

Understanding Java Language

Innova Lee(이상훈)
gcccompil3r@gmail.com

Table of Contents

1. Java란 무엇인가 ?
2. JVM의 내부
3. Java의 기본 골격
4. 자료형
5. 제어문과 반복문
6. Java의 Class
7. 생성자와 배열
8. 상속과 interface
9. Inner Class와 Package
10. Java의 Collection
11. Java의 Exception
12. Logger
13. Thread
14. GUI Programming
15. Event
16. Swing
17. I/O Stream
18. Networking
19. Java Native Interface(JNI)
20. JDBC Programming

What is the Java ?

Java란 무엇인가 ?

C, C++보다 좋은 언어 ?!

쉬운 언어 ?

JVM을 통해 HW와 SW를 분리함

그러나 실제로 밑에서 C, C++이 동작

속도가 느리지만 안정성이 뛰어남

개발 속도가 타 언어에 비해 빨라짐

History of Java

Java 1.2에서 생긴 범주

SE – 일반용

ME – 임베디드용

EE – 기업용

Java Platform

Java Platform의 구성요소

JVM – 가상 머신

Java API – Library의 모음(Java의 핵심)

How to Install Java ?

1. <http://www.oracle.com/technetwork/java/index.html>로 이동
2. Downloads -> Java for Developers
3. Java SE
4. License 동의 후 다운로드
5. 설치를 수행함(위치는 C:\Java에 두자)

Environment Setting

1. 내 컴퓨터 -> 속성 -> 고급 -> 환경변수
2. 시스템 변수 PATH에 “;C:\Java\bin” 입력
3. 시스템 변수 CLASSPATH에 “%CLASSPATH%;.”을 입력

What is the Language ?

언어가 왜 필요 할까 ?

당장 내 옆 사람과 소통하기 위해
먹고 살기 위해
최소한의 의사를 표현하기 위해

What is the C Language ?

그렇다면 C 언어는 뭘까 ?

인간과 기계가 소통하기 위한 언어
C언어 외에도 이런것들은 많다
그러나 C언어는 HW와 매우 밀접함

What is the Java Language ?

그렇다면 Java는 뭘까 ?

역시 인간과 기계가 소통하기 위한 언어
C언어의 HW적인 요소를 숨김
실제로 눈에 보이지 않게 C가 동작함
가상 머신을 통해 진행되므로 속도가 떨어짐
그러나 어떤 프로그램의 개발 시간이 단축됨

Why does it needs ?

이딴게 왜 필요한데 ?

인간은 인간말 밖에 못한다.
기계도 기계어 밖에 못한다.



```
0000000000400599 <main>:
400599:      55
40059a:      48 89 e5
40059d:      48 83 ec 10
4005a1:      48 c7 45 f8 00 00 00
4005a8:      00
4005a9:      bf 10 00 00 00
4005ae:      e8 85 fe ff ff
4005b3:      48 89 45 f8
```

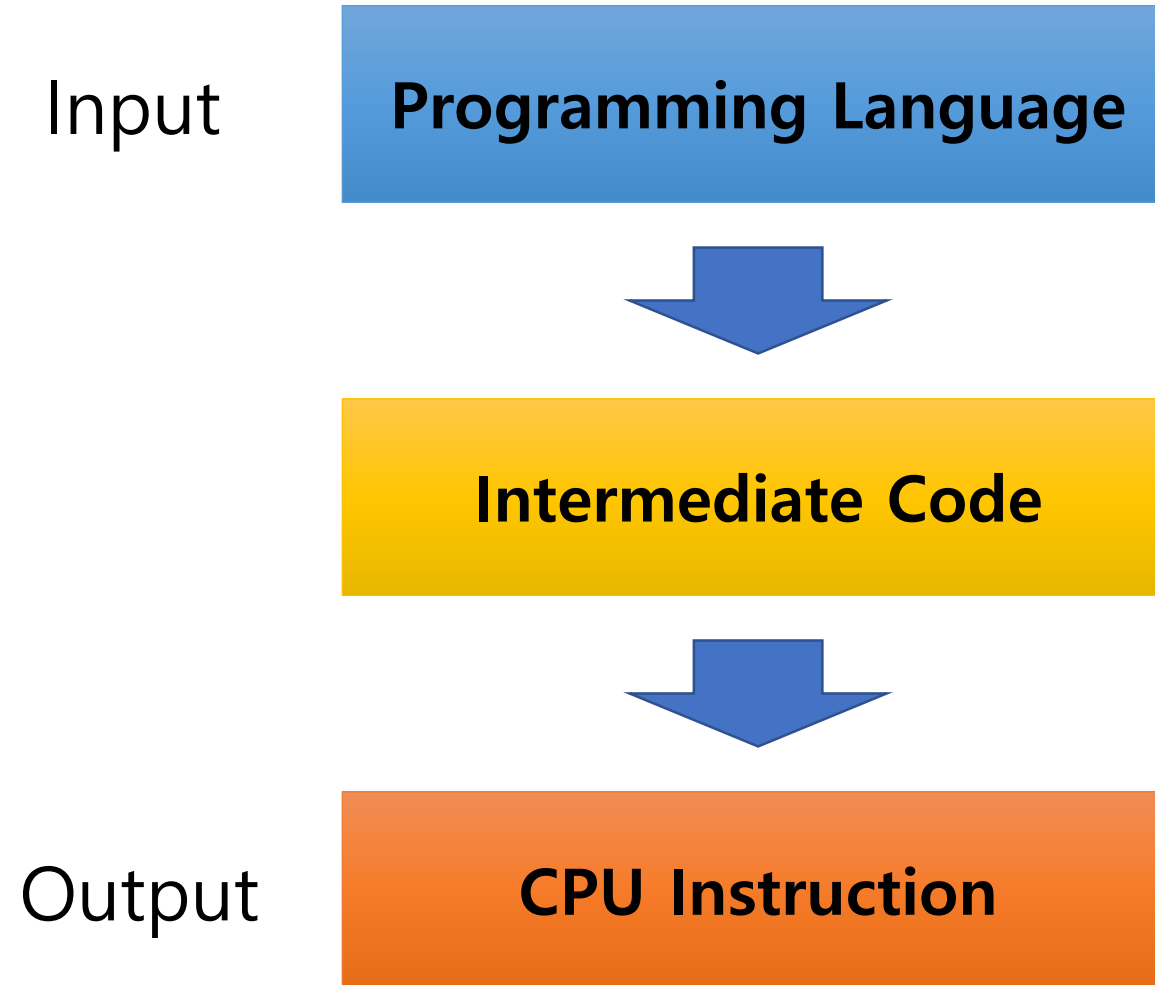
Then, What we need ?

그럼 이제 무엇이 필요할까 ?

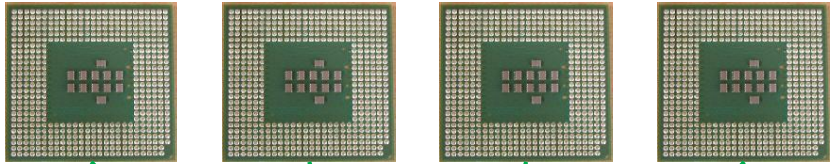
통역사가 필요하다

그것이 Compiler(컴파일러)다

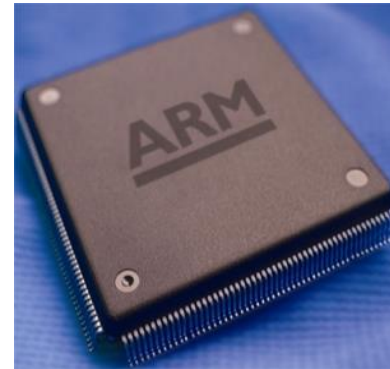
What is the Compiler ?



Target Machine



x86



ARM

각 CPU에 해당하는 기계어를
CPU Instruction이라고 한다.

Anything Else ?

더 필요한 것은 없는가 ?

Assembler와 Linker가 필요하다
이들은 부가적인 역할을 수행한다
현재 우리에게 주요 대상이 되진 않는다.

Example

```
push    %rbp
mov     %rsp,%rbp
sub     $0x10,%rsp
movq    $0x0,-0x8(%rbp)

mov     $0x10,%edi
callq   400438 <malloc@plt>
mov     %rax,-0x8(%rbp)
lea     -0x8(%rbp),%rax
mov     %rax,%rdi
callq   400568 <test>
```

x86

```
push    {fp, lr}
add     fp, sp, #4      ; 0x4
sub     sp, sp, #8      ; 0x8
mov     r3, #0          ; 0x0
str     r3, [fp, #-8]
mov     r0, #8          ; 0x8
bl      82f4 <_init+0x50>
mov     r3, r0
str     r3, [fp, #-8]
sub     r3, fp, #8      ; 0x8
mov     r0, r3
bl      83e0 <test>
```

ARM

What is the Assembler ?

Assembler는 무엇을 하는가 ?

Compiler가 만들어낸 Assembly를 해석
최종적으로 재배치 가능한 Object를 생성
Linux에서는 ELF Format을 씀

What is the Linker ?

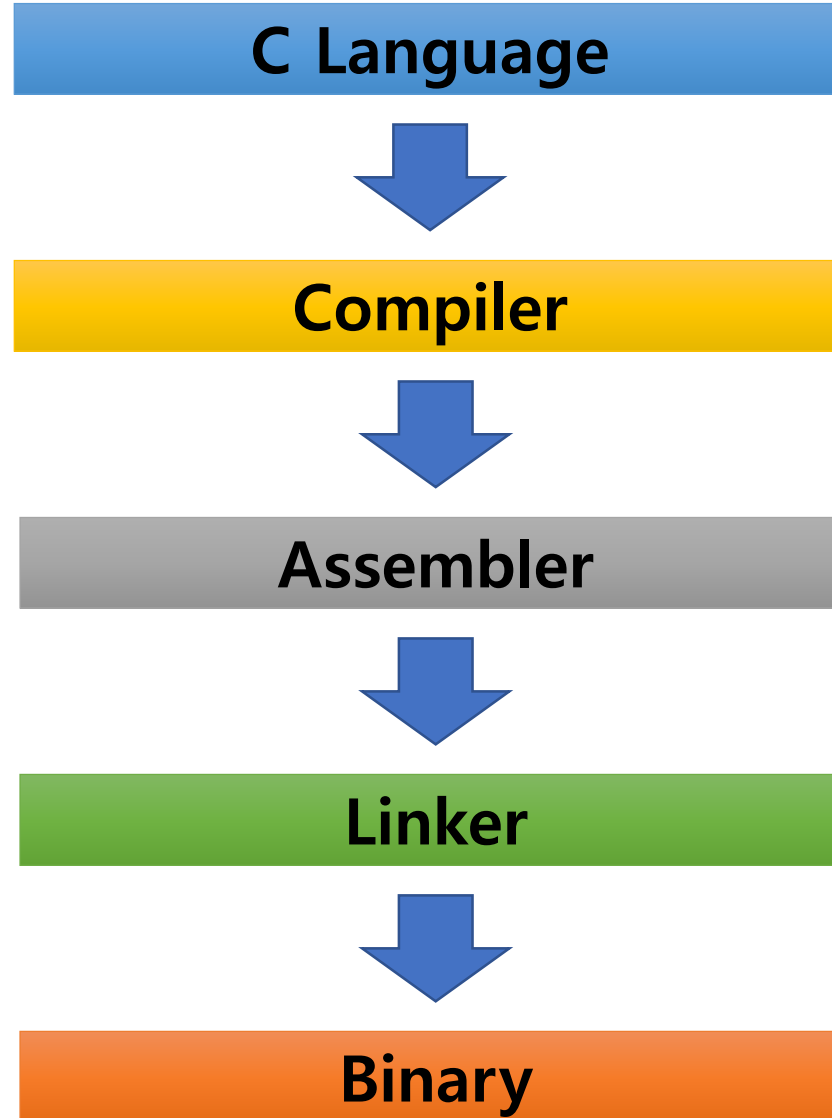
Linker는 무엇을 하는가 ?

Assembler가 만든 조각들을 합침

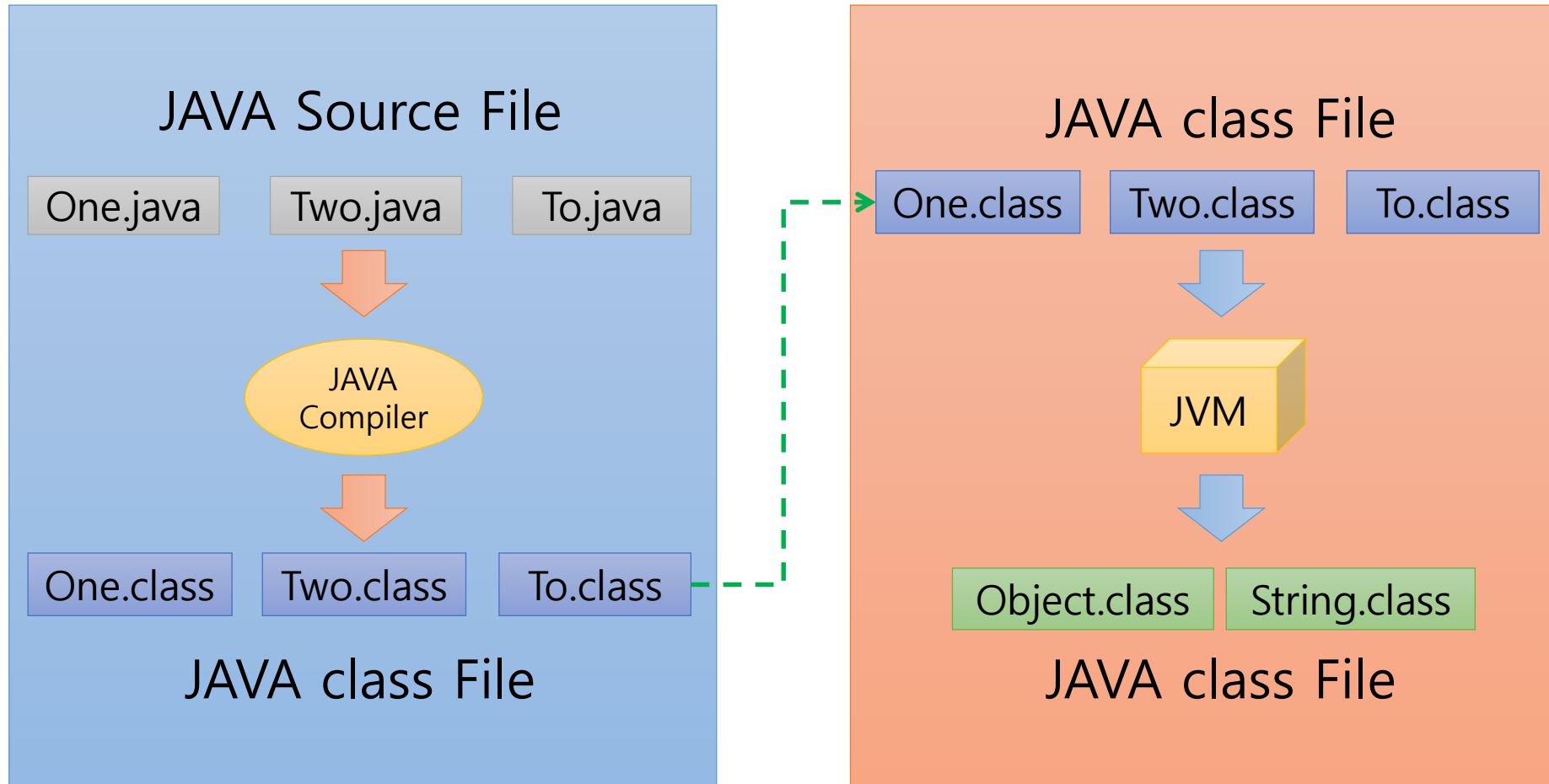
이때 Library가 함께 합쳐짐

Library란 printf, strlen등의 함수를 의미
최종적으로 실행 파일인 Binary를 생성

Full Compilation Flow



How to Java Operate ?



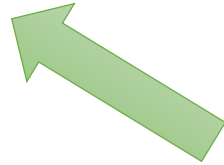
Basic of Java Programming

Java Programming에 필요한건 ?

C와 마찬가지로 main()함수가 필요하다

Example

```
Public class Hello{  
    public static void main(String args[]){  
        System.out.println("Hello Java\n");  
    }  
}
```



이것이 기본 골격이다

1. Window + R키를 누르고 cmd 입력
2. 이후에 소스코드를 만든 위치로 이동
3. javac 클래스 이름.java
4. java 클래스 이름

How to Install Eclipse ?

1. <http://www.eclipse.org>로 이동
2. Workspace는 어디에 저장할지 지정하는 것임

Definition of Class

Java에서 Class란 무엇인가 ?

Class는 객체를 만드는 설계도

객체란 컴퓨터 메모리 상에 load된 것

Java에는 반드시 1개 이상의 Class가 필요

Class로 설정할 영역을 {}으로

Definition of Method

Java에서 Method란 무엇인가 ?

Program이 해야할 일을 정의해 놓은 것
Input이 들어가면, Output이 나옴
Input은 여러 개, Output은 유일 혹은 없다
C와 마찬가지로 ()안에 매개변수를 받는다.

Definition of Statement

Java에서 Statement란 무엇인가 ?

작업을 지시하는 단위로 마지막에 ;을 사용

System은 Java에서 제공하는 class

Java의 System과 관련된 Method를 가짐

Example

```
import java.util.Scanner;

public class Add{
    public static void main(String args[]){
        Scanner input = new Scanner(System.in);

        int x, y, sum;
        System.out.println("1번째 수를 입력하세요 : ");
        x = input.nextInt();

        System.out.println("2번째 수를 입력하세요 : ");
        y = input.nextInt();

        sum = x + y;
        System.out.println(sum);
    }
}
```

Example

```
import java.util.Scanner;

public class Salary{
    public static void main(String args[]){
        int salary;
        int deposit;
        Scanner input = new Scanner(System.in);

        System.out.print("월급을 입력하시오 : ");
        salary = input.nextInt();

        deposit = 10 * 12 * salary;
        System.out.printf("10년 동안의 저축액 : %d\n", deposit);
    }
}
```

Data Type of Java

1. 변수의 Memory 공간과 Data를 참조하기 위해 결정함
2. 정수, 실수, 문자, 논리형, Class, Array, Instance가 자료형
3. C와 동일한 자료형들을 가지고 있음
4. C의 char형은 Java의 byte형과 같으며,
boolean(논리)형태가 따로 존재함
5. C와 동일하게 e를 사용하여 10의 승수를 표현 가능!

Example

```
public class FloatLiteral{  
    public static void main(String args[]){  
        double r1 = 123.5;  
        double r2 = 123.5f;  
        double r3 = 1.235e2;  
        double r4 = .235e-2;  
  
        System.out.println(r1);  
        System.out.println(r2);  
        System.out.println(r3);  
        System.out.println(r4);  
    }  
}
```

Example

```
public class IntegerLiteral{  
    public static void main(String args[]){  
        int i = 26;  
        int oi = 032;  
        int xi = 0x1a;  
  
        System.out.println(i);  
        System.out.println(oi);  
        System.out.println(xi);  
    }  
}
```


Symbolic Constant

1. C에서는 `const`였지만 Java에서는 `final`임
2. 자료형의 범위를 초과할 경우 `Overflow`
3. 아예 유효하지 않은 연산을 할 시 `Not a Number(NaN)`
4. 양의 무한대, 음의 무한대, `NaN`등이 발생할 수 있음

Example

```
public class FloatPoint{  
    public static void main(String args[]){  
        double r1 = 123.5;  
        double r2 = 1.2E-300;  
        double r3 = 1.2E+300;  
  
        System.out.println(0.0/0.0);  
        System.out.println(r1/0.0);  
        System.out.println(r2/r3);  
        System.out.println(r3/r2);  
    }  
}
```

Character Data Type

Hardware는 문자 자체를 인식하지 못하여 숫자로 처리함!

그러나 Java의 경우 Unicode를 지원하므로

```
int a = '가';  
int 인덱스 = 0; 등을 지원함
```

문제는 그다지 보기 좋은 형태는 아니므로 지양하도록 한다!

Example

```
public class Unicode{  
    pulbic static void main(String[] args){  
        int 인덱스 = 0;  
        인덱스++;  
        System.out.println(인덱스);  
    }  
}
```

Basic Operator

기본 연산자는 C와 동일하되 Pointer 계열의 연산자가 없음

+, -, *, /, %, +=, -=, *=, /=, %=, !=, ==, >, <, >=, <=, >>, <<, >>>, <<<등이 존재함

>>, <<은 부호를 따지는 Shift이며,
>>>, <<<의 경우 부호를 따지지 않는 Shift임

Control Statement

C의 제어문과 동일함

1. 제어문을 사용함으로써
순차적 구조에 유연성을 제공할 수 있다.
2. 제어문에는 조건문과 반복문이 존재함
3. 조건문 내에서 실행하고자 하는 것을 Block으로 감쌘

Example

```
import java.util.Scanner;

public class Pay{
    public static void main(String args[]){
        final int RATE = 5000;
        int pay, hours;
        Scanner input = new Scanner(System.in);

        System.out.print("시간을 입력하시오 : ");
        hours = input.nextInt();

        if(hours > 8)
            pay = RATE + 8 + (int)(1.5 * RATE * (hours - 8));
        else
            pay = RATE * 8;

        System.out.println("pay" + " = " + pay);
    }
}
```

Example

```
import java.util.Scanner;

public class Compare{
    public static void main(String args[]){
        final int def = 77;
        int result;
        Scanner input = new Scanner(System.in);

        System.out.print("숫자를 입력하시오 : ");
        result = input.nextInt();

        if(result >= def)
            System.out.println("Large");
        else
            System.out.println("Small");
    }
}
```


Example

```
import java.util.Scanner;

public class TriComp{
    public static void main(String args[]){
        int size;
        final int small = 100, medium = 200;

        Scanner input = new Scanner(System.in);
        System.out.print("숫자를 입력하시오 : ");

        size = input.nextInt();
        if(size < small)
            System.out.println("Small");
        else if(size < medium)
            System.out.println("Medium");
        else
            System.out.println("Large");
    }
}
```

Example

```
import java.util.Scanner;

public class Test{
    public static void main(String args[]){
        int result;
        Scanner input = new Scanner(System.in);

        System.out.print("숫자를 입력하시오 : ");
        result = input.nextInt();

        if(result != 0)
            System.out.println("A");
        else if(result > 3)
            System.out.println("B");
        else
            System.out.println("C");
    }
}
```

Example

```
import java.util.Scanner;

public class DayInMonth{
    public static void main(String args[]){
        int month, year = 2013, days = 0, check = 0;

        System.out.print("일수를 알고 싶은 달을 입력하시오 : ");
        Scanner scan = new Scanner(System.in)

        month = Scan.nextInt();

        switch(month){
            case 1 :
            case 3 :
            case 5 :
            case 7 :
            case 8 :
            case 10 :
            case 12 :
                days = 31;
                break;
            case 4 :
            case 6 :
            case 9 :
            case 11 :
                days = 30;
                break;
            case 2 :
                if(((year % 4 == 0) && !(year % 100 == 0)) || (year % 400 == 0))
                    days = 29;
                else
                    days = 28;
                break;
            default :
                System.out.println("월이 잘못 입력되었습니다");
                check++;
                break;
        }
        if(check == 0)
            System.out.println("월의 날수는 " + days);
    }
}
```

Example

```
import java.util.Scanner;

public class CompSwitch{
    public static void main(String args[]){
        double result;
        int check;
        Scanner scan = new Scanner(System.in)

        System.out.print("점수를 입력하시오 : ");
        result = scan.nextDouble();
        check = (int) (result / 10);

        switch(check){
            case 10 :
                System.out.println("A+");
                break;
            case 9 :
                System.out.println("A");
                break;
            case 8 :
                System.out.println("B");
                break;
            case 7 :
                System.out.println("C");
                break;
            case 6 :
                System.out.println("D");
                break;
            default :
                System.out.println("F");
        }
    }
}
```

Example

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

```
public class Gcd{  
    public static void main(String args[]){  
        int x, y, ret;  
        Scanner input = new Scanner(System.in);  
  
        System.out.print("x값(y보다 큼) 입력 : ");  
        x = input.nextInt();  
  
        System.out.print("y값(x보다 작음) 입력 : ");  
        y = input.nextInt();  
  
        if(y != 0){  
            ret = x % y;  
            x = y;  
            y = ret;  
        }  
        System.out.println("최대 공약수는 " + x);  
    }  
}
```

Example

```
import java.util.Scanner;

public class DoWhile{
    public static void main(String args[]){
        double result;
        int check;
        Scanner scan = new Scanner(System.in)

        do{
            System.out.print("점수를 입력하시오 : ");
            result = scan.nextDouble();
        } while((100 - result) < 0);
        result /= 10;

        switch((int)result){
            case 10 :
                System.out.println("A+");
                break;
            case 9 :
                System.out.println("A");
                break;
            case 8 :
                System.out.println("B");
                break;
            case 7 :
                System.out.println("C");
                break;
            case 6 :
                System.out.println("D");
                break;
            default :
                System.out.println("F");
        }
    }
}
```

Example

```
import java.util.Scanner;

public class Factorial{
    public static void main(String args[]){
        long fact = 1;
        int n;

        System.out.print("정수를 입력하시오 : ");
        Scanner input = new Scanner(System.in);
        n = input.nextInt();

        for(int i = 1; i <= n; i++){
            fact = fact * i;
        }

        System.out.println(n + "Factorial은 " + fact + "입니다");
    }
}
```