

# 객체지향 프로그래밍

## Chapter 10 ~ 11

클래스 변수와 클래스 메소드, 메소드 오버로딩과 스트링 클래스

5주차 과제

# 문제1

---

String text에 "I love you" 값으로 초기화하고 다음 결과와 같이 출력되도록 프로그램을 작성하시오.

```
public class Ex1 {  
    public static void main(String[] args) {  
        String text = "I love you";  
        ...  
    }  
}
```

\* Console 결과

```
love youl  
love youl  
ove youl I  
ve youl lo  
e youl lov  
youl love  
youl love  
oul love y  
ul love yo  
I love you
```

## 문제2

1달러에 대한 원화환율을 1121원으로 정하고, 다음 main()과 같이 실행되어 같은 결과가 나오도록 CurrencyConverter 클래스를 작성하시오.

```
public class Ex2 {  
    public static void main(String[] args) {  
        CurrencyConverter.setRate(1121);  
        System.out.println("환율1달러 : " + CurrencyConverter.getRate());  
        System.out.println("백만원은 $" +  
            CurrencyConverter.toDollar(1000000) + "입니다.");  
        System.out.println("$120는 " +  
            CurrencyConverter.toKW(120) + "원 입니다.");  
    }  
}
```

\* Console 결과

```
환율1달러 : 1121.0  
백만원은 $892.0606601248885입니다.  
$120는 134520.0원 입니다.
```

## 문제3

---

주어진 text1~text4에 대해 결과와 같은 문장이 되도록 프로그램을 작성하시오.  
단, `StringBuilder`, `String` 클래스의 멤버 메소드를 활용하여 완성하라.

```
public class Ex3 {  
    public static void main(String[] args) {  
        String text1 = "eW";  
        String text2 = "LOVE";  
        String text3 = "C++";  
        String text4 = "programming.";  
        ...  
    }  
}
```

\* Console 결과

We love JAVA programming.

---

## 문제4-1

---

Circle 클래스를 통해 인스턴스 3개를 만들어 그 원의 중심점과 반지름을 출력한다. 그리고 a원에 대해 b와 c의 원과 비교하는데 그 원의 중심이 같으면 같은 원이고 중심이 다르면 다른 원이다.

Circle 클래스를 인스턴스 할 때 주어진 Ex4 클래스와 같이 각각 다른 3개의 생성자를 구현해야 한다. 생성자에 대해 중심점의 값이 없으면 (0, 0)으로 초기화하고, 반지름이 없으면 반지름을 5로 초기화한다. 이 때 키워드 this를 사용한다.

## 문제4-2

```
public class ex4 {  
    public static void main(String[] args) {  
        Circle a = new Circle(2, 3, 30);  
        Circle b = new Circle(5);  
        Circle c = new Circle(2, 3);  
  
        System.out.println("원 a : " + a);  
        System.out.println("원 b : " + b);  
        System.out.println("원 c : " + c);  
  
        if (a.equals(b))  
            System.out.println("같은 원");  
        else  
            System.out.println("서로 다른 원");  
        if (a.equals(c))  
            System.out.println("같은 원");  
        else  
            System.out.println("서로 다른 원");  
    }  
}
```

\* Console 결과

```
원 a : Circle(2,3) 반지름30  
원 b : Circle(0,0) 반지름5  
원 c : Circle(2,3) 반지름5  
서로 다른 원  
같은 원
```

## 문제5-1

---

클래스 ExId를 설명과 같이 작성하시오. 인스턴스를 생성할 때마다 식별번호를  $n$ 씩 증가시켜서 부여한다. ( $n$ 은 양의 정수)  $n$ 값은 지정하지 않으면 1로 설정하고 메서드를 통해서 확인 및 변경할 수 있다. 예를 들어 인스턴스를 2개 생성한 후에  $n$ 을 3으로 변경한 경우, 다시 인스턴스를 2개 생성한다. 이때 인스턴스에 부여하는 식별 번호는 생성된 순으로 1, 2, 5, 8이 된다.

\* Console 결과

```
a의 식별 번호 : 1  
b의 식별 번호 : 2  
c의 식별 번호 : 5  
d의 식별 번호 : 8  
마지막 부여한 식별 번호 : 8  
다음에 부여할 식별 번호 : 9
```

## 문제5-2

---

```
public class ex5 {  
    public static void main(String args[]) {  
        ExId a = new ExId();  
        ExId b = new ExId();  
        ExId.setStep(3);  
        ExId c = new ExId();  
        ExId d = new ExId();  
  
        System.out.println("a의 식별 번호 : " + a.getId());  
        System.out.println("b의 식별 번호 : " + b.getId());  
        System.out.println("c의 식별 번호 : " + c.getId());  
        System.out.println("d의 식별 번호 : " + d.getId());  
        ExId.setStep(1);  
        int max = ExId.getMaxId();  
        System.out.println("마지막 부여한 식별 번호 : " + max);  
        System.out.println("다음에 부여할 식별 번호 : " +  
            (max+ExId.getStep()));  
    }  
}
```