



# JAVA HomeWork

## Class 문제

---

경북대학교  
소프트웨어융합과  
배희호 교수



# Report 1



- 다음과 같이 소수(Prime Number)를 판별하는 Program을 객체 지향으로 작성하여라.

자연수 입력 : 2

2는 소수(Prime Number) 입니다.

자연수 입력 : 4

4는 합성수(Composite number) 입니다.



# Report 1



## ■ 소수(素數, Prime Number)

- 1보다 큰 자연수 중 1과 자기 자신만을 약수로 가지는 수
- 예) 5는  $1 \times 5$  또는  $5 \times 1$ 로 수를 곱한 결과를 적는 유일한 방법이 그 수 자신을 포함하기 때문에 5는 소수임
- 예) 6은 자신보다 작은 두 숫자의 곱( $2 \times 3$ )이므로 소수가 아닌데, 이렇듯 1보다 큰 자연수 중 소수가 아닌 것은 합성수(Composite Number)라고 함



# Report 1



## ■ Prime 클래스

```
public class Prime {  
    private int number;  
  
    public Prime(int number) {  
        this.number = number;  
    }  
  
    private boolean check() {  
        int count = 0;  
  
        for(int i = 2; i <= this.number; ++i) {  
            if (this.number % i == 0) {  
                ++count;  
            }  
        }  
        if (count == 1) {  
            return true;  
        } else {  
            return false;  
        }  
    }  
}
```



# Report 1



## ■ Prime 클래스

```
public String toString() {  
    return this.number + (this.check() ? "는 소수(Prime Number) 입니다." :  
                                "는 합성수(Composite number) 입니다.");  
}
```



# Report 1



## ■ Main 클래스

```
public class Main {  
  
    public static void main(String[] args) {  
        Scanner keyboard = new Scanner(System.in);  
        System.out.print("자연수 입력 : ");  
        int number = keyboard.nextInt();  
        Prime prime = new Prime(number);  
        prime.check();  
        System.out.println(prime);  
    }  
}
```



# Report 2



- 다음과 같이 자연수를 받아 약수 구하는 프로그램을 작성해 보자

자연수 입력 : 8

8의 약수는 1, 2, 4, 8 입니다

약수의 합은 15 입니다

자연수 입력 : 19

19의 약수는 1, 19 입니다

약수의 합은 20 입니다



# Report 2



## ■ Divisor 클래스

```
public class Divisor {  
    private int number;  
  
    public Divisor(int number) {  
        this.number = number;  
    }  
  
    private String divisor() {  
        String result = "";  
        for(int i = 1; i <= number; i++) {  
            if ((number % i) == 0) {  
                if (i == 1)  
                    result += "" + i;  
                else  
                    result += ", " + i;  
            }  
        }  
        return result;  
    }  
}
```





# Report 2



## ■ Divisor 클래스

```
private int sum() {  
    String[] numberList = divisor().split(",");  
    int total = 0;  
    for (int i = 0; i < numberList.length; i++) {  
        total += Integer.parseInt(numberList[i].trim());  
    }  
    return total;  
}
```

@Override

```
public String toString() {  
    return number + "의 약수는 " + divisor() + " 입니다" +  
        "\n약수의 합은 " + sum() + " 입니다";  
}
```



# Report 2



## ■ Main 클래스

```
public class Main {  
  
    public static void main(String[] args) {  
        Scanner keyboard = new Scanner(System.in);  
        System.out.print("자연수 입력 : ");  
        int number = keyboard.nextInt();  
        Divisor divisor = new Divisor(number);  
        System.out.println(divisor);  
    }  
}
```



# Report 3

- 다음과 같이 자연수 2개를 받아 최대공약수 구하는 프로그램을 작성해보자

자연수 2개 입력 : 72 30

72, 30의 GCD(최대공약수)는 6입니다



# Report 3



- 최대공약수(GCD : Greatest Common Divisor)
  - 최대공약수는 두 자연수의 공통된 약수 중 가장 큰 수를 의미
  - 예) 72와 30의 최대공약수는 6
- 최소공배수(LCM : Least Common Multiple)
  - 최소공배수는 두 자연수의 공통된 배수 중 가장 작은 수를 의미
  - 예) 72와 30의 최소공배수는 360
  - 최소공배수 = 두 자연수의 곱 / 최대공약수

$$lcm(a, b) = \frac{a \times b}{GCD(a, b)}$$



# Report 3



## ■ GCD 클래스

```
public class GCD {  
    private int number1;  
    private int number2;  
  
    public GCD(int number1, int number2) {  
        this.number1 = number1;  
        this.number2 = number2;  
    }  
  
    private int gcd(int number1, int number2){  
        if(number2 == 0)  
            return number1;  
        return gcd(number2, number1 % number2);  
    }  
  
    @Override  
    public String toString() {  
        return number1 + ", " + number2 +  
            "의 GCD(최대공약수)는 " + gcd(number1, number2) + "입니다";  
    }  
}
```



# Report 3



## ■ Main 클래스

```
public class Main {  
  
    public static void main(String[] args) {  
        Scanner keyboard = new Scanner(System.in);  
        System.out.print("자연수 2개 입력 : ");  
        int number1 = keyboard.nextInt();  
        int number2 = keyboard.nextInt();  
        GCD gcd = new GCD(number1, number2);  
        System.out.println(gcd);  
    }  
}
```



# Report 4



- 다음과 같이 자연수 2개를 받아 최소공배수 구하는 프로그램을 작성해보자

자연수 2개 입력 : 72 30

72, 30의 GCD(최대공약수)는 6입니다

72, 30의 LCM(최소공배수)는 360

- Hint

- 최대공약수는 앞의 Report 3을 그대로 사용할 것



# Report 4



## ■ GCD 클래스

```
public class GCD {  
    private int number1;  
    private int number2;  
  
    public GCD(int number1, int number2) {  
        this.number1 = number1;  
        this.number2 = number2;  
    }  
  
    private int gcd(int number1, int number2){  
        if(number2 == 0)  
            return number1;  
        return gcd(number2, number1 % number2);  
    }  
  
    @Override  
    public String toString() {  
        return number1 + ", " + number2 +  
            "의 GCD(최대공약수)는 " + gcd(number1, number2) + "입니다";  
    }  
}
```





# Report 4



## ■ LCM 클래스

```
public class LCM {  
    private int number1;  
    private int number2;  
    private GCD gcd;  
  
    public LCM(int number1, int number2) {  
        this.number1 = number1;  
        this.number2 = number2;  
        gcd = new GCD(number1, number2);  
    }  
  
    private int lcm() {  
        return (number1 * number2) / gcd.gcd(number1, number2);  
    }  
    @Override  
    public String toString() {  
        return gcd.toString() + "\n" +  
            number1 + ", " + number2 +  
            "의 LCM(최소공배수)는 " + lcm();  
    }  
}
```



# Report 4



## ■ Main 클래스

```
public class Main {  
  
    public static void main(String[] args) {  
        Scanner keyboard = new Scanner(System.in);  
        System.out.print("자연수 2개 입력 : ");  
        int number1 = keyboard.nextInt();  
        int number2 = keyboard.nextInt();  
        LCM lcm = new LCM(number1, number2);  
        System.out.println(lcm);  
    }  
}
```



# Report 제출 방법



- 보고서는 기본적으로 PPT 파일에 작성한다
  - 문제
  - 문제 해결에 필요한 이론적인 내용
  - 소스 파일을 텍스트 형태로 PPT 파일에 복사하여 완성할 것
  - 실행 결과는 실행 결과 화면을 캡처하여 PPT 파일에 넣어서 작성할 것
  - 숙제를 한 이후의 느낀 점, 하고싶은 말, 또는 불평~~
- 보고서와 소스/바이트 코드를 하나로 묶은 ZIP 파일을 e-강의실에 업로드 할 것
  - 소스 코드, 바이트 코드 반드시 포함
  - 파일명: X차-홍길동-1401234.zip