숫자 연산에서 정수와 정수 연산의 결과 ->정수

숫자 연산에서 정수와 실수 연산의 결과 ->실수

타입 불일치 : 대입할 값과 변수의 타입이 서로 다른 경우, 문제가 발생.

타입 변환 : 작은 타입의 데이터 는 큰 데이터 타입으로 자동 변환됨ex) 정수는 실수로 자동 변환됨.

(캐스팅) : 소괄호 사이에 해석 타입을 명시하여 해석하게 하는것.

-다운 캐스팅 : 넓은 범위에서 좁은 범위로 해석

-업 캐스팅 : 좁은 범위에서 넓은 범위로 해석

타입.pasre타입(문자); = 문자열을 정수로 변환하는 메소드

타입.toString(숫자) = 정수를 문자열로 변환하는 메소드

<입력>

Import java.util.Scanner;

-프로그램 첫 행에 다음을 추가해 Scanner 클래스의 경로 이름을 컴파일러에 알림.

Scanner in = new Scanner(System.in);

-키보드로 데이터를 입력 받기 위해 System.in 객체와 연결된 Scanner 객체를 생성.

Int x = in.nextint();

-Scanner 클래스가 제공하는 다양한 메서드를 이용해 키보드로 데이터를 입력 받음.

메소드 : 코드 묶음을 입력 값을 사용하여 수행 할 수 있음.

호출부 -> 정의부

<메소드 구조>

Public static TYPE NAME PARMAETERS {

return VALUE;

}

<메소드 호출>

메소드명(입력값1, 입력값2…)제어문

-프로그램의 흐름에 영향을 주고 때에 따라 제어가 가능하도록 하는 것.

-> 즉 실행문의 수행 순서를 변경

-제어문의 종류

-> 조건문, 반복문, 분기문 등

조건문

* 상황별 실행 흐름을 만드는 문법
* 조건식을 만족하는 경우와 그렇지 못한 경우를 나누어 실행을 달리하는 기법

조건문의 종류

1. If 문
2. If else 문
3. 다중 if 문
4. 중첩 if 문
5. Switch 문
6. 삼항 연산자

If 문

If 문은 조건에 따른 수행 여부를 결정한다.

If else 문

If else 문은 두 조건식 중 하나를 선택할 때 사용

조건식이 참인 경우 if 문의 코드가 거짓인 경우 else문의 코드가 수행

다중 IF 문

추가 조건을 붙여야 할 때 사용

-조건문이 수행은 윗줄에서부터 차례대로 진행

-다중 IF문을 활용하면 여러 상황 중 하나를 선택할 수 있음

조건문

-소괄호 사이에 기준 값 GRADE를 두고,

-그 상태를 A, B, C, DEFAULT의 경우로 분리

Switch문

-콜론 레이블을 사용하는 기존switch문

-기존 switch문은 낙하 방식으로 콜론 case 레이블 이용

삼항 연산자

-삼항 연산자는 조건식에 따른 반환값을 지정하는 문법.

-조건 식이 참이면 A, 거짓이라면 B가 반환

반복문

-반복문은 규칙적 반복 코드를 단순화 하는 문법

While 문과 for문으로 나누어진다.

While 문 : 반복 횟수가 명확하지 않을 때 사용하기 적합하다.

while문은 반복 횟수가 조건에 따라 변할 때 사용하기 좋은 반복문이다.

For 문 : 반복 횟수가 명확할 때 사용하기 적합

Do while 문 : 조건식이 거짓이라도 한번은 본체를 실행한다.

Break 문 : 반복문을 탈출하는 문법

Continue 문 : 실행 흐름을 다음 반복으로 넘기는 문법

메소드 오버로딩

메소드 이름은 같지만 메소드 시그니처가 다른 메서드를 정의하는 것.