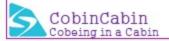
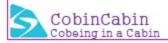
ID:iRaCha

Java Java 표준 입출력



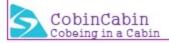
정의

- 스트림(Stream)
 - 운영체제는 키보드, 모니터, 프린트, 파일등을 스트림 이라는 논리적인 동등한 장치로 표현되기 때문에 동일한 방법으로 입출력이 가능
 - 바이트단위로 데이터를전송한다
 - 입력과 출력을 동시에 수행 하려면,입력 스트림과 출력 스트림 2개의 스트림이 필요 a. 단 방향 통신만 가능하기 때문에 하나의 스트림으로 입력과 출력을 동시에 처리 할 수 없다
- System.in : 표준 입력 스트림
 - 키보드
- System.out: 표준 출력 스트림
 - 모니터
- System.err: 표준 에러 스트림
 - 모니터
- 표준 입 . 출력 함수와 버퍼(Buffer)
 - 표준입. 출력 함수는 스트림 내에 버퍼(Buffer)를 사용
 - 버퍼: 입/출력되는데이터가저장되는임시기억장소



출력 함수

- System.out
 - 표준 출력 객체
- println
 - 한 줄 씩 출력 후 줄 바꿈 수행
- print
 - 내용만 출력
- printf
 - 서식문자사용가능



Escape Sequence

- Escape Sequence
 - 키보드에 나타나지 않는 문자나, 화면에 출력되지 않는 제어 문자들은 역 슬러시())를 사용하여 표현할 수 있다.
 - '∖n': New line(새로운 줄로이동)
 - '\t': Tab(탭 크기만큼이동)
 - '\\':\



<u>서식 제어 문자</u>

• 서식 제어 문자

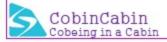
- %d: 정수 입력

- %f: float형 실수 입력

- %c: 문자 입력

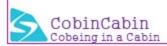
- %s: 문자열 입력





입력 함수

- System.in
 - 표준 입력 객체
- System.in.read()
 - 키보드로부터 입력 받은 값을 byte형태로 저장
- 사용법
 - int num = System.in.read();



입력 함수

- Scanner
 - 다양한 타입의 입력 값들을 읽어 들이기 위한 표준 Java 클래스라이브러리
 - next(): 문자열
 - nextLine() : 한 라인 전체을 읽어 온다
 - nextByte();
 - nextInt();
 - nextLong();
 - nextFloat();
 - nextDouble();
- 사용법
 - Scanner str = new Scanner(System.in);
 - str.next자료형();
 - str.nextLine();

프로그램

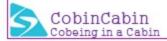
• 학생의 점수를 입력 받아 계산하기

자바 점수 입력: 95 C 언어 점수 입력: 80 이름 입력: 홍길동

[홍길동님의 성적]

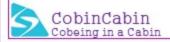
자바: 95 점 C언어: 80 점 합계: 175점 평균: 87.5점





ID:iRaCha

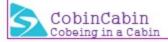
Java Dava 연산자



산술 연산자

<u>산술 연산자</u>

산술 연산자	사용예	의 미
+	a+b	두수의 합
	a-b	두수의 차
*	a*b	두수의 곱
/ 🥯	a/b	나누기 몫
%	a%b	나누기 나머지



산술 연산자

산술 연산자

- 크기가 4byte이하인 자료형을 int형으로 변환한다
 - byte, char, short => int
- 자료형의 표현범위가 큰 쪽에 맞춰서 형 변환된 후 연산 수행
 - int + float =>float+ float
- 정수연산의 결과는 정수
 - byte + byte \Rightarrow int + int = int
 - byte + short => int + int = int
 - char + char => int + int = int
- 정수와 실수연산의 결과는 실수
 - float+int => float+float = float
 - long + float =>float + float = float
 - float + double => double + double = double

비교연산자

비교 연산자

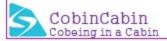
• 두 개의 피 연산자 간의 대소관계를 비교하기 위하여 사용

연산자	의미	사용 예
	보다 작다	if(a<10)~
> /	보다 크다	if(a>10)~
<=	보다 작거나 같다	if(a<=10)~
>=	보다 크거나 같다	if(a>=10)~
==	와 같다	if(a==10)~
!=	와 같지 않다	if(a!=10)~

비교 연산자

비교 연산자

- 대소비교연산자
 - <,>,<=,>=
 - 기본형 중에서 boolean형을 제외한 나머지 자료형에 사용할 수 있다
 - 참조형에는 사용할 수 없다
- 등가 비교 연산자
 - ==,!=
 - 기본형,참조형 모든 자료형에 사용 할 수 있다
 - 참조형: 두 개의 피 연산자가 같은 객체를 가리키고 있는지를 알 수 있다
 - 기본형과 참조형 간에는 서로 형 변환을 허용하지 않는다.
 - String만 예외적으로 문자 결합에 '+'연산을 수행할 수 있다
 - 문자열 비교에는 equals()를 사용 할 수 있다



대입 연산자

대입 연산자

- 우측에 수행한 결과를 좌측에 지정된 변수로 대입
- 복합 대입 연산자
 - 대입연산자를 다른 연산자와 결합하여 사용

복합 대입 연산자	사용예	의미
+=	a+=b	a=a+b
-= 0	a-=b	a=a-b
=	a=b	a=a*b
/=	a/=b	a=a/b
%=	a%=b	a=a%b

논리 연산자

논리 연산자

- 참과 거짓을 판별하는 연산
 - boolean형 또는 boolean형 값을 결과로 하는 조건식만을 허용
 - 비교가 참인 경우는 결과는 1되고, 거짓인 경우는 0

논리 연산자	의미	사용 예
	논리합(OR)	if(a<=0 a>=65535) ~
&&	논리곱(AND)	if(b>=60 && b<70) ~
!	부정(NOT)	if(!(c%2==1))~

A	В	∥(OR, +, 합집합)	&&(AND, *, 교집합)
true	true	true	true
true	false	true	false
false	true	true	false
false	false	false	false

논리 연산자

논리 연산자

- or연산
 - 피 연산자중 어느 한 쪽만 true이면 true를 결과로 얻는다.
 - 왼쪽의 피연산자가 true이면 오른쪽의 피연산자의 값을 검사하지 않는다
- and연산
 - 피연산자 모드 true이어야 true를 결과로 얻는다
 - 왼쪽의 피연산자가 false이면 오른쪽의 피연산자의 값을 검사하지 않는다.



증감연산자

증감연산자

• 피연산자를 1씩 증가 혹은 감소하는 기능

증감연산자	의미	사용 예
++	피연산자의 값을 1만큼 증가	++a 또는a++
(+ (피연산자의 값을 1만큼 감소	a 또는a

• 전치와 후치에 따른 연산자 비교

- 전치: ++a로 표기하며 a=a+1을 먼저 처리한다

- 후치: a++로표기하며 a의 데이터를 사용한 후 a=a+1을 처리한다

조건 연산자

조건 연산자

- 조건식에 따라 참인 경우 앞부분의 식을 거짓인 경우 뒷부분을 수행
- 사용예
 - (조건식)?참인 경우:거짓인 경우;
 - 조건식이 참인 경우 ?뒤의 내용을 거짓일 경우 : 뒤의 내용을 실행

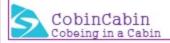


비트 연산자

비트 연산자

- 10진수를 2진수로 변환하여 각 비트별로 논리/이동 연산을 한다.
- 실수형을 제외한 모든 기본형에 사용 가능하다.

비트연산자	의미	
	비트 단위 논리합(OR)	
&	비트 단위 논리곱(AND)	
^	비트 단위 배타적 논리합(XOR)	
~	비트 부정(NOT)	
<<	비트 좌측 이동(Left Shift)	
>>	비트 우측 이동(Right Shift)	



연산자

연산자 우선순위

• 연산자 우선순위

연산자	연산순서	우선순위	비고
(), [], ->, .(점)	좌에서 우		
sizeof, (type), &, *, -(단항), +(단항),, ++, ~,!	좌에서 우		단항
*(곱셈), / , %, +, /	좌에서 우		산술
<<,>>>	좌에서 우		비트
<, <=, >, >=, ==, !=	좌에서 우		비교
8, ^,	좌에서 우		비트
&&,	좌에서 우		논리
?:	우에서 좌		삼항
%=, /=, *=, -=, +=, =	좌에서 우		대입
	좌에서 우		콤마