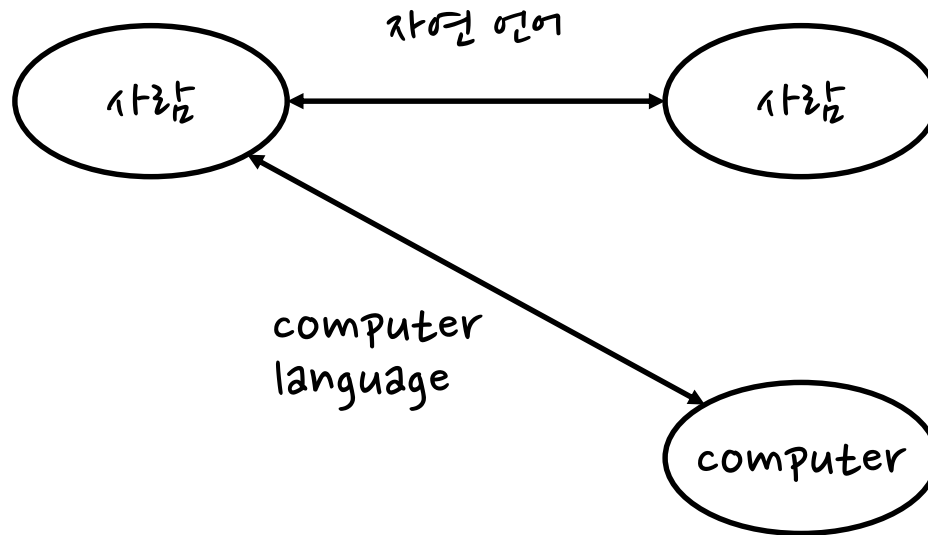


컴퓨터 프로그래밍

Basic concepts and a Sample

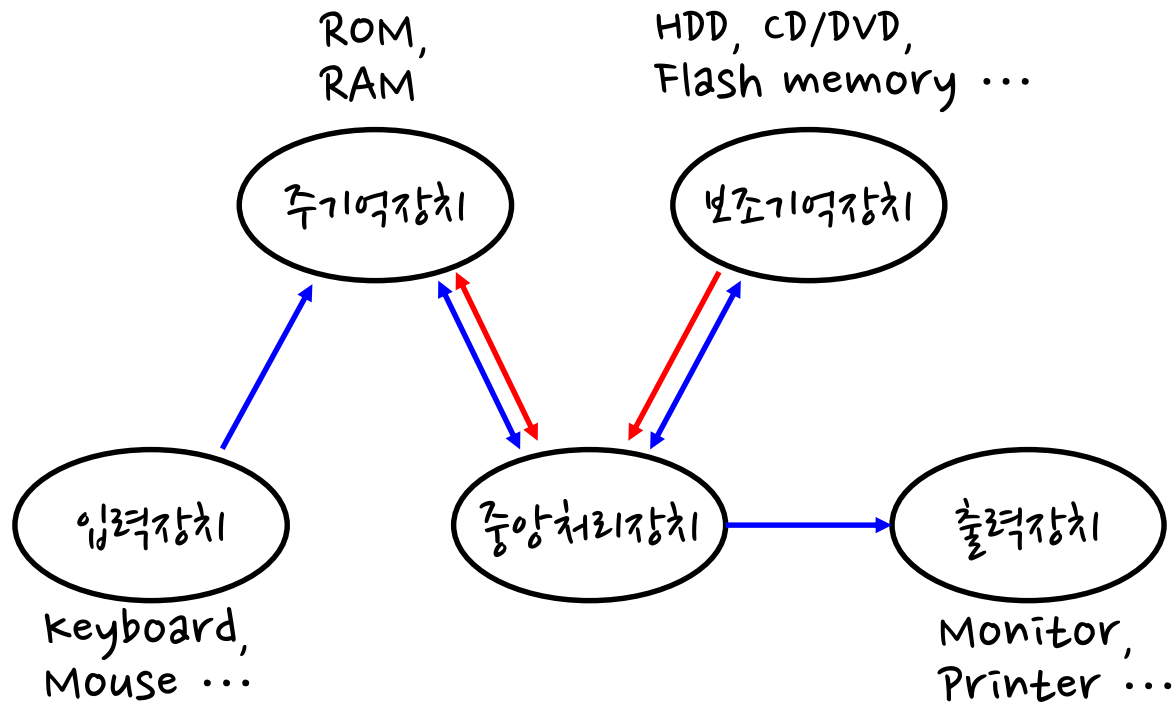
computer Programming Language

- 사람들이 컴퓨터와 대화하기 위해 만들어낸 언어 (language)



Hardware & Software

- Program and Data

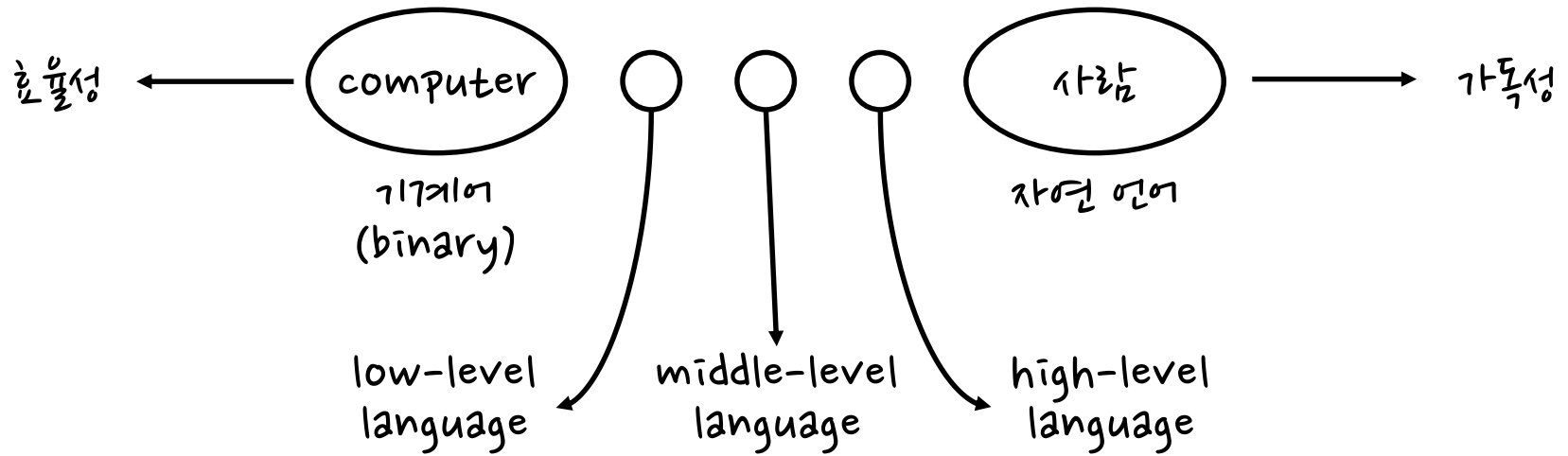


Programming Language

- Program
 - Sequence of instructions
- Programming language
 - Machine language
 - Assembly language
 - Procedural language / Nonprocedural language

기계어	1001 0001 1100 0010 1010 0011	0001 위치의 값을 누산기(AC)에 저장하라 누산기에 0010 위치의 값을 더하라 누산기의 값을 0011 위치에 저장하라
어셈블리어	LOAD Y ADD Z STORE X	Y의 값을 누산기에 저장하라 누산기에 Z의 값을 더하라 누산기의 값을 X에 저장하라
고급 언어	$X = Y + Z$	Easy to understand!!! FORTRAN, Basic, C, C++, Java, Python, ...

Programming Language



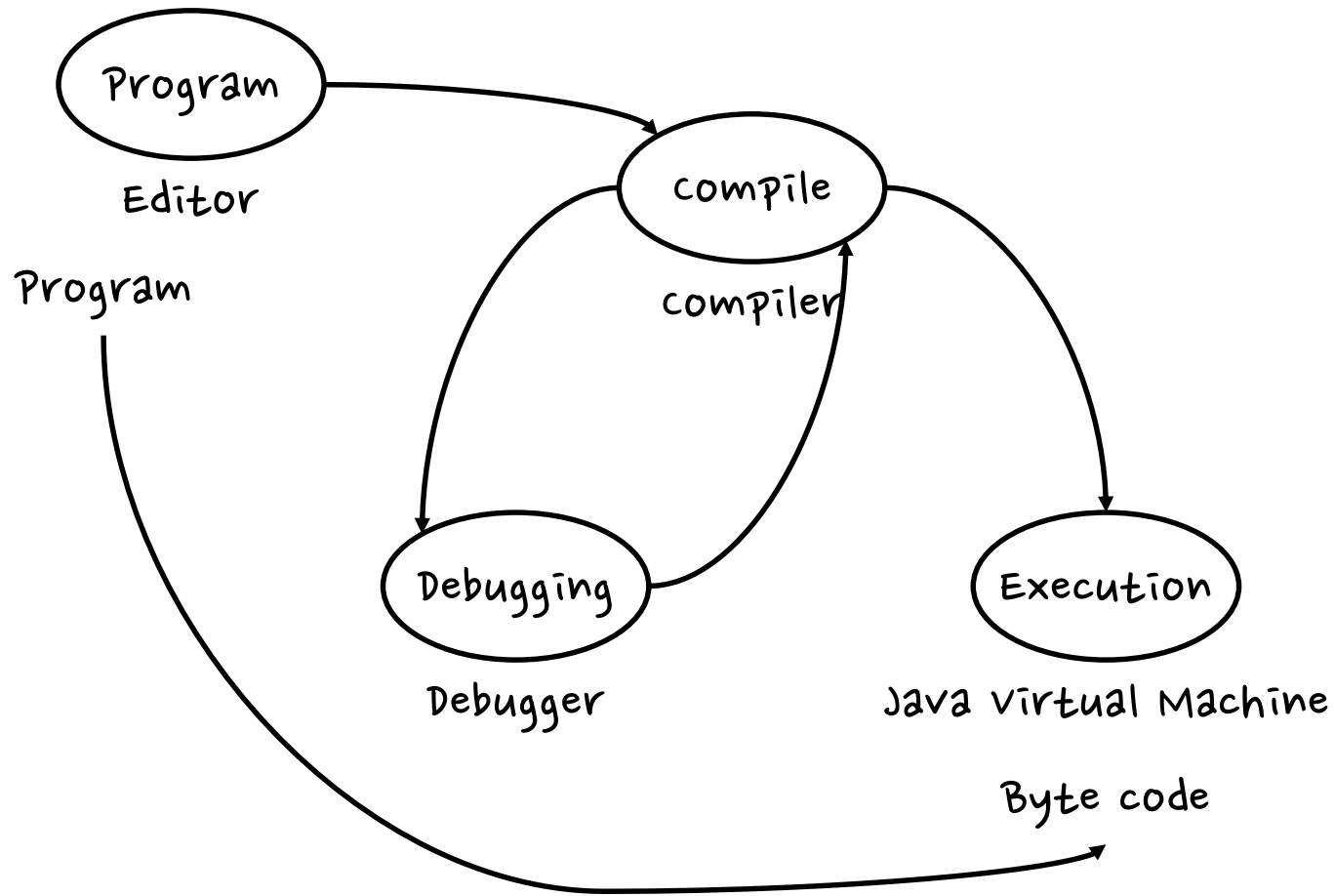
compile

- Source code / file
 - 각종 프로그래밍 언어로 개발자가 작성한 프로그램 코드
 - 프로그래밍 언어의 문법을 따르는 텍스트 파일
- compile
 - Source code를 기계어로 변환하는 작업
- compiler
 - compile을 수행하는 소프트웨어
 - 각 프로그래밍 언어마다 고유의 컴파일러가 존재

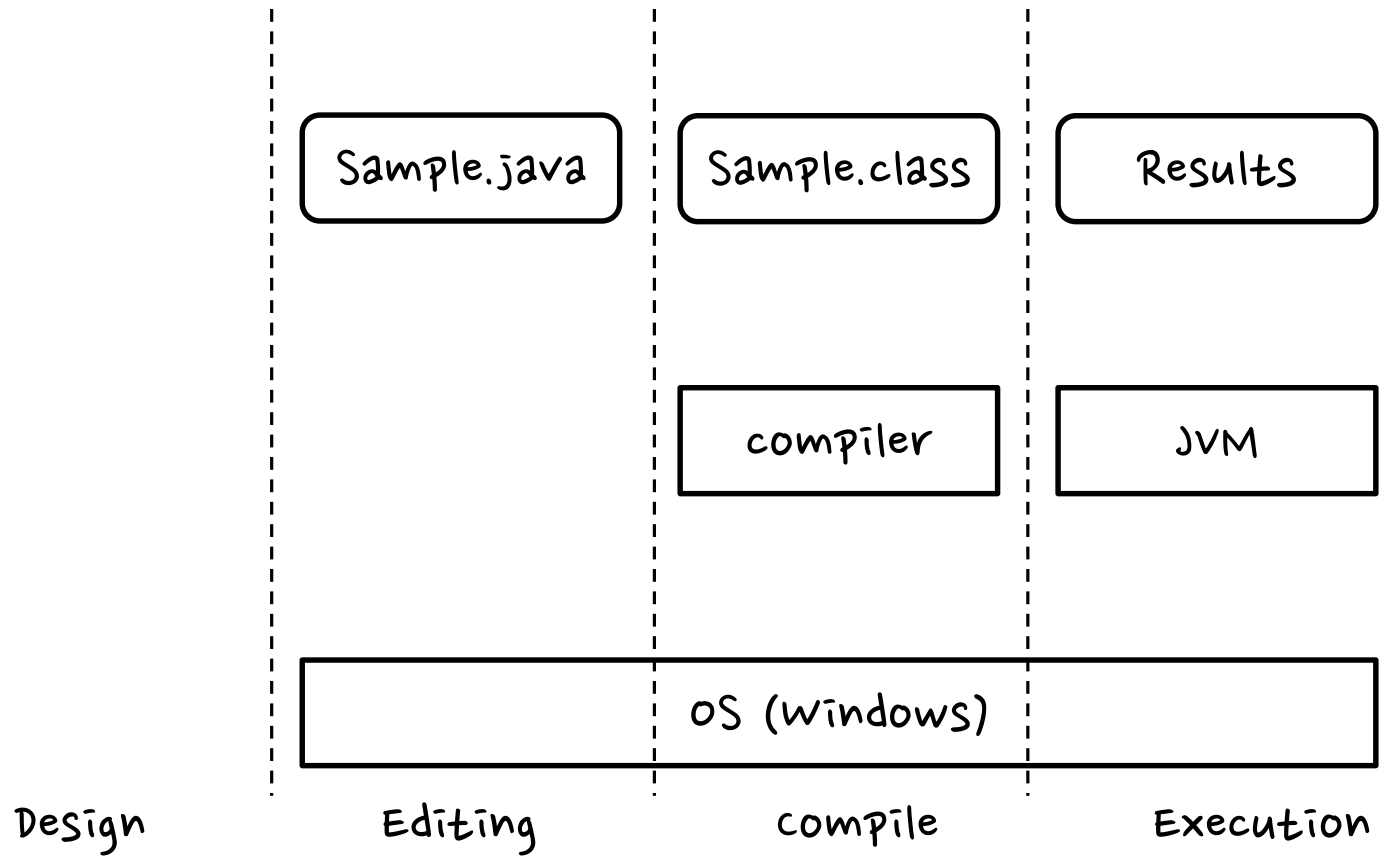
Java

- 1991년 SUN Microsystems, James Gosling
- Platform independent, low memory usage
- 1995년 발표 (Java 1.0)
 - Write Once, Run Everywhere
- 소형 기기, 인터넷의 발달과 함께 성장
- Java Virtual Machine (JVM) 상에서 동작
 - JVM은 platform별로 제공

Java Program



Procedure



Sample Program

```
/* 입력된 정수의 각 자리수 합을 구한다 */
import java.util.Scanner;

public class DigitSum {
    public static void main(String[] args) {
        int num;
        int digitSum = 0;
        Scanner input = new Scanner(System.in);

        System.out.print("정수를 입력하세요: ");
        num = input.nextInt();

        // 0보다 크면 반복
        while (num > 0) {
            digitSum = digitSum + num % 10;
            num = num / 10;
        }

        System.out.printf("자리수의 합: %d\n", digitSum);
    }
}
```

Sample Program

/* 입력된 정수의 각 자리수 합을 구한다 */

import java.util.Scanner;

public class DigitSum {

public static void main(String[] args) {

int num;

int digitSum = 0;

Scanner input = new Scanner(System.in);

System.out.print("정수를 입력하세요: ");

num = input.nextInt();

// 0보다 크면 반복

while (num > 0) {

digitSum = digitSum + num % 10;

num = num / 10;

}

System.out.printf("자리수의 합: %d\n", digitSum);

}

}

block: 프로그램의 단위

1 program = n blocks

1 block = k (nested) blocks

Sample Program

```
/* 입력된 정수의 각 자리수 합을 구한다 */
import java.util.Scanner;

public class DigitSum {
    public static void main(String[] args) {
        int num;
        int digitSum = 0;
        Scanner input = new Scanner(System.in);

        System.out.print("정수를 입력하세요: ");
        num = input.nextInt();

        // 0보다 크면 반복
        while (num > 0) {
            digitSum = digitSum + num % 10;
            num = num / 10;
        }

        System.out.printf("자리수의 합: %d\n", digitSum);
    }
}
```

statement: 명령의 기본 단위
";"로 끝나는 명령 집합

연산(operation)
- 산술, 논리, 비교 등 계산
대입(assignment)
- 변수에 새로운 값을 지정

Sample Program

```
/* 입력된 정수의 각 자리수 합을 구한다 */
import java.util.Scanner;

public class DigitSum {
    public static void main(String[] args) {
        int num;
        int digitSum = 0;
        Scanner input = new Scanner(System.in);

        System.out.print("정수를 입력하세요: ");
        num = input.nextInt();

        // 0보다 크면 반복
        while (num > 0) {
            digitSum = digitSum + num % 10;
            num = num / 10;
        }

        System.out.printf("자리수의 합: %d\n", digitSum);
    }
}
```

Method

- 일정한 기능을 수행하는 단위

main method

- 프로그램의 시작점

Method? Function?

- 동일한 개념
- class 내부의 function을 method라 부른다
- Java에서 모든 function은 class 내부에 있으므로 모두 method

Sample Program

```
/* 입력된 정수의 각 자리수 합을 구한다 */
```

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

```
public class DigitSum {  
    public static void main(String[] args) {  
        int num;  
        int digitSum = 0;  
        Scanner input = new Scanner(System.in);  
  
        System.out.print("정수를 입력하세요: ");  
        num = input.nextInt();  
  
        // 0보다 크면 반복  
        while (num > 0) {  
            digitSum = digitSum + num % 10;  
            num = num / 10;  
        }  
  
        System.out.printf("자리수의 합: %d\n", digitSum);  
    }  
}
```

import 선언

- 특정 class를 사용함을 명시
- Scanner라고 쓰면
java.util package에 있는
Scanner class를 말함

Sample Program

```
/* 입력된 정수의 각 자리수 합을 구한다 */
```

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

```
public class DigitSum {
```

```
    public static void main(String[] args) {
```

```
        int num;
```

```
        int digitSum = 0;
```

```
        Scanner input = new Scanner(System.in);
```

```
        System.out.print("정수를 입력하세요: ");
```

```
        num = input.nextInt();
```

```
        // 0보다 크면 반복
```

```
        while (num > 0) {
```

```
            digitSum = digitSum + num % 10;
```

```
            num = num / 10;
```

```
        }
```

```
        System.out.printf("자리수의 합: %d\n", digitSum);
```

```
    }
```

```
}
```

주석(annotation, comment)
- 프로그램의 이해를 돕기 위한 추가문
- compiler는 주석 무시

개발 환경 설정

- Java Development Kit (JDK)
 - java.sun.com

The image displays two screenshots of the Oracle Java SE Downloads website. The left screenshot shows the 'Java SE Downloads' page, where the 'Java Platform (JDK) 7u25' download button is highlighted with a red box. The right screenshot shows the 'Java SE Development Kit 7u25' download page, where the 'Accept License Agreement' radio button is selected, and the download links for Windows x86 and x64 are highlighted with a red box.

Java SE Downloads

Overview Downloads Documentation Community Technologies Training

Next Releases (Early Access) Embedded Use Previous Releases

Java Platform (JDK) 7u25

Java Platform, Standard Edition

Java SE 7u25

This release includes important security fixes. Oracle strongly recommends that all Java SE 7 users upgrade to this release. Learn more

Which Java package do I need?

- JDK:** (Java Development Kit). For Java Developers. Includes a complete JRE plus tools for developing, debugging, and monitoring Java applications.
- Server JRE:** (Server Java Runtime Environment) For deploying Java applications on servers. Includes tools for JVM monitoring and tools commonly required for server applications, but does not include browser integration (the Java plug-in), auto-update, nor an installer. Learn more
- JRE:** (Java Runtime Environment). Covers most end-users needs. Contains everything required to run Java applications on your system.

JDK **Server JRE** **JRE**

Java SE Development Kit 7u25

You must accept the Oracle Binary Code License Agreement for Java SE to download this software.

☐ Accept License Agreement ☒ Decline License Agreement

Product / File Description	File Size	Download
Linux ARM v6/v7 Soft Float ABI	65.12 MB	jdk-7u25-linux-arm-sfp.tar.gz
Linux x86	80.38 MB	jdk-7u25-linux-i586.rpm
Linux x86	93.12 MB	jdk-7u25-linux-i586.tar.gz
Linux x64	81.46 MB	jdk-7u25-linux-x64.rpm
Linux x64	91.85 MB	jdk-7u25-linux-x64.tar.gz
Mac OS X x64	144.43 MB	jdk-7u25-macosx-x64.dmg
Solaris x86 (SVR4 package)	136.02 MB	jdk-7u25-solaris-i586.tar.Z
Solaris x86	92.22 MB	jdk-7u25-solaris-i586.tar.gz
Solaris x64 (SVR4 package)	22.77 MB	jdk-7u25-solaris-x64.tar.Z
Solaris x64	15.09 MB	jdk-7u25-solaris-x64.tar.gz
Solaris SPARC (SVR4 package)	136.16 MB	jdk-7u25-solaris-sparc.tar.Z
Solaris SPARC	95.5 MB	jdk-7u25-solaris-sparc.tar.gz
Solaris SPARC 64-bit (SVR4 package)	23.05 MB	jdk-7u25-solaris-sparcv9.tar.Z
Windows x86	89.09 MB	jdk-7u25-windows-i586.exe
Windows x64	90.66 MB	jdk-7u25-windows-x64.exe

Java SE Development Kit 7u25 Demos and Samples Downloads

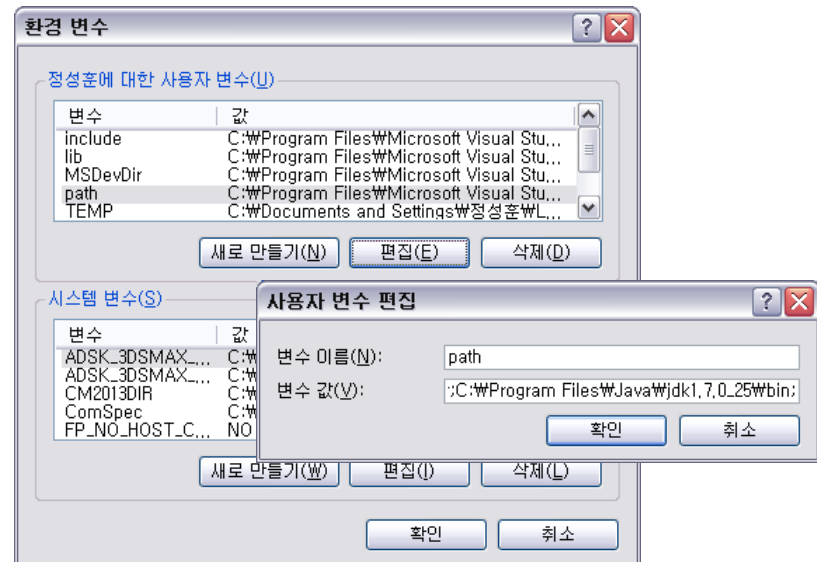
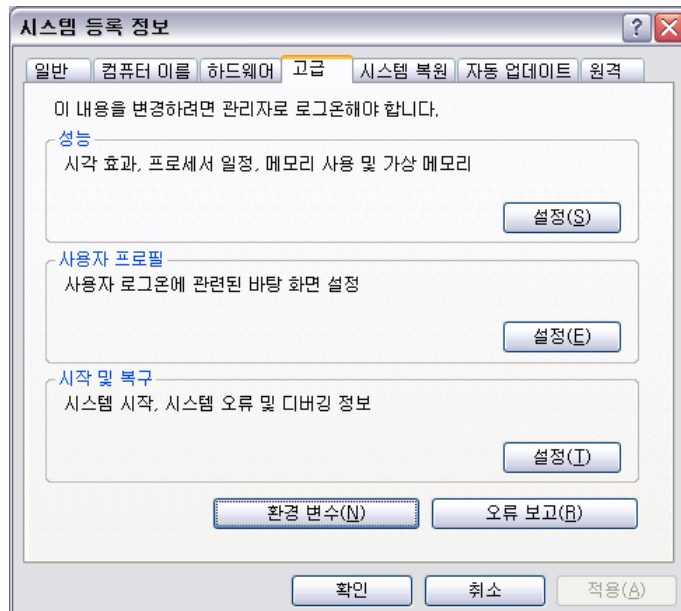
Java SE Development Kit 7u25 Demos and Samples Downloads are released under the Oracle BSD License

Product / File Description	File Size	Download
Linux ARM v6/v7 Soft Float ABI	8.1 MB	jdk-7u25-linux-arm-sfp-demos.tar.gz
Linux x86	14.15 MB	jdk-7u25-linux-i586-demos.rpm
Linux x86	14.08 MB	jdk-7u25-linux-i586-demos.tar.gz
Linux x64	14.21 MB	jdk-7u25-linux-x64-demos.rpm
Linux x64	14.14 MB	jdk-7u25-linux-x64-demos.tar.gz
Mac OS X	12.8 MB	jdk-7u25-macosx-x86_64-demos.tar.gz
Solaris x86	20.22 MB	jdk-7u25-solaris-i586-demos.tar.Z

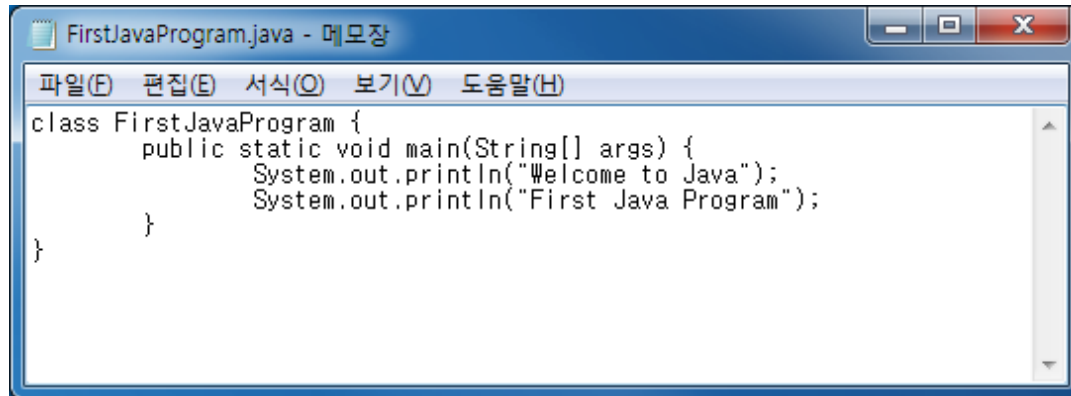
개발 환경 설정

- Path 설정

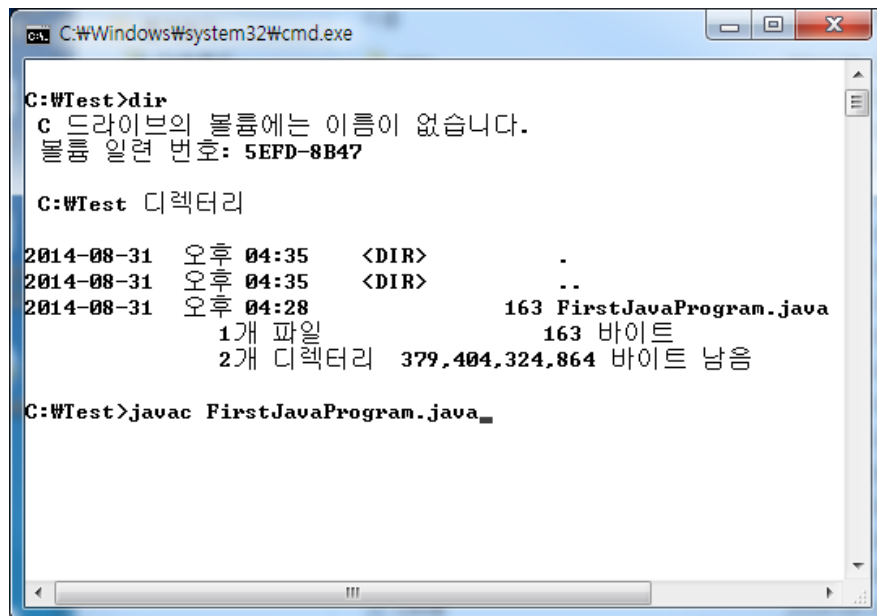
- Java virtual machine와 compiler의 실행을 위해
- 제어판 -> 시스템 및 보안 -> 시스템 -> 고급 시스템 설정 -> 환경변수
or 컴퓨터 우클릭 -> 속성 -> 고급 시스템 설정 -> 환경변수
- path 변수 -> 편집
- JDK 설치 경로의 bin 폴더 추가



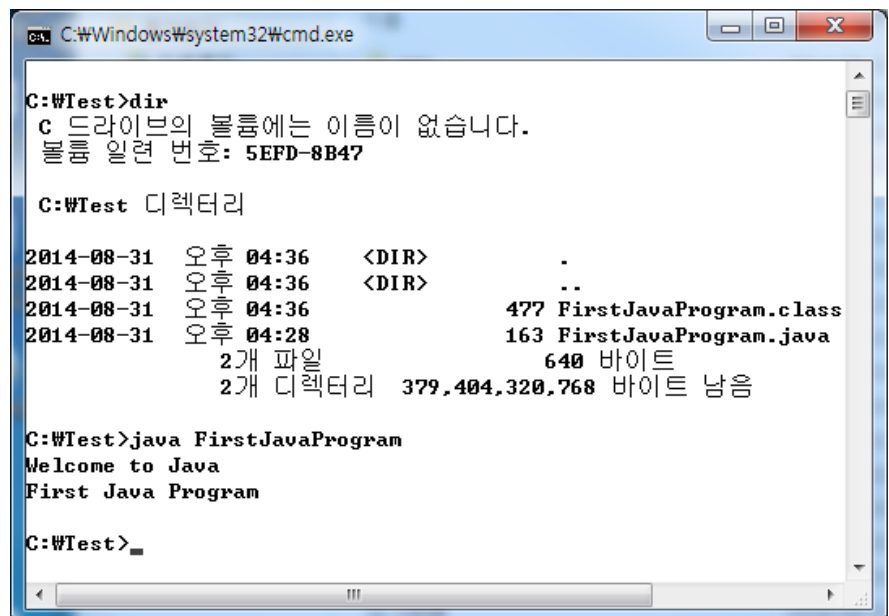
Program and Execution



```
class FirstJavaProgram {  
    public static void main(String[] args) {  
        System.out.println("Welcome to Java");  
        System.out.println("First Java Program");  
    }  
}
```

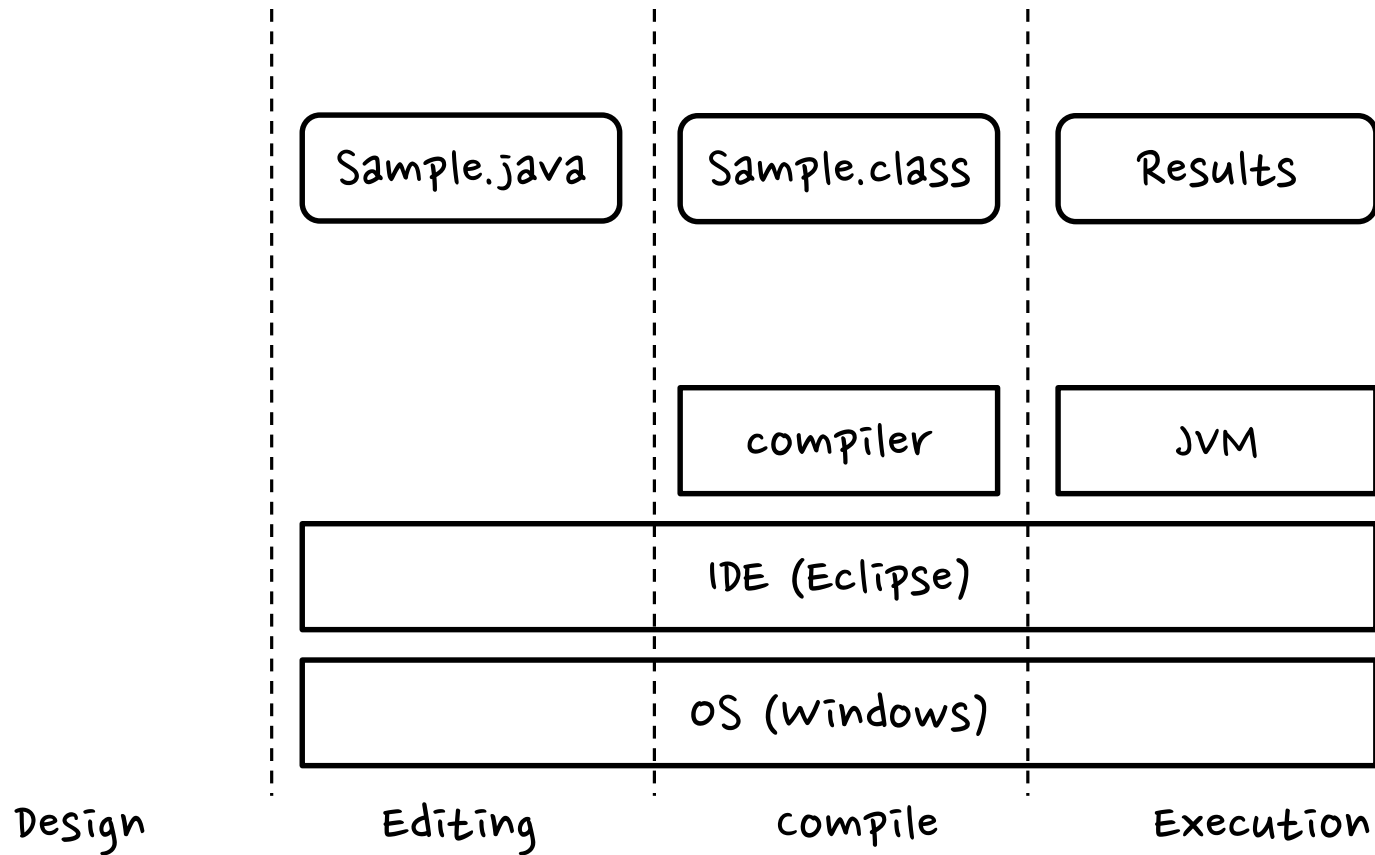


```
C:\Windows\system32\cmd.exe  
  
C:\WTest>dir  
C 드라이브의 볼륨에는 이름이 없습니다.  
볼륨 일련 번호: 5EFD-8B47  
  
C:\WTest 디렉터리  
  
2014-08-31 오후 04:35 <DIR> .  
2014-08-31 오후 04:35 <DIR> ..  
2014-08-31 오후 04:28 163 FirstJavaProgram.java  
1개 파일 163 바이트  
2개 디렉터리 379,404,324,864 바이트 남음  
  
C:\WTest>javac FirstJavaProgram.java_
```



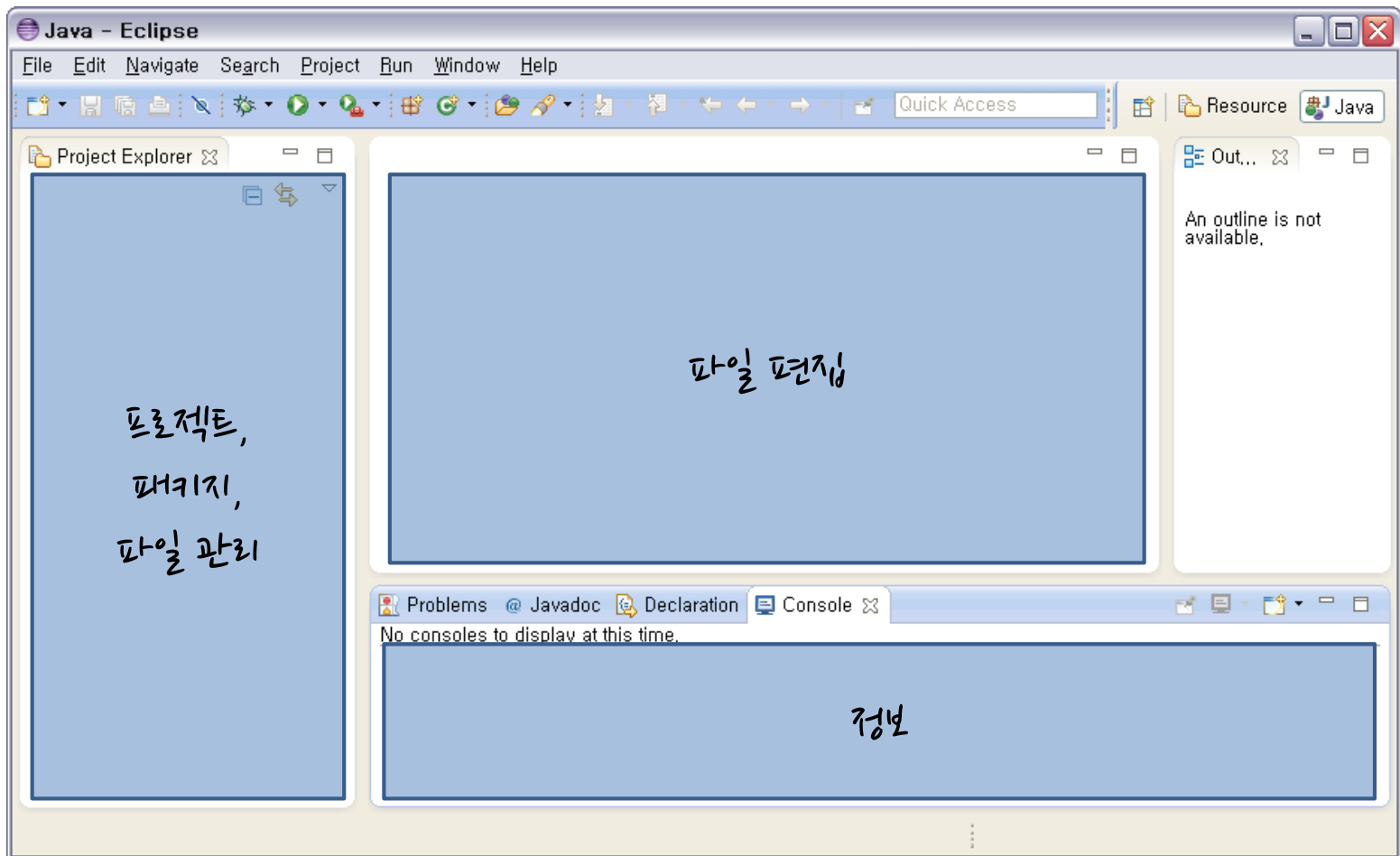
```
C:\Windows\system32\cmd.exe  
  
C:\WTest>dir  
C 드라이브의 볼륨에는 이름이 없습니다.  
볼륨 일련 번호: 5EFD-8B47  
  
C:\WTest 디렉터리  
  
2014-08-31 오후 04:36 <DIR> .  
2014-08-31 오후 04:36 <DIR> ..  
2014-08-31 오후 04:36 477 FirstJavaProgram.class  
2014-08-31 오후 04:28 163 FirstJavaProgram.java  
2개 파일 640 바이트  
2개 디렉터리 379,404,320,768 바이트 남음  
  
C:\WTest>java FirstJavaProgram  
Welcome to Java  
First Java Program  
  
C:\WTest>_
```

Integrated Development Environment



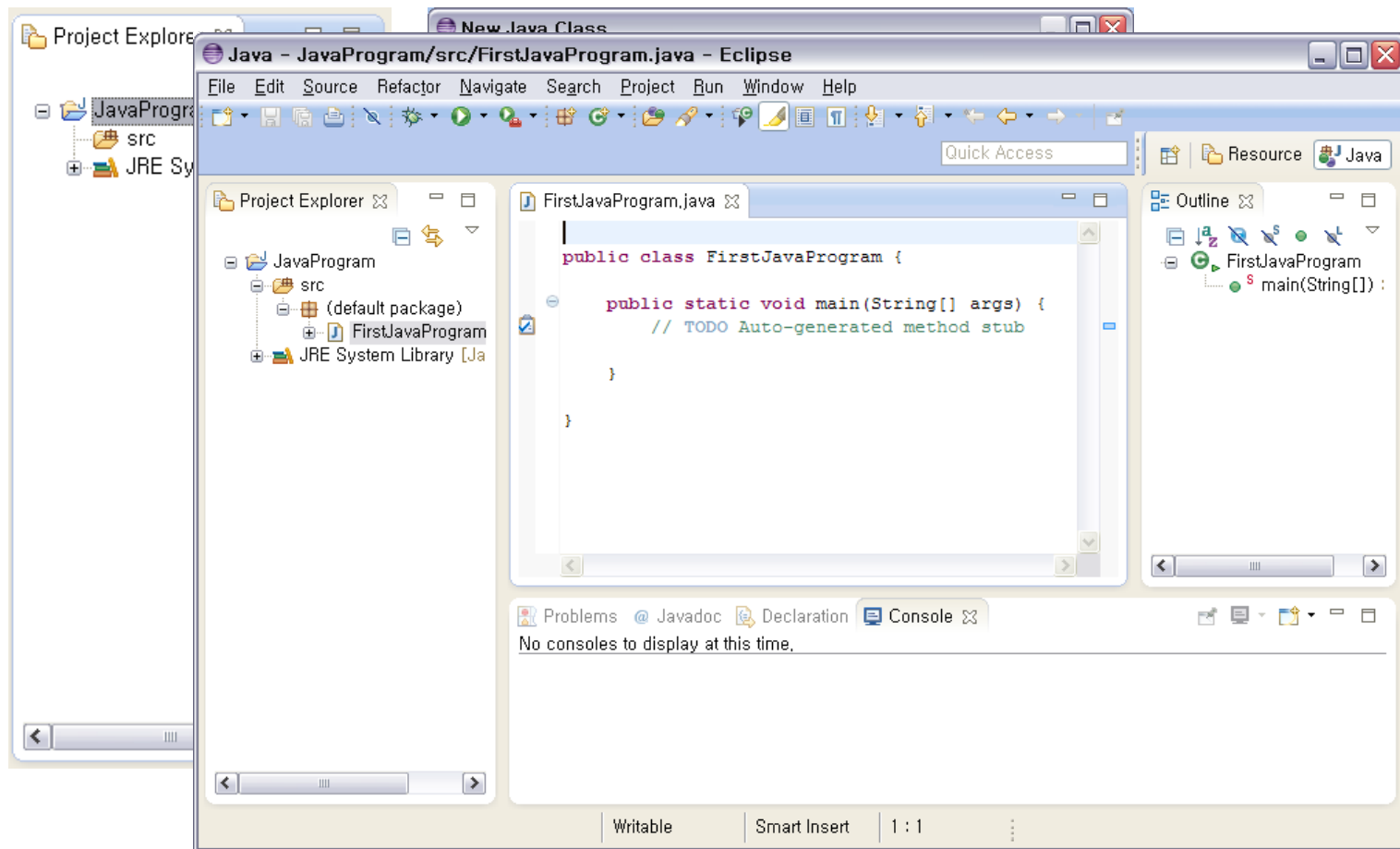
개발환경

- Eclipse

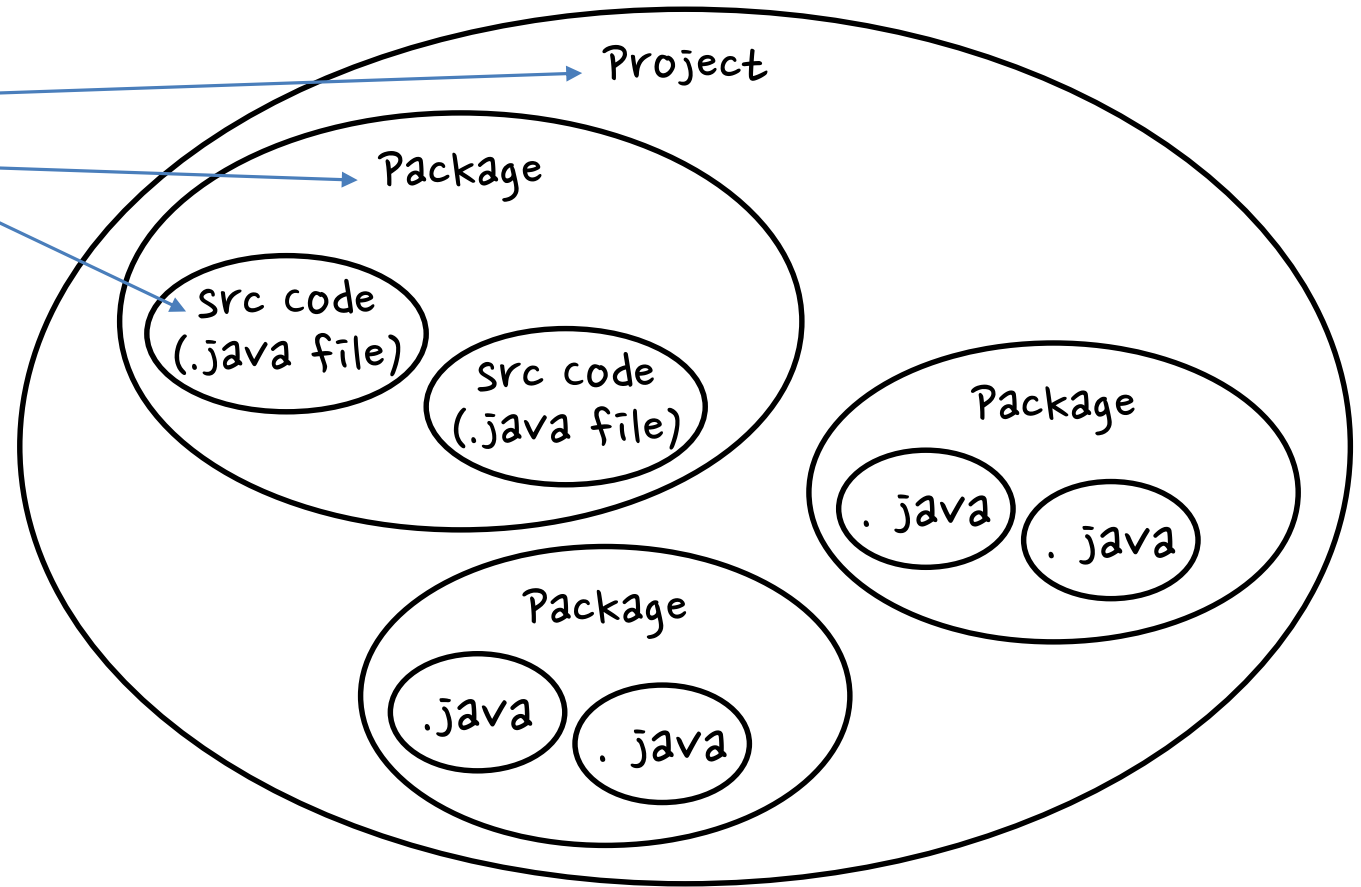
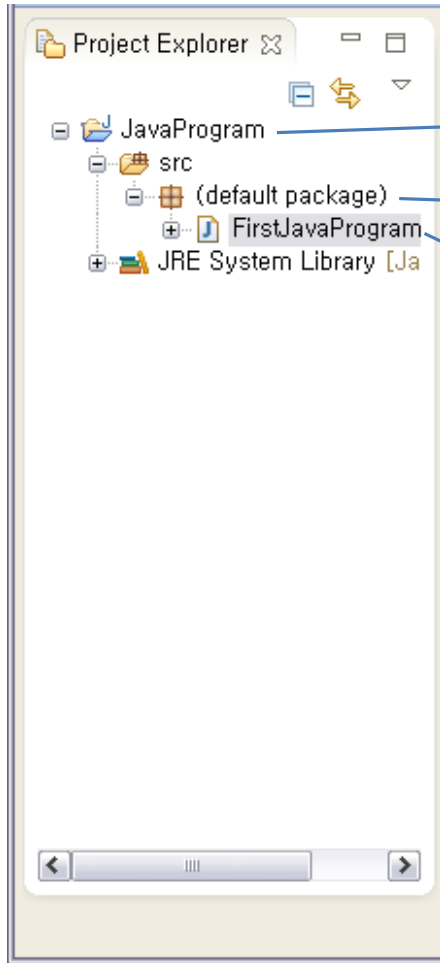


개발할 환경

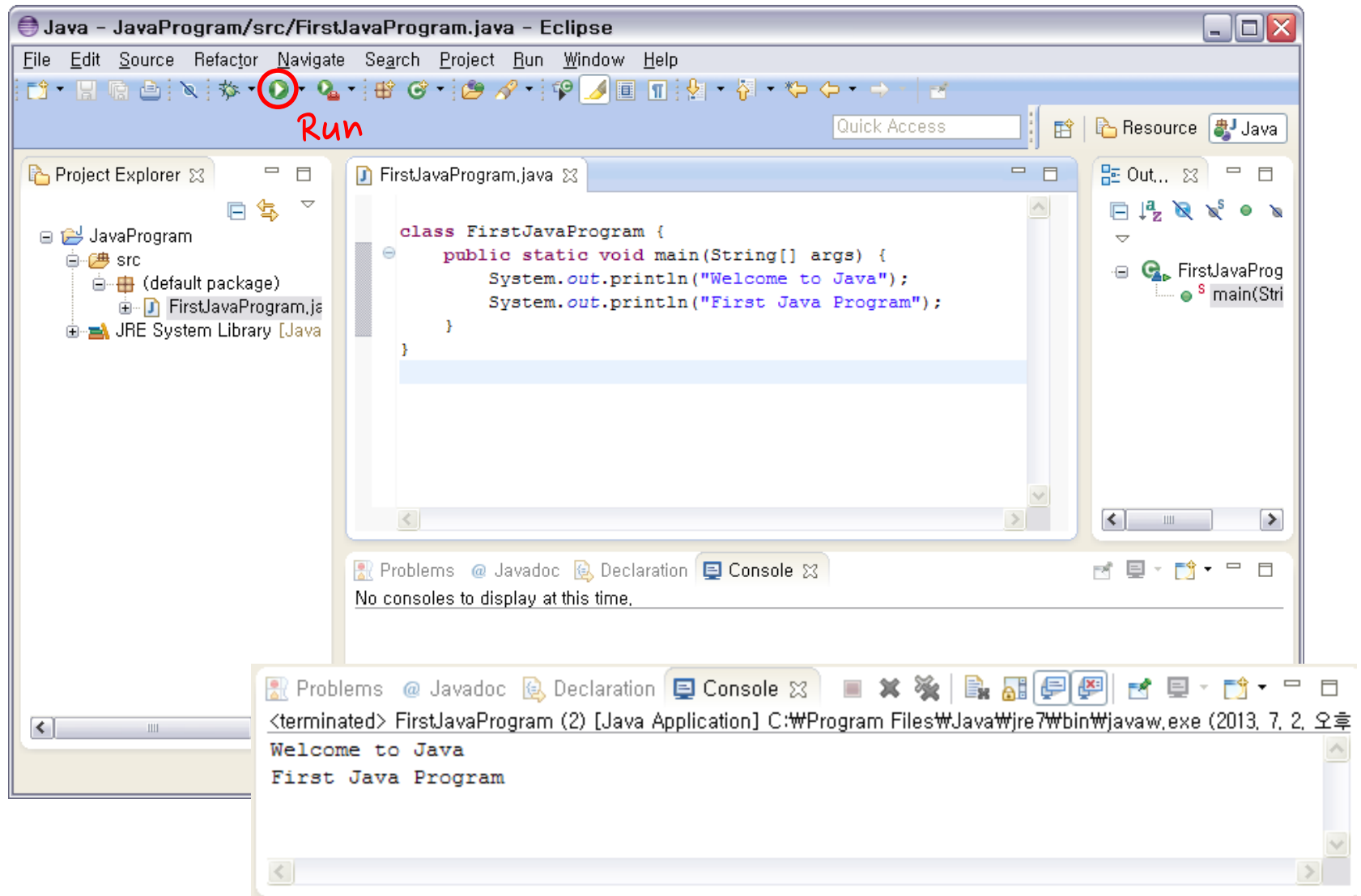
- Project 생성
- Package 생성 (option)
- Java 파일 생성, 편집



개발할 환경 (코드 구성)



개발할 환경



Object oriented Programming

- Procedural programming
 - 일반적으로 프로그램은 sequence of instructions
- Object oriented programming
 - New concept of programming
 - 객체(object)의 입장에서 프로그램을 디자인

object oriented Programming

- 여러 개의 사각형 정보를 관리한다면...

(a)

ID	Top-left		Bottom-right	
	x	y	x	y
1	10	30	30	0
2	30	70	100	60
3	50	50	110	-20
4	-20	10	20	-30
5	60	40	90	0

(b)

ID: 1 x1: 10 y1: 30 x2: 30 y2: 0	ID: 2 x1: 30 y1: 70 x2: 100 y2: 60	ID: 3 x1: 50 y1: 50 x2: 110 y2: -20	ID: 4 x1: -20 y1: 10 x2: 20 y2: -30	ID: 5 x1: 60 y1: 40 x2: 90 y2: 0
--	--	---	---	--

object oriented Programming

- 3번 사각형의 높이를 알고 싶다면...

ID	Top-left		Bottom-right	
	x	y	x	y
1	10	30	30	0
2	30	70	100	60
3	50	50	110	-20
4	-20	10	20	-30
5	60	40	90	0

ID가 3인 행(r)을 알아낸다

getHeight(r) 함수를 수행한다

getHeight(r) 함수는...

r의 3번째 열(top-left y)와 5번째 열(bottom-right y)의 차이를 구한다

계산값을 반환한다

Object Oriented Programming

- 3번 사각형의 높이를 알고 싶다면...

ID: 1 x1: 10 y1: 30 x2: 30 y2: 0	ID: 2 x1: 30 y1: 70 x2: 100 y2: 60	ID: 3 x1: 50 y1: 50 x2: 110 y2: -20	ID: 4 x1: -20 y1: 10 x2: 20 y2: -30	ID: 5 x1: 60 y1: 40 x2: 90 y2: 0
--	--	---	---	--

ID가 3인 객체(o)를 알아낸다

o.getHeight() 메소드를 수행한다

getHeight() 메소드는...

y1과 y2의 차이를 구한다

계산 값을 반환한다

Object Oriented Programming

- 객체(object)는 데이터(data / variable)와 메소드(method)로 구성
 - 멤버 변수 / 멤버 메소드
- 멤버 변수에 저장된 데이터는 해당 객체의 특성을 표현
- 멤버 메소드는 해당 객체의 행위를 표현
- 일반적으로, 멤버 변수는 멤버 메소드로만 접근 가능
 - 정보 은닉(data hiding)
 - OOP의 중요 개념
 - (a)의 getHeight() 함수는 모든 사각형 데이터에 접근 가능
 - (b)의 getHeight() 메소드는 개별 사각형 데이터에만 접근 가능
 - 1번 사각형의 높이는 1번 사각형의 getHeight() 메소드로

class

- Java에서 객체를 만들기 위한 틀
- 일반적으로 하나의 class는 하나의 java 파일로 구성
- class를 잘 만들어놓고 객체를 찍어내라

OOP도 어쨌든 순차적

- 1번부터 3번까지 사각형의 너비, 높이, 면적을 구한다면...

ID 1에서 3까지

ID n 사각형의

너비는

`w = rectangles[n].getWidth();`

높이는

`h = rectangles[n].getHeight();`

면적은

`a = w * h;`

이건 어디에?

main method

main method

- 프로그램이 수행될 순서대로 코드 나열
- class 내부에 작성되나 해당 class의 멤버라고 하지는 않는다
- main method만을 가지는 별도의 class를 만드는 경우가 많다
— ex. Rectangle / RectangleManager

```
class FirstJavaProgram {  
    public static void main(String[] args) {  
        System.out.println("Welcome to Java");  
        System.out.println("First Java Program");  
    }  
}
```

FirstJavaProgram.java

- class로 구성되어 있지만 멤버 변수 / 멤버 메소드 없음
- 프로그램은 main method로부터 시작되어 두 라인을 출력하고 종료

Summary

- Program
 - Sequence of instructions
- Java program
 - 코드(.java) -> 바이트코드(.class) -> JVM 상에서 실행
- Object oriented Programming
 - 객체를 기본 단위로 디자인
- class
 - 객체 생성을 위한 틀
 - 멤버 변수: 생성된 객체의 상태를 담은 장소
 - 멤버 메소드: 생성된 객체가 할 행위 정의
- main method
 - 프로그램의 시작점
 - 프로그램의 동작을 순차적으로 구현