해당 인터페이스를 구현하는 클래스에서 이루어짐.

Day 9 (황병훈**)**
1.어노테이션**,** 함수형 인터페이스

2. 1) X 2) X 3) O

3. 2번 **, 6, -6**

4. 정답 **:**

```
Arrays.sort(numbers, new Comparator<Integer>() {

  @Override
   public int compare(Integer o1, Integer o2) {
      return o2 - o1;

   }
});
Arrays.sort(numbers2,(o1,o2) -> {return o2 - o1;});
System.out.println(Arrays.toString(numbers2));
```

5. 정답 **:**

```
Arrays.sort(students,new Comparator<Student>() {

   @Override
   public int compare(Student o1, Student o2) {

      return o2.getAge() - o1.getAge();      // 내림차순 o1 > o2이 되어야합니다.
                                             // o2 - o1 < 0 이면 교환.
   }

});
System.out.println(Arrays.toString(students));
```

배열**(**한주영**)**

문제 1번 -한주영

아래 설명은 배열에 대한 것으로 참,거짓을 판별하시오**.**

(1) 배열은 인덱스를 이용해서 자료형이 같은 데이터를 관리한다 (O)

(2) System.out.println(배열); 을 할 경우 참조값을 출력한다 (O)

(3) 정수배열의 경우 배열값이 변경되어도 메모리의 위치는 동일합니다 (O)

(4) 문자열 String 은 배열이 변경되어도 메모리 위치가 변경 안됩니다 (X)

(5) 배열 참조값은 10진수로 되어있어 '배열이름.hashCode'를 이용하면 변한다 (X)

문제 2번 -한주영

다음 중 배열을 선언하는 올바른 방법을 고르시오**. (2)**

(1) int arr = new int[5];

(2) int[] arr = new int(5);

(3) int arr[5];

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

문제 3번 -한주영

다음과 같은 배열이 있을 때, 배열 인자를 보기 위해 어떻게 출력해야하는지 빈칸을 채우시오.

```
int[] intel ={4,5,6,7,8,98,};

System.out.println(Arrays.toString(intel));
```

문제 4번 -한주영

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 = 0;

		for (int i = 0; i < score.length; i++) {

			sum += score[i];

		}

		double avg = (double) sum / score.length;		System.out.println("점수 합계 : " + sum); // 결과 : 409

		System.out.println("점수 평균 : " + avg); // 결과 : 81.8

	}

}
```

문제 5번 -한주영

문자열을 입력 받아 하나씩 배열에 넣고 검색할 문자가 문자열에 몇 개 들어가 있고, 몇 번째 인덱스에 있는지 출력하는 코드를 작성하시오.

출력문 예시)

문자열 : test

문자 : e

test에 e가 존재하는 위치(인덱스) : 1

e 개수 : 1

```
public void Test() {

	Scanner sc = new Scanner(System.in);


	System.out.print("문자열 : ");

	String str = sc.nextLine();
```

```
        System.out.print("문자 : ");
        char ch = sc.nextLine().charAt(0);

    char[] arr = new char[str.length()];

    for(int i = 0; i < arr.length; i++) {
        arr[i] = str.charAt(i);
    }

        int count = 0;
        System.out.print("test에 c가 존재하는 위치(인덱스) : ");
        for(int i = 0; i < arr.length; i++) {
        if(arr[i] == ch) {
        System.out.print(e +" ");
        count++;
        }
        }
    System.out.println();
    System.out.println("e 개수 : " + count);
        sc.close();
}
```