|  |
| --- |
|  |
| Java programming |
|  |
| 멀티미디어 프로그래밍 |

**20131143 최현**

2014 June 2

권오병 교수님

Java programming

멀티미디어 프로그래밍

# 표준 출력문

System.out.println > 출력 후 줄을 바꿈(c언어 printf(“\n”);)

System.out.print > 출력 후 줄을 바꾸지 않음 (c언어 printf(“”);)

# 값을 입력 받을 때

Scanner클래스를 이용 (c언어 scanf)

Import.java.util.Scanner //라이브러리 클래스 포함

Scanner stdIn= new Scanner(System.in) //키보드 입력받는 Scanner 객체 생성

String s = stdIn.next(); //사용자 입력을 문자열로 받음

*\*정수는 nextInt() 실수는 nextDouble()로 받음*

*\*한 줄에 입력되는 서로 다른 데이터는 Space를 이용해 구분함*

# 주석

설명하는 글임

/\* ….\*/ >여러 줄 주석처리

//…. >한 줄 주석처리

/\*\*….\*/ >여러 줄을 주석처리하고 javadoc 도구에 의해 사양서를 만들 수 있음

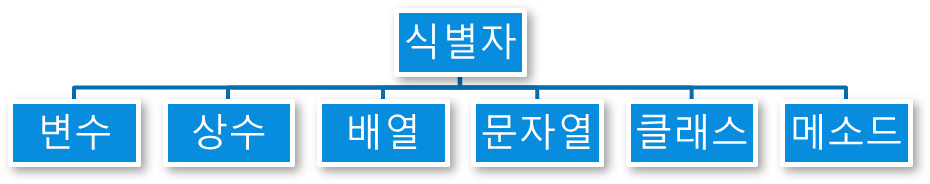
# 프로그래밍 오류

|  |  |
| --- | --- |
| 구문 오류 | 가장 초보적인 오류, 컴파일 할 때 발생(ex.’;기호 생략’,’오타’,’괄호생략’ 등) |
| 실행시간 오류 | 프로그램 실행 시 발생(ex.’숫자를 0으로 나눔’,’데이터 형태틀림’ 등) |
| 논리 오류 | 작성자의 의도와 다른 결과를 나타낼 때 발생, 컴파일러에 의해 오류로 발견되지 않음 |

Java programming

멀티미디어 프로그래밍

# 식별자



### 식별자의 사용원칙

 문자, 숫자 특수문자(\_, $)로 구성 가능

 첫 문자는 문자나 특수 문자로 시작할 수 있으나 숫자는 사용불가

 예약어 사용 불가

 True, false, null 사용불가

 길이제한 없음

 대소문자 구분함

* 클래스는 대문자로 메소드,변수,배열,문자열 등은 소문자로 시작하는 것이 관례
* 상수는 모두 대문자가 관례

# 변수

프로그램에서 필요로 하는 데이터를 저장하기 위한 기억 장소에 주어진 이름

데이터를 보관하는 그릇역할

변수명에는 공백이 포함될 수 없음

값을 배정할 때 ‘=’기호 이용(‘=’는 배정의 의미임, ‘같다(equal)’는 ‘==’이용)

Ex. Sum, count, age

Java programming

멀티미디어 프로그래밍

# 자료형

변수의 타입(형)을 지정해 주는 것

참조 자료형은 변수의 위치에 객체가 저장된 주소가 들어감

Java programming

멀티미디어 프로그래밍

### 정수형

수치 정수형 : byte, short, int, long

Long 사용할 땐 수치 뒤에 L을 붙임

\*자바에서 정수는 묵시적으로 int 형 취급함

\*int 형에 실수 값을 넣으면 오류로 실행 안됨

문자 정수형: char

자바에서 사용하는 문자표현체계는 16비트 유니코드

### 실수형

float형과 double 형이 있음

\*자바에서 실수는 묵시적으로 double 형 취급함

Float 사용할 땐 수치 뒤에 F를 붙임

# 형 변환

연산 할 때 두 개의 피연산자가 다른 형일 경우 자동으로 확대 형 변환을 수행하고 연산함

좁은 자료형 -> 넓은 자료형 일 때만 가능함

Ex double a = 4.0F; -> double이 더 크기 때문에 자동 형 변환 오류 없음

*byte >> short/char >> int >> long >> float >> double*

float a= 3.0; -> 3.0은 묵시적으로 double이고 더 작은 형이 되기 때문에 오류가 생김

실수의 표현범위가 정수보다 넓음(정수 -> 실수 형 변환 가능)

Char -> int는 변환 가능하지만 int -> char는 변환 불가

(char형 자료를 정수로 표현하는 것은 가능,정수를 유니코드 문자로 표현하는 것은 명시적 형변환 필요)

# 연산

정수형에 실수형 값을 대입하는 것은 실행 불가능 이지만 자동적으로 형변환이 일어나 출력은 가능

Ex. Double avg = 9/2 -> 4출력

여러 자료형이 섞인 연산은 큰 자료형으로 자동 형 변환 됨

Java programming

멀티미디어 프로그래밍

# 연산자

+.-와 같이 연산을 나타내는 기호

### 증감연산자

|  |  |  |
| --- | --- | --- |
| 구분 | 증가 | 감소 |
| 전위형 | ++a  하나를 더하고 대입 | - - a  하나를 빼고 대입 |
| 후위형 | a++  대입하고 하나를 더함 | - - a  대입하고 하나를 뺌 |

값이 누적됨

### 관계 연산자

두 값을 비교하여 참, 거짓으로 값을 반환

주로 선택문, 반복문의 조건식에 사용

|  |  |  |  |
| --- | --- | --- | --- |
| x | y | x||y | x&&y |
| 참 | 참 | 참 | 참 |
| 참 | 거짓 | 거짓 | 참 |
| 거짓 | 참 | 참 | 거짓 |
| 거짓 | 거짓 | 거짓 | 거짓 |

두 피연산자의 자료형이 다르면 큰 쪽으로 자동 형 변환시킴

### 논리 연산자

두 값을 평가하여 참, 거짓으로 값을 반환

AND, OR은 이항연산자

NOT은 단항연산자

|  |  |
| --- | --- |
| X | Y |
| 참 | 거짓 |
| 거짓 | 참 |

Java programming

멀티미디어 프로그래밍

# 비트 연산자

### 비트 논리 연산자

비트 단위의 논리 연산자는 2진수인 비트 단위로 연산을 수행하는 연산자로서 피연산자로 정수를 지정

정수를 비트 단위로 출력 하기 위해서 Integer.toBinaryString()메소드 필요

|  |  |  |
| --- | --- | --- |
| 연산자 | 사용법 | 설명 |
| & | 5&7 | 5에 해당하는 101과 7에 해당하는 111을 비트 단위로 AND 함 |
| | | 5|7 | 5에 해당하는 101과 7에 해당하는 111을 비트 단위로 OR 함 |
| ^ | 5^7 | 5에 해당하는 101과 7 에 해당하는 111을 비트 단위로 XOR(exclusive OR) 함 |
| ~ | ~5 | 5에 해당하는 101의 보수를 취함 |

### 음수를 2진수->10진수, 10진수->2진수

Ex) 10110110(2)

1을 0으로 보수화

01001001(2)

2진수를 10진수로 바꿈

1+8+64=73

-부호를 붙임

-73

-1을 함

답 : -74

Ex) -74

절대값을 구함

74

-1을 함

73

10진수 73을 2진수로

01001001(2)

보수화

답 : 10110110(2)

### 비트 논리 연산자 진위표

|  |  |  |
| --- | --- | --- |
| X | Y | X&Y |
| 참 | 참 | 참 |
| 참 | 거짓 | 거짓 |
| 거짓 | 참 | 거짓 |
| 거짓 | 거짓 | 거짓 |

|  |  |  |
| --- | --- | --- |
| X | Y | X|Y |
| 참 | 참 | 참 |
| 참 | 거짓 | 참 |
| 거짓 | 참 | 참 |
| 거짓 | 거짓 | 거짓 |

|  |  |  |
| --- | --- | --- |
| X | Y | X^Y |
| 참 | 참 | 거짓 |
| 참 | 거짓 | 참 |
| 거짓 | 참 | 참 |
| 거짓 | 거짓 | 거짓 |

|  |  |
| --- | --- |
| X | ~X |
| 참 | 거짓 |
| 거짓 | 참 |

### 시프트 연산자

비트 단위로 이동하는 연산자, 정수형 변수에서만 가능하다

|  |  |  |
| --- | --- | --- |
| 연산자 | 사용법 | 설명 |
| << | a<<n | 정수 a를 비트 단위로 왼쪽으로 n비트 시프트함. 비게 된 비트는 0으로 채움  결과는 a\*2^n |
| >> | a>>n | 정수 a를 비트 단위로 오른쪽으로 n비트 시프트함. 비게 된 비트는 0으로 채움  결과는 a/2^n |
| >>> | a<<<n | 정수 a를 비트 단위로 오른쪽으로 n비트 시프트함. 비게 된 비트는 0으로 채움. 부호 비트를 고려하지 않기 떄문에 a가 음수일 경우 시프트 결과는 양수로 나타냄 |

* Ex) 5의 경우

5(101(2))<<2 -> 20(10100(2))

5(101(2))>>2 -> 1(1(2)) =>양수라 부호비트 0 을 채움

5(101(2))>>>2 -> 1(1(2)) =>무조건 0을 채움

* Ex2) -8의 경우

-8을 2진수로 변환하면 11111111111111111111111111111000(2)

-8<<2 -> -32(11111111111111111111111111100000(2))

-8>>2 -> -2(11111111111111111111111111111110(2))

-8>>>2 -> 1073741822(111111111111111111111111111110(2))