자바의 소개및 기본구문

자바의소개&개요

1. 자바(Java) 소개
2. 자바(Java) 특징
3. 자바(Java) 프로그램 종류
4. 자바(Java) 플랫폼 종류
5. 자바 개발 및 실행환경 구축
6. 자바 프로그램 기본 구조(Desktop Application)
7. Application의 컴파일과 실행
8. 문자와 문자열
9. 정수의 오버플로우(Overflow)
10. 형변환(Casting)

1. 자바(Java) 소개

▶ 객체지향 프로그래밍 (Object Oriented Programming) 언어이다

1991년 Sun Microsystems의 James Gosling에 의해 다양한 가전제품 통합 제어를 위해 최초 고안됨(Green Project) – OAK라 명명, 프로젝트 실패

1995년 인터넷의 급격한 발전과 더불어 웹에 적용할 수 있는 언어로 탈바꿈

1995년 5월 23일 Java라는 이름으로 공식 발표 – Java 1.0

개발환경 지원 – JDK 1.0(Java Development Kit)

지원 클래스 : 약 250개

2015년 Java 8

2020년 Java 15

▶ C++ 기반으로 만들어진 언어이긴 하지만, C++처럼 복잡한 형태가 아닌 단순하면서도 객체지향 개념을 완벽하게 구현할 수 있는 형태로 구성됨

구조체(Struct), 공용체(Union), 포인터(Pointer)를 지원하지 않음

Garbage Collection 기능을 통한 메모리 자동 관리 등

▶ 현재 자바는 데스크톱 어플리케이션, 서버 정보시스템, 무선기기와 같이 서로 다른 OS에 배치되어 시스템 개발 및 실행 환경(Platform)으로 제공되고 있다.

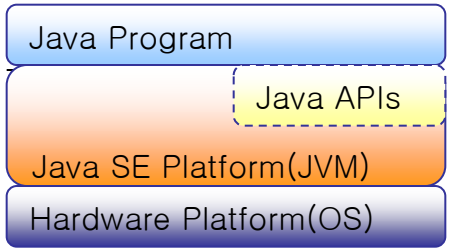
2. 자바(Java) 특징

- 단순한 문법(Simple)
  - C나 C++에서 프로그래머에게 많은 혼란을 주는 요소(전처리, 포인터, 구조체, 공용체, 다중상속 등)들을 제거함
- 객체지향 언어(Object Oriented Language)
  - 추상화, 캡슐화, 상속, 다형성 등과 같은 특성을 완벽하게 지원함
- 플랫폼 독립성(Platform Independence)
  - 소스코드(*.java) 컴파일을 통해 하드웨어 의존적인 바이너리 코드가 중립적인 바이트 코드(*.class) 생성
  - 바이트 코드는 JVM(Java Virtual Machine:자바 실행환경)만 있으면 어떠한 시스템에서도 이를 해석하여 실행 가능
  - "Write Once, Use Anywhere!"
- 메모리 자동 관리(Garbage Collection)
  - Garbage Collector에 의해 필요 없는 메모리를 자동 처리
- 보안성
  - 바이트 코드 실행 전에 보안에 위배되는 요소가 있는지 여부를 미리 검사
- 쉬운 예외처리, 멀티스레드, 네트워킹, 분산 시스템 구축 등...

3. 자바(Java) 프로그램 종류

- Desktop Application
  - 워드프로세스, 메신저 등과 같이 개인용 PC에서 독립적으로 실행되어 특정한 기능을 수행하도록 작성된 프로그램(CUI/GUI)
- Applet
  - 웹 클라이언트(익스플로러, 파이어폭스 등) 내장되어 실행되는 작은 웹 프로그램
  - 웹 서버로부터 동적 다운로드
- Servlet/JSP
  - 웹 클라이언트의 HTTP 요청에 대해 HTML/XML 문서를 동적 생성하고 응답하기 위해 웹 어플리케이션서버에서 실행되는 프로그램
- Java Beans
  - Desktop Application, Servlet/JSP등에서 쓰일 수 있는 재사용 가능한 컴포넌트
- EJB(Enterprise Java Beans)
  - Desktop Application, Servlet/JSP, Java Beans 등에서 쓰일 수 있는 재사용 가능한 분산 컴포넌트
- Device Embed Application 등
  - 핸드폰, PDA등에 내장되어 실행되는 작은 프로그램

4. 자바(Java) 플랫폼 종류



- Java SE(Java Platform - Standard Edition)
  - Desktop이나 Server에서 Java Application/Applet등을 개발, 배치, 실행 할 수 있는 환경을 제공(Software Platform)
  - Java SE Development Kit 다운로드 및 설치 필요
    - https://www.oracle.com/kr/java/technologies/javase-downloads.html
    - Compiler, Interpreter, 표준 API 등 제공

- Java EE(Java Platform - Enterprise Edition)
  - Java SE를 기반으로 대규모 기업용 서버를 구축하고, 실행 할 수 있는 환경을 제공
  - Java EE SDK 다운로드 및 설치 필요
    - https://www.oracle.com/java/technologies/javaee-8-sdk-downloads.html
    - Web Application Server(GlassFish)와 Servlet, JSP, JDBC, DataSource, JPA, JTA, JNDI, RMI, EJB, JMS 등 다수의 API 제공

5. 자바 개발 및 실행환경 구축

- Java Development Kit (SE) 다운로드 및 설치
  - https://www.oracle.com/java/technologies/javase-downloads.html
  - jdk-16.0.1_windows-x64_bin.exe 실행
- Java SE API Document 보기
  - https://www.oracle.com/java/technologies/javase-jdk16-doc-downloads.html
- Eclipse 다운로드 및 설치
  - http://www.eclipse.org
- OS 환경 변수 설정

– PATH=C:\Program Files\Java\jdk-16.0.1\bin;

// 현재 위치에 상관없이 도스 명령어로 프로그램 실행

6. 자바 프로그램 기본 구조(Application)

- 문자열(텍스트)을 출력하는 Desktop Application

```
public class HelloWorld {
        public static void main(String[] args){
            // 문자열 출력 명령문(주석)
            System.out.println("Hello Java!");
            System.out.println("안녕 자바!");
        }
}
```

➢ 소스코드 작성시 주의 사항

- 영문 대소문자를 구별한다
- 저장시 파일명은 클래스명과 동일하여야 하며 확장자는 *.java이다
  예) HelloWorld.java
- 컴파일 : javac HelloWorld.java
- 실 행 : java HelloWorld

6. 자바 프로그램 기본 구조(Application)

- 클래스 정의
  - 자바는 클래스 단위로 프로그램을 작성하기 때문에 소스파일 안에 반드시 클래스를 정의해야 한다.
  - 클래스의 이름은 첫 문자를 대문자로 시작하는 것이 관례이다
  - 클래스의 구성요소(속성, 메소드 등...)들은 { } 안에 위치한다
  - 소스파일 저장 시 파일명이 클래스명과 반드시 일치해야 한다

- main(String[] args) 메소드 정의
  - Desktop Application이 실행되려면 최소 1개 존재하여야 한다
  - 프로그램의 진입점으로 JVM에 의해 최초 호출되며
  - 메인 메소드 블록 내부에 기술된 명령문들을 순차적으로 실행한다
  - JVM은 세미콜론(;)으로 끝나는 문장을 하나의 명령문으로 인식한다
  - 주석은 프로그램 소스코드를 쉽게 이해 하기 위해서 사용하며, 컴파일 및 실행에 영향을 미치지 않는다

7. Application의 컴파일과 실행

- 소스 파일 컴파일

  DOS> javac HelloWorld.java

  소스파일을 컴파일하고 나면 byte code로 이뤄진 class파일이 생성됨

- Application 실행

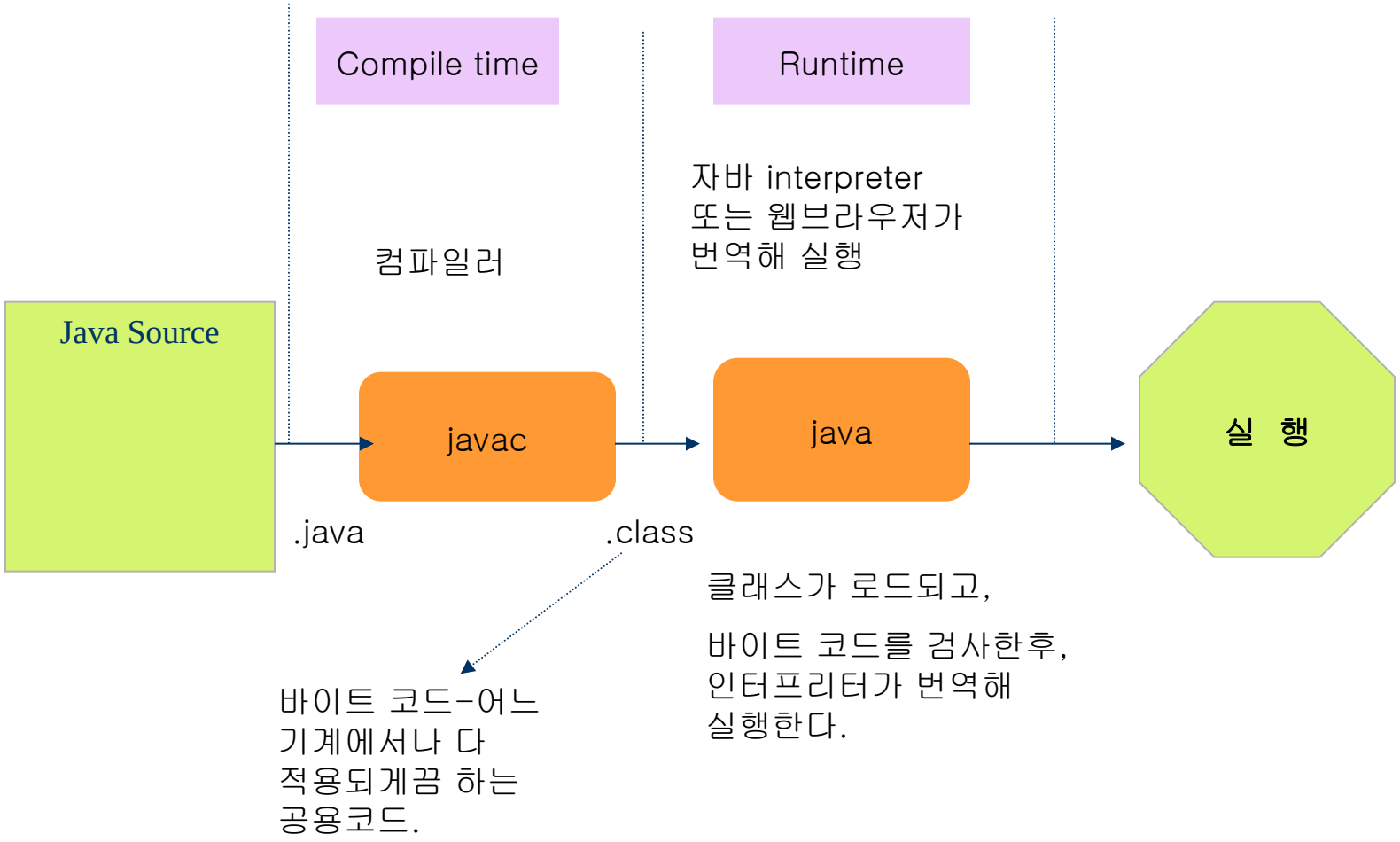
  DOS> java HelloWorld

  클래스 파일을 실행하기 위해서는 자바 해석기인 java를 이용해야 한다.

  실행하면 Hello World!가 출력된다.

7. Application의 컴파일과 실행

컴파일 및 실행과정



8. Application예제 소스분석

import java.lang.System;

- import문이 먼저 온다.

  (주의: package 선언문이 있을 경우에는 package선언문이 항상 최상단에위치함)
- import java.lang.*; 해도 됨.

  보통 lang패키지 import는 생략 가능.

public class HelloWorld{

- public 은 접근 지정자.
- HelloWorld는 클래스명
- 파일명은 이 클래스명과 동일해야.

public static void main(String args[]){

- main메소드는 자바 프로그램을 실행할때 JVM에서 호출되는 최초의 메소드
- main메소드는 각 프로그램의 시작 지점인 동시에 종료 지점임.
- 여러 클래스로 이뤄진 프로그램의 경우에는 반드시 시작클래스(주 클래스)는 main()를 갖고 있어야 한다.

}

}

9. 자바 기본 구문 – 주석

- **Singleline Comment**
  - // 뒤에 한 라인에 대하여 주석 처리
- **Multiline Comment**
  - /* … */ 범위의 모든 라인에 대하여 주석 처리
- **Document Comment**
  - /** … */ 범위의 모든 라인에 대하여 주석 처리
  - 클래스나 메소드 앞에 사용되어지며, javadoc.exe(도큐먼트 생성툴)를 이용하여 HTML Document 생성시 주석내용이 문서에 포함된다
  - 주석 내용에 HTML 태그 사용 가능

10. 자바 기본 구문 – 예약어(Keyword)

abstract	const	finally	int	public	throw
boolean	continue	float	interface	return	throws
break	default	for	long	short	transient
byte	do	goto	native	static	try
catch	else	implements	package	switch	volatile
char	extends	import	private	synchronized	assert
class	final	instanceof	protected	this	enum

- Java의 모든 예약어는 소문자이다
- const와 goto는 예약어로 지정되어 있지만 사용되지 않는다

11. 자바 기본 구문 – 식별자

- **식별자(Identifier)**
  - 프로그램 구성요소인 변수, 상수, 배열, 메소드, 클래스 등을 구분하기 위해 사용자가 정의하는 이름
- **식별자 규칙**
  - 대소문자를 구분하며, 첫 글자는 영문자나 특수문자('_','$')로 시작되어야 한다.
  - 첫 글자로 숫자를 사용할 수 없다. 첫 글자 외에 사용하는 것은 허용한다
  - 예약어(this, true, null 등…)는 식별자로 사용할 수 없다
  - 16비트 유니코드를 지원하므로 한글도 식별자로 사용 가능하다(비권장)
    - 아스키코드 : ANSI(American National Standards Institute: 미국규격협회)에서 제정한 8비트 문자코드로 256개의 문자를 코드화
    - 유니코드 : 유니코드(Apple, IBM, MS등의 컨소시엄)에서 제정한 16비트로 확장한 문자코드로 전세계의 모든 문자를 표현하기 위한 표준 문자 코드이다.
    - 유니코드는 현재 34,168개의 글자들을 코드화 하고 있으며 최대 65,536개의 글자를 코드화 할 수 있다
- **식별자 관례**
  - 클래스 이름은 대문자로 시작하고, 변수, 메소드 등의 이름은 소문자로 시작하는 것이 관례이다
  - 두 단어를 조합하여 이름을 정 할 때는 조합하는 문자의 첫 글자는 대문자로 한다
    - Camel 표기법