

자료형



- 자바 프로그램 구문
- 식별자 규칙
- 키워드
- 주석
- 자료형
- 정수형 / 문자형 / 실수형 / 논리형
- 형변환

자바 프로그램 구문

● Student.java

```
package com.oopsw.basic;
import java.util.Date;
public class Student
{
    public int count;
    private Date inputDate=new Date();

    public void print()
    {
        System.out.println(++count+ " : " + inputDate);
    }
}
```

} 패키지 명

} 외부 패키지 참조

} 클래스 명

} 멤버 데이터 명

} 멤버 메서드 명

2-2

자바를 기반으로 소프트웨어를 개발하기 위해서는 식별자, 예약어, 주석, 자료형, 연산자

자바 프로그램 구문

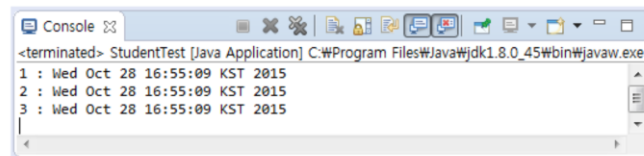
● StudentTest.java

```
package com.oopsw.basic.test;
```

```
import com.oopsw.basic.Student;
```

```
public class StudentTest {  
    public static void main(String[] args) {  
        Student s=new Student();  
        s.print();  
        s.print();  
        s.print();  
    }  
}
```

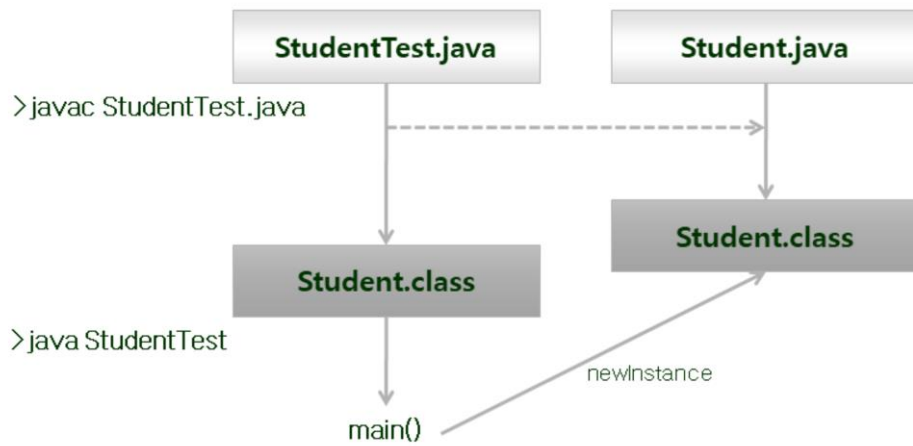
}
main()
JVM 실행



2-3

자바 프로그램 구문

- 프로그램 실행 과정



식별자 규칙

- 식별자(Identifier)
 - 영문 또는 한글과 같은 유니코드로 정의
 - 패키지 명, 클래스 명, 변수 명, 메서드 명 등을 정의
 - 숫자로 시작할 수 없으나 두번째 문자부터는 숫자 사용이 가능
 - `_`, `$` 외에 특수문자로 정의 불가
 - 키워드(예약어)는 식별자 불가
 - 대소문자를 구분
 - 이름의 길이에 제한 없음

식별자 규칙

- 식별자 권장사항
 - 쉽게 이해할 수 있는 단어나 용어를 선택
 - 패키지는 소문자를 사용하고 도메인과 프로젝트 명을 활용
 - 클래스명은 대문자로 시작
 - 멤버 데이터나 메서드는 소문자로 시작
 - 단 사용자 정의 상수일 경우는 모두 대문자로 선언
ex) `Integer.MAX_VALUE`, `Math.PI`
 - 합성어 일경우 연결된 두번째 단어의 첫자는 대문자 표기
ex) `NullPointerException`, `StudentInformation`

키워드

● Java Keywords

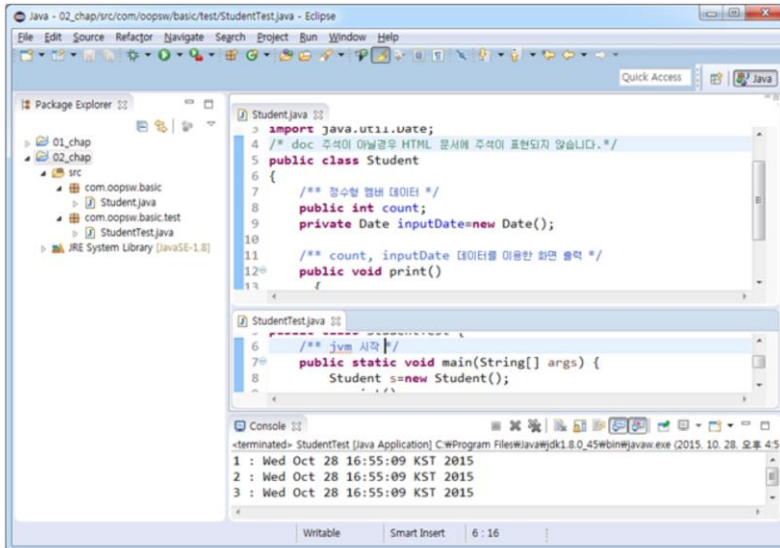
public	private	protected	package	import
class	abstract	interface	this	super
new	extends	implements	for	while
if	else	switch	case	do
boolean	true	false	byte	short
int	long	char	float	double
throw	throws	try	catch	finally
void	return	final	static	default
break	continue	synchronized	transient	instanceof
const	native	strictfp	volatile	assert
goto				

주석(Comments)

- 주석 종류
 - 단일라인 주석
// 소스 중간에 한줄 설명
 - 다중라인 주석
/* 다중
문자열
설명 */
 - 문서화를 위한 주석
/** 이름은 필수로 입력*/
javadoc.exe 컴파일시 html로 된 소스 문서 제공

주석(Comments)

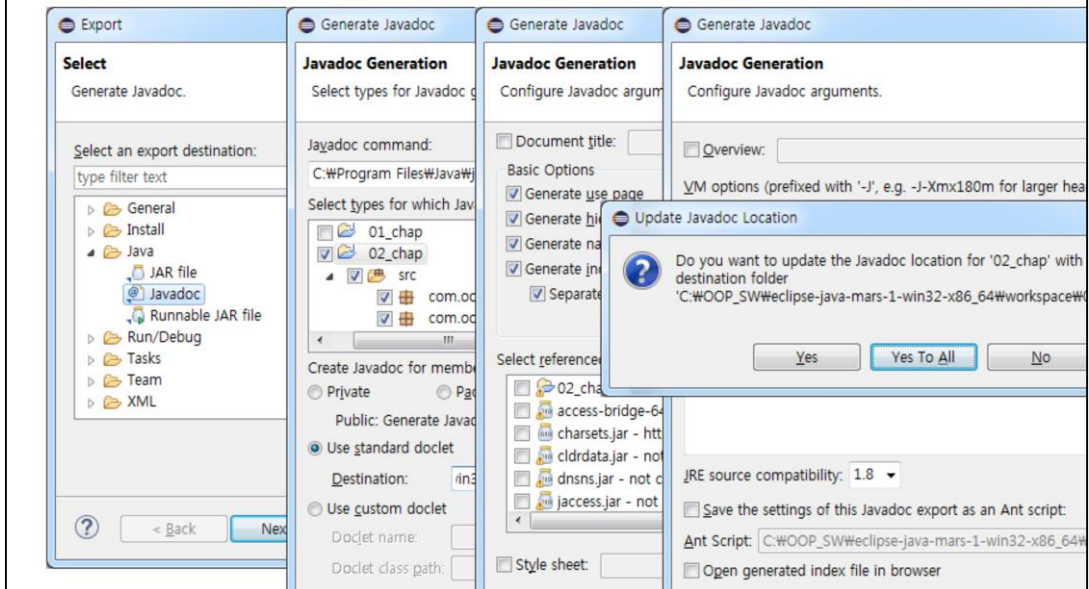
- 주석을 이용한 문서화



2-9

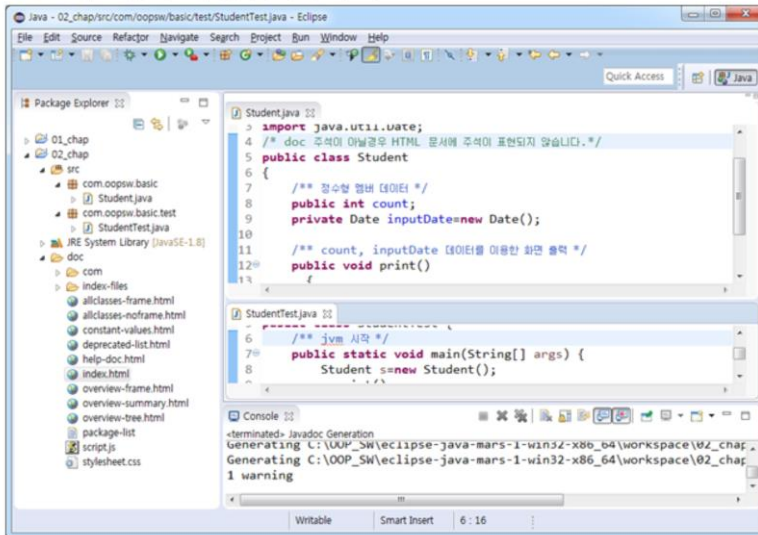
주석(Comments)

● 주석을 이용한 문서화



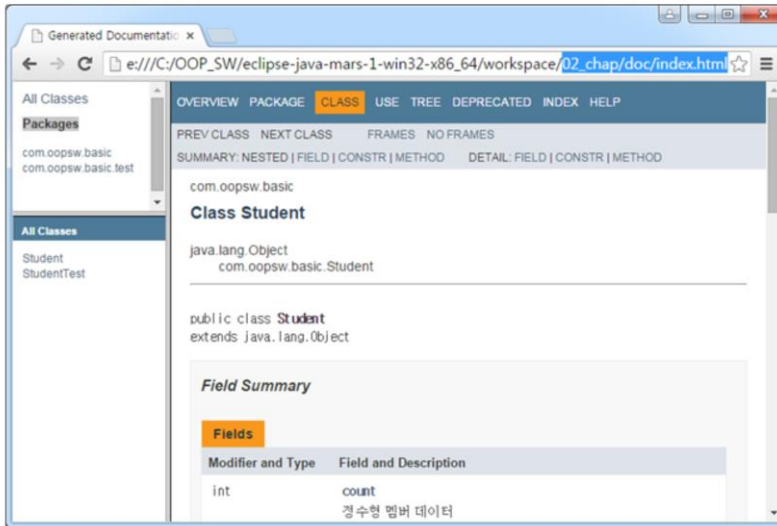
주석(Comments)

● 주석을 이용한 문서화



주석(Comments)

● 주석을 이용한 문서화



2-12

자료형

- 기본형 데이터 타입(Primitive Data Type)
 - JVM에서 키워드로 제공되는 기본 자료형
 - 정수, 실수, 문자, 논리
 - 기본형은 선언, 초기화의 구문을 갖는다.
- 객체형 데이터 타입(Reference Data Type)
 - 클래스로 제공되고 주소로 참조값을 관리하는 객체 자료형
 - 객체형은 선언, 생성, 초기화의 구문을 갖는다.
 - new 키워드를 이용하여 heap 메모리에 객체 할당
 - 객체형 데이터 타입
 - VM이 기본제공 : array ([])
 - API
 - 사용자 정의 객체

자료형

● 진수

사람은 한 자리에 표현할 수 있는 범위 (0 ~ 9) 10 가지수 → 10 진수

0 기계는 한 자리에 표현할 수 있는 범위 (0 ~ 1) 2 가지수 → 2 진수
1

0 0 기계는 두 자리에 표현할 수 있는 범위 (0 ~ 3) 4 가지수
0 1
1 0
1 1

0 0 0 0 0 0 0 0 기계는 8자리에서 표현할 수 있는 수의 범위는 2^8 가지수
.....
1 1 1 1 1 1 1 1

자료형

● 진수

0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---

0 ~ 255 표현 음수를 표현하기 위해서는 부호가 필요
 $0 \sim 2^8 - 1$

.....
 1 1 1 1 1 1 1 1

0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---

양수 0 (+0)

1	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---

음수 0 (-0)

1	1	1	1	1	1	1	1
---	---	---	---	---	---	---	---

음수 최소값 -127

$-2^7 - 1$

0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---

1	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---

+0 and -0

0	1	1	1	1	1	1	1
---	---	---	---	---	---	---	---

양수 최대값 127

$2^7 - 1$

자료형

● 진수

0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---

+0

0	1	1	1	1	1	1	1
---	---	---	---	---	---	---	---

양수 최대값 127

$2^7 - 1$

1	1	1	1	1	1	1	1
---	---	---	---	---	---	---	---

음수 최소값 -128

-2^7

1	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---

-1

1byte

--	--	--	--	--	--	--	--

8bit

--	--	--	--	--	--	--	--

1byt → 8byte → 10진수의 범위 ($-2^{n-1} \sim 2^{n-1} - 1$) → $-2^7 \sim 2^7 - 1$ → -128 ~ 127

(n은 총 비트수)

기본형 데이터 타입 종류

크기(byte)	1	2	4	8
정수형	byte	short	int	long
실수형			float	double
문자형		char (Unicode)		
논리형	boolean (false/true)			

2-17

기본형 데이터 타입 종류

● 정수형

정수	크기		범위	관련객체	기타
	byte	bit			
byte	1	8	$-2^7 \sim 2^7-1$	Byte	
short	2	16	$-2^{15} \sim 2^{15}-1$	Short	
int	4	32	$-2^{31} \sim 2^{31}-1$	Integer	기본
long	8	64	$-2^{63} \sim 2^{63}-1$	Long	10L or 10l

2-18

고정값

정수 기본값 0

정수 기본형 int

byte, short는 기본형보다 작아 연산시 int로 변형

```
byte b1=10, b2=10;
```

```
byte b3=b1+b2; //컴파일 오류
```

long 타입 접미사(l or L)

```
long i=10L
```

기본형 데이터 타입 종류

● 문자

정수	크기		범위	관련객체	기타
	byte	bit			
char	2	16	$0 \sim 2^{16}-1$	Charactor	Unicode

0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

65 'A' ➔ 'Wu0041'

2-19

char

Unicode의 사용(4byte 모두 양수로만 코드값 관리)

' (작은 따옴표)로 문자 표현

'A' //char 형

"A" //String 형임으로 주의

기본값 'Wu0000'

사용법

```
char c1='A'
```

```
char c2=(char)65
```

기본형 데이터 타입 종류

- 실수형

- IEEE(전기전자기술자 협회) 표준 표기법



기본형 데이터 타입 종류

● 실수형

정수	크기		표기법	관련객체	기타
	byte	bit			
float	4	32	123.5F 1.235F * 10 ² 1.235E2 32F	Float	F or f
double	8	64	1234.56678999 1.23456678999 * 10 ³	Double	기본 D or d

2-21

근사값

실수 기본값 0.0

실수 표기법

10.5

1.05*10

1.05E1

실수 기본형 double

double d=10.0;

float f=10.0; //컴파일 오류

float 타입 접미사(f or F)

float f=10.0F

기본형 데이터 타입 종류

- 논리형
 - `boolean`
 - `false`, `true` 키워드로 값 관리
 - 기본값 `false`
 - 사용법
 - `boolean result=true;`
 - `boolean resultOK=num1>num2;`

형변환

- 자동 형변환(Promotion)

- 변수가 자신의 자료형 보다 클 경우 자동 형변환

`int num = 'A'`
↑
자동형 변환

2-23

`long l=10; //10은 int 형`

`double d=10+20;`

`int code='A';`

`float f=20;`

형변환

- 타입 캐스팅(Type Casting)

- 변수가 자신의 자료형 보다 작은 경우 명시적인 타입 캐스팅 필요
- (type)연산자 필요
- 데이터 손실이 있을 수 있다.

`float f = 12.7`
↑
X

`float f = (float)12.7`
↑
타입 캐스팅

2-24

```
byte b1=10, b2=20, b3;  
b3=(byte)(b1+b2);
```

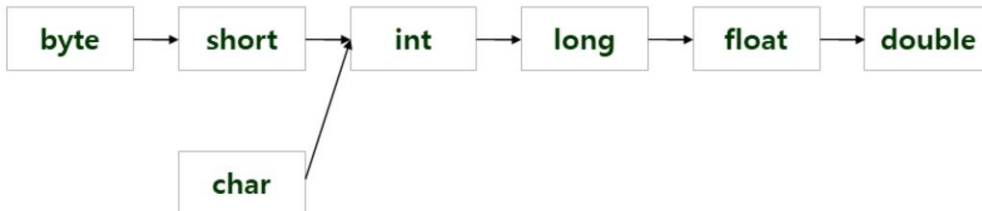
```
float f=(float)12.7 or f=12.7F
```

```
long a=(long)10.0;  
long b=(long)f;
```

```
long c=Integer.MAX_VALUE+1L;
```


자료형의 관계

- 같은 자료형이면 byte 크기에 따라 결정
- 정수형 < 실수형
- short와 char가 같은 2byte라도 음수를 쓰지 않으므로 char는 int형과 연결하여 캐스팅 한다.



LAB