<제어문>

1.조건문

1.1 if

1.2 switch

2.반복문

2.1 for

2.2 while

2.3 do-while

3.분기문

3.1 break

3.2 continue

1. 조건문

어떤 조건에 참인지 거짓인지에 따라 결과가 달라져 나오는 명령문이다.

조건문 안에서 if와 switch로 나눌수 있다.

if문에서는 {if, else if, else}로 구성되어 있고, switch문에서는 {switch, case, default, break}로 구성되어 있다.

1.1 if

*1.1.1 if*

if(조건식){

System.out.printIn(“조건식이 참일 경우 실행")

}

: 조건식 결과에 따라서 명령문 실행 여부가 결정되는데, 조건식에는 참 또는 거짓의 값을 산출할 수 있는 연산식이나 불린 자료형이 올 수 있다.

*1.1.2 else*

if(조건식){

System.out.printIn(“조건식이 참일 경우 실행")

}else{

“조건식이 거짓일 경우 조건식의 false인 이곳에서 실행”

}

: if문의 조건식이 참이면 if문의 명령문이 실행되고, 조건식이 거짓이면 명령문의 거짓인 else 구문에서 실행되며 전체 if문에 벗어난다.

*1.1.3 else if*

if(조건식1){

조건식1이 참일 경우 이곳에서 실행된다.

}else if(조건식2){

조건식2이 참일 경우 이곳에서 실행된다.

}else if(조건식3){

조건식3이 참일 경우 이곳에서 실행된다.

}else if(조건식4){

조건식4이 참일 경우 이곳에서 실행된다.

}

: 조건식이 하나가 아닌 여러 개인 if문에서는 if문 블록 끝에 else if문을 붙인다. else if문의 갯수 제한은 없고, 여러 조건식 중 참인 조건식에만 실행되고 전체 if문을 벗어난다. 마지막에는 else문을 추가하는데 else문은 위에 if-else문 내용과 동일하게 모든 조건식이 거짓인 경우에 실행되고 전체 if문을 벗어나게 된다.

1.2 switch

switch(변수){

case 1:

실행 코드;

break;

case 2:

실행 코드;

break;

case 3:

실행 코드;

break;

case 4:

실행 코드;

break;

default:

실행 코드;

break;

}

: switch문은 if-else if문과 유사하지만, 변수의 값에 따라 코드를 실행할 수 있게 제어할 수 있다. if문은 조건식이 true, false밖에 없어 경우의 수가 많아질 경우, 코드가 복잡해질 수 있지만 switch문은 변수의 값에 따라 실행문을 결정하기 때문에 if문보다 코드가 간결하고 가독성이 좋은 장점이 있다.

변숫값에 동일한 case가 없으면 default로 가서 실행된다. 그리고, default는 생략 가능하다.

case끝에 break;가 있는 이유는 다음 case를 실행하지말고 switch문을 빠져나가기 위함이므로 만약, break가 없다면 다음 case가 연달아 실행되는데 이때는 case값과는 상관없이 실행된다.

2. 반복문

반복문은 주어진 조건이 충족될 때까지 반복적으로 문장을 수행시키는 명령문이다. 프로그램을 처리하는 부분에 있어서 반복적인 형태가 많으므로, 가장 많이 사용되는 제어문 중 하나다.

2.1 for

for(초기화식; 조건식; 증감식){

실행문;

}

: 반복문의 대표라고 할 수 있다. 초기화식이 가장 먼저 실행되고, 조건식이 참이면 실행문을 실행시킨다. 만일, 조건식이 거짓이 나오면 for문은 실행하지 않고 끝나게 된다. 초기화식은 조건식, 실행문, 증감식에서 사용할 변수를 초기화하는 역할을 한다. 실행문이 모두 실행되면 증감식을 실행시키고 다시 조건식으로 가서 참이 나오면 다시 실행문을 실행시킨다. 이 순으로 계속 실행문이 반복적으로 실행되고 조건문이 거짓이 되면 for문은 종료한다.

2.2 while

while(조건식){

실행문;

}

: for문이 반복되는 횟수가 정해져 있다면, while문은 for문과 다르게 조건식이 참일 경우에 계속해서 반복하는 반복문이다. 조건식이 거짓이 나오면 반복 실행을 멈추고 while문을 종료한다.

2.3 do-while

do{

실행문;

}while(조건식);

: while문은 조건식이 참이지 않으면 아예 실행이 되지 않지만, do-while문은 조건이 충족되지 않아도 무조건 한번은 실행되는 점에 있어서 차이가 있다.

3. 분기문

분기문은 조건문과 반복문에서 중간에서 주어진 조건의 흐름을 바꿀 수 있는 구문이다.

3.1 break

for(조건식1){

실행문;

if(조건식2){

break;

}

}

: break문은 for문, while문 등 반복문을 실행 중지할 때 사용한다.

예를 들어, 조건식1에 충족되는 for문이 반복 실행 중에 있어서 조건식2에 충족되어 break문이 사용되면 for문은 종료된다.

3.2 continue

for(조건식1){

실행문;

if(조건식2){

continue;

}

}

: continue문은 반복문에만 사용되는데, 해당 반복을 건너뛰는 역할을 한다. 즉, 그 다음 반복은 진행된다. 반복을 건너뛸 뿐 반복문이 종료되는 것은 아니다.

예를 들어, 조건식1에 충족되는 for문이 반복 실행 중에 있어서 조건식2에 충족되어 continue문이 사용되면 조건식2에 맞는 실행문만 실행되지 않는다. 그리고 그 뒤로는 계속 조건식1에 맞게 반복 진행된다.