

JAVA Setting File Download & Install

Download

- JDK8

zulu 8.33.0.1 : (https://cdn.azul.com/zulu/bin/zulu8.33.0.1-jdk8.0.192-win_x64.msi)

- Eclipse

eclipse-jee-2018-09-win32-x86_64.zip : (https://www.eclipse.org/downloads/download.php?file=/technology/epp/downloads/release/2018-09/R/eclipse-jee-2018-09-win32-x86_64.zip)

- STS

Spring Tool Suite 3.9.14 (Eclipse 4.15) : (https://download.springsource.com/release/STS/3.9.14.RELEASE/dist/e4.15/spring-tool-suite-3.9.14.RELEASE-e4.15.0-win32-x86_64.zip)

- Apache Tomcat

Tomcat 9.0.64 : (<https://downloads.apache.org/tomcat/tomcat-9/v9.0.64/bin/apache-tomcat-9.0.64-windows-x64.zip>)

Install & setting

- zulu 8.33.0.1 install

- 설정 : 내PC -> 속성 -> 고급시스템설정 -> 고급(탭) -> 환경변수 -> 시스템변수

변수 이름	변수 값
JAVA_HOME	C:\Program Files\Zulu\zulu-8
CLASSPATH	.
Path	%JAVA_HOME%\bin;기존경로

- 설정확인 : CMD창열기

```
C:\Users\troment>java -version
openjdk version "1.8.0_192"
OpenJDK Runtime Environment (Zulu 8.33.0.1-win64) (build 1.8.0_192-b01)
OpenJDK 64-Bit Server VM (Zulu 8.33.0.1-win64) (build 25.192-b01, mixed mode)

C:\Users\troment>javac -version
javac 1.8.0_192
```

- Eclipse & STS & Apache Tomcat : 압출풀기

```
c:\SSAFY\eclipse\java
      \algorithm
      \sts
\workspace\java
      \algorithm
      \sts
\apache-tomcat9.x.x
```

- Eclipse & STS 설치후 설정 : ini file

```
-Dfile.encoding=UTF-8
```

JAVA

자바의 특성

1. 이식성이 높은 언어이다 ... platform 독립.(운영체제 상관 X)
2. 객체 지향 언어이다 ...
3. 메모리를 자동으로 관리한다 ... Garbage Collector
4. 멀티스레드를 쉽게 구현할 수 있다 ...
5. 동적 로딩(Dynamic Loading)을 지원한다 ...
6. 오픈소스 라이브러리가 풍부하다

java code 실행과정

사람이 인식할수 있는 코드		기계(JVM)가 인식할수 있는 코드 (== bytecode)		
HelloJava.java	compile	HelloJava.class	run	결과
	javac HelloJava.java		java HelloJava	Hello Java

lab 진행 순서.

- 변수 : data type(자료형)
- 연산자 : 산술, 논리, 비교, 삼항, 비트, 쉬프트,
- 제어문
 1. 조건문 : if, if ~ else, if ~ else if ~ else, switch ~ case
 2. 반복문 : for, while, do ~ while
 3. break, continue, return
- method (기능)
- class (객체) : 생성자, class 상속, 추상(abstract), interface, interface 구현(implements),

- package
 1. 사용자 정의 package
 2. system package
 - java.lang
 - java.util
 - java.io
 - java.sql
 - etc...

변수 (Variable) : 하나의 값을 저장하는 공간.

- 사용방법.(기본[Primitive] DataType 변수)

1-1. 선언.

형식) 접근제어자 DataType 변수명;

예) `public int age;`
`private boolean flag;`
`char c;`
`String name;(Object..)`

1-2. 사용.

형식) 변수명 = 값;

예) `age = 28;`
`flag = true;`
`c = 'a';`
`name = "안효인";`

>>>> 1, 2를 동시에.. 접근제어자 DataType 변수명 = 값;

```
public int age = 28;  
private boolean flag = true;  
char c = 'a';  
String name = "안효인";
```

■ 접근제어자(access modifier) : 전역변수

- a. `private` : 같은 클래스 안에서 사용가능.
- b. 생략(default) : `private` + 같은 package에서 사용가능.
- c. `protected` : `default` + 두클래스가 상속관계일 경우 package가 달라져도 사용가능.
- d. `public` : 접근 제한 없이 모든 클래스에서 사용가능.

■ 자료형 (Data Type)

■ 기본(Primitive) Data Type

Data Type	Data	bytes	표현
boolean	참과거짓	1	true, false [0, 1 : X]
char	문자	2	'a', '1', '김' >> O, 'ab' >> X
			특수문자 : '\n', '\t', '\'', '\"', '\\'
byte	정수	1	-128 ~ 127
short	정수	2	
int	정수	4	약21억
long	정수	8	
float	실수	4	
double	실수	8	

■ 참조형(Reference) Data Type

- Object(class) : 문자열 (String)
- 배열(array)

■ 명명규칙

1. 클래스는 대문자로, 메소드와 변수는 소문자로 시작
2. 합성어의 경우 단어의 첫글자 대문자 >> camel case (carName)
3. 숫자로 시작 X, 중간이나 끝은 가능.
4. 특수문자 X (단, \$ _는 가능.)
5. 예약어 X
6. 의미부여.

■ 주석문 : 제외하고 compile

// : 한줄 주석

/* ~ */ : 여러줄 주석

/** ~ */ : document 주석

연산자 (Operator)

1. 단항연산자 : +, -, ++, --, !, (type) : 형변환

- 1증가(감소)

선증가(감소) 후실행 : ++숫자형변수

선실행 후증가(감소) : 숫자형변수++

2. 산술연산자 : +, -, *, /, %(나머지)

3. 관계연산자 : >, >=, <, <=, ==, !=, instance of

4. 논리연산자 : &&, ||, !(not)

형식) 피연산자1 &&(||) 피연산자2

- and : &&

and	T	F
T	T	F
F	F	F

- or : ||

or	T	F
T	T	T
F	T	F

▪ not : !

!	T	F
	F	T

5. 대입연산자(=) : 변수 = 값; >> 모든 연산의 최후위연산자

```
int x = 4 + 5 * 2 - 6;
```

6. 연산후대입연산자 : +=, -=, *=, /=, %=, ...

```
x += 3; == x = x + 3;
```

7. 삼항연산자 : 조건식 ? "값1" : "값2"

```
int jumin = 3;
String gender = jumin % 2 == 0 ? "여자" : "남자";
```

자판기 거스름돈

김싸피는 자판기에서 2370원인 음료를 뽑으려한다.
현재 가지고 있는 돈은 10000원권 1장이다.
거스름돈은 각각 몇장(개)를 받아야 할까?

결과)
투입금액 : 10000원
물건금액 : 2370원
거스름돈 : 7630원

5000원 : 1장.
1000원 : 2장.
500원 : 1개.
100원 : 1개.
50원 : 0개.
10원 : 3개.
、、、

제어문

조건문

1. if문

형식)

```
if(조건식) {  
    실행문1;  
}  
실행문2;
```

결과)

```
조건식 true   : 실행문1 >> 실행문2  
false  : 실행문2
```

2. if ~ else문

형식)

```
if(조건식) {  
    실행문1;  
} else {  
    실행문2;  
}  
실행문3;
```

결과)

```
조건식 true   : 실행문1 >> 실행문3  
false        : 실행문2 >> 실행문3
```

3. if ~ else if ~ else문

형식)

```
if(조건식1) {  
    실행문1;  
} else if(조건식2) {  
    실행문2;  
} else {  
    실행문3;  
}  
실행문4;
```

결과)

```
조건식1 true : 실행문1 >> 실행문4  
false       : 조건식2 true : 실행문2 >> 실행문4  
false       : 실행문3 >> 실행문4
```

4. switch ~ case문 : 값의 비교는 숫자형만 가능. (단, jdk7이상부터 문자열(String)도 가능)

형식)

```
switch(값) {  
    case 값1: 실행문1; break;  
    case 값2: 실행문2; break;  
    case 값3: 실행문3; break;  
    .  
    case 값N: 실행문N; break;  
    default: 실행문D;  
}
```

결과)

switch의 값과 case의 값N과 같은 실행문 실행.

학점관리

김싸피교수는 중간고사 결과를 채점하고 점수에 따른 학점을 부여하고자 한다.
아래 조건에 맞는 학점을 출력해보자.

조건 1. 점수가 90이상 A학점.

80이상 B학점.

70이상 C학점.

70미만 F학점.

조건 2. x5점 이상은 '+'가 붙는다.

예) 98점은 A+학점.

학생의 점수가 98점일경우 학점을 구하세요.
(점수는 여러 점수로 바꾸어 테스트 하세요.)

결과)

점수가 98점이므로 A+입니다.

반복문

1. **for문** : 반복구문이 있을 경우, 횟수가 정해져있을 경우, 규칙이 있을 경우.

형식)

```
for(초기값;조건식;증감식) {  
    실행문;  
}
```

결과)

초기값 >> 조건비교 >> true >> 실행문 >> 증감식 >> 조건비교

2. **while문** : only 조건식.

형식)

```
while(조건식) {  
    실행문;  
}
```

결과)

조건식이 true일 경우 실행문반복.

■ 무한루프

```
while(true) {  
  
    if(조건)  
        반복문종료  
}
```

3. **do ~ while문** : 최소 1번이상 실행해야 하는 구문이 있을 경우. 선실행 후조건.

형식)

```
do {  
    실행문;  
} while(조건식);
```

결과)

선 실행 후 조건식이 true일 경우 실행문반복.

행운의 로또

김싸피는 싸피 입과 기념으로 로또를 구입하려 한다.

결정장애가 있는 김싸피는 랜덤 로또번호 생성 프로그램을 구현하려한다.

아래 조건에 맞게 프로그램을 구현해 보자.

조건 1. 1 ~ 45까지의 난수.

조건 2. 중복 숫자는 불가.

조건 3. 게임당 5개의 숫자 발생.

조건 4. 게임수 입력은 아래의 코드를 참고한다.

```
import java.util.Scanner;  
.  
.  
Scanner sc = new Scanner(System.in);  
int cnt = sc.nextInt();
```

결과)

게임수 입력 : 3

=== 금주 행운의 로또번호 ===

1. 3 5 23 27 44

2. 7 22 34 38 42

3. 1 2 3 4 5