

4. 제어문(Control Flow Statements)

조건문 if

- 조건문으로서 조건식이 참이면 블록 안의 문장들이 수행됨
 - if와 else로 구문이 구성되며, 조건식은 일반적으로 논리/비교 연산식을 사용

[사용형식 1]

```
if (조건식){  
    실행문;  
}
```

[사용형식 2]

```
if (조건식){  
    실행문;  
} else{  
    실행문;  
}
```

[사용형식 3]

```
if (조건식){  
    실행문;  
} else if(조건식){  
    실행문;  
} else{  
    실행문;  
}
```

실습(1)

[Ex1] IfControl.java

```
package chap04.control;

public class IfControl {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        int score = 97;

        if(score >= 90) {
            System.out.println("A학점 입니다.");
        }else if(score >= 80) {
            System.out.println("B학점 입니다.");
        }else if(score >= 70) {
            System.out.println("C학점 입니다.");
        }else if(score >= 60) {
            System.out.println("D학점 입니다.");
        }else {
            System.out.println("F학점 입니다.");
        }
    }
}
```

실습(2)

[Ex2] IfDice.java

```
package chap04.control;

public class ifDice {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        int num = (int) (Math.random() * 6) + 1;

        if(num==1) {
            System.out.println("1번이 나왔습니다.");
        } else if(num==2) {
            System.out.println("2번이 나왔습니다.");
        } else if(num==3) {
            System.out.println("3번이 나왔습니다.");
        } else if(num==4) {
            System.out.println("4번이 나왔습니다.");
        } else if(num==5) {
            System.out.println("5번이 나왔습니다.");
        } else {
            System.out.println("6번이 나왔습니다.");
        }
    }
}
```

조건문 switch

- if문과 같은 조건문으로 if문 보다 간결한 구현 가능

```
switch(조건식){  
    case 값1:  
        실행문;  
        break;  
    case 값2:  
        실행문;  
        break;  
    case 값3:  
        실행문;  
        break;  
    ..... // case의 개수 제한 없음  
    default: // 조건식의 결과와 일치하는 case 문이 없을 경우 default의 실행문이 수행됨  
        실행문;  
}
```

실습(1)

[Ex3] SwitchDice.java

```
package chap04.control;

public class SwitchDice {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        int num = (int) (Math.random() * 6) + 1;

        switch(num) {
            case 1:
                System.out.println("1번이 나왔습니다.");
                break;
            case 2:
                System.out.println("2번이 나왔습니다.");
                break;
            case 3:
                System.out.println("3번이 나왔습니다.");
                break;
            case 4:
                System.out.println("4번이 나왔습니다.");
                break;
            case 5:
                System.out.println("5번이 나왔습니다.");
                break;
            default:
                System.out.println("6번이 나왔습니다.");
                break;
        }
    }
}
```

실습(2)

[Ex4] SwitchSeason.java

```
package chap04.control;

import java.util.Scanner;

public class SwitchSeason {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        int iMonth = 0;
        System.out.print("월을 입력하세요. (1월 ~ 12월)>");

        Scanner scanner = new Scanner(System.in);
        String sMonth = scanner.nextLine(); // 월을 입력받은
        iMonth = Integer.parseInt(sMonth); // 입력받은 월을 int 타입으로 변환

        switch(iMonth) {
            case 3: case 4: case 5:
                System.out.printf("%d월은 봄입니다.", iMonth);
                break;
            case 6: case 7: case 8:
                System.out.printf("%d월은 여름입니다.", iMonth);
                break;
            case 9: case 10: case 11:
                System.out.printf("%d월은 가을입니다.", iMonth);
                break;
            default: // case 12: case 1: case 2:
                System.out.printf("%d월은 겨울입니다.", iMonth);
        }
    }
}
```

반복문 for

- 초기화식, 조건식, 증감식으로 구성된 반복제어 명령으로서 조건식을 만족하는 상황에서 블록 안의 코드를 반복적으로 실행하게 하는 제어문

```
for (초기화식; 조건식; 증감식) {  
    실행문;  
}
```

[Ex5] ForLoop.java

```
package chap04.control;  
  
public class ForLoop {  
    public static void main(String[] args) {  
        // TODO Auto-generated method stub  
        for (int i = 0; i < 10; i++) {  
            System.out.print(i);  
            if(i != 9)  
                System.out.print(", ");  
        }  
    }  
}
```


반복문 while

- 조건을 만족하는 상황에서 블록 안의 코드를 반복적으로 실행하게 하는 제어문

[일반 형식]

```
while (조건식) {  
    실행문;  
}
```

[for(초기화식; 조건식; 증감식)과 동일]

```
초기화식;  
while (조건식) {  
    실행문;  
    증감식;  
}
```

[Ex6] WhileLoop.java

```
package chap04.control;  
  
public class WhileLoop {  
  
    public static void main(String[] args) {  
        // TODO Auto-generated method stub  
        int i = 0;  
        while(i < 10) {  
            System.out.print(i);  
            if(i != 9)  
                System.out.print(", ");  
            i++;  
        }  
    }  
}
```

반복문 do-while

- while과 조건식을 만족해야 하는 형식은 비슷하나 조건식을 체크하기 전에 우선 실행을 한다는 차이가 있음 (그런 이유로 while문과 달리 최소한 한번의 수행은 보장)

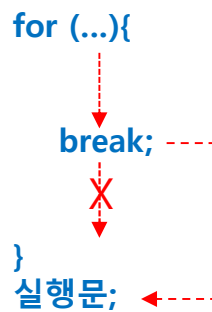
```
do {  
    실행문;  
} while(조건식);
```

[Ex7] DoWhileLoop.java

```
package chap04.control;  
  
public class DoWhileLoop {  
  
    public static void main(String[] args) {  
        // TODO Auto-generated method stub  
        boolean b = false;  
  
        System.out.println("[While]");  
        while(b) {  
            System.out.println("b는 " + b + " 입니다.");  
        }  
  
        System.out.println("[Do-While]");  
        do{  
            System.out.println("b는 " + b + " 입니다.");  
        }while(b);  
    }  
}
```

break

- 조건문 및 반복문의 블록에서의 실행을 중지하고 블록을 빠져 나오도록 하는 명령



[반복문에서의 사용]

```
while (조건식) {
    실행문1;
    break;
    실행문2; // 실행 안 됨
}
실행문3; // break로 빠져나온 다음 실행
```

[조건문에서의 사용]

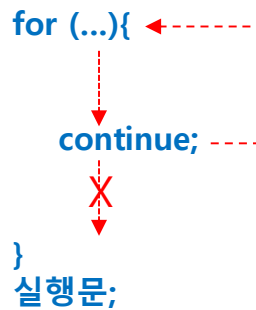
```
switch(조건식){
    .....
    case 값:
        실행문;
        break; // switch 블록을 빠져나감
    .....
    default:
        실행문;
}
```

[Ex8] BreakEx.java

```
package chap04.control;
public class BreakEx {
    public static void main(String[] args) {
        int i = 0;
        while(true) {
            System.out.print(i);
            if(i != 9)
                System.out.print(", ");
            else
                break;
            i++;
        }
    }
}
```

continue

- 반복문의 블록 안에서 실행을 중지하고 다시 반복문의 조건식 부분으로 돌아감



[사용 형식]

```
while (조건식) {  
    실행문1;  
    continue; // 아래 실행문을 건너 뛰고 다시 조건식으로 돌아감  
    실행문2; // 실행 안 됨  
}
```

[Ex9] ContinueEx.java

```
package chap04.control;  
  
public class ContinueEx {  
    public static void main(String[] args) {  
        int i = 1;  
        for(i = 1; i < 20; i++) {  
            if(i%2 == 0)  
                continue;  
            System.out.print(i + " ");  
        }  
    }  
}
```

실습(1)

[Ex10] Gugudan.java

```
package chap04.control;

public class Gugudan {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        for(int i = 2; i <= 9; i++) {
            for(int j = 1; j <= 9; j++) {
                System.out.printf("%d x %d = %d\n", i, j, i*j);
            }
            System.out.println("=====");
        }
    }
}
```

실습(2)

[Ex11] StarFor.java

```
package chap04.control;

import java.util.Scanner;

public class StarFor {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        int num = 0;

        System.out.print("*을 출력할 라인의 수를 입력하세요.>");

        Scanner scanner = new Scanner(System.in);
        String tmp = scanner.nextLine();
        num = Integer.parseInt(tmp);

        for(int i=0;i<num;i++) {
            for(int j=0;j<=i;j++) {
                System.out.print("*");
            }
            System.out.println();
        }
    }
}
```

실습(3)

[Ex12] Multiple.java

```
package chap04.control;

import java.util.Scanner;

public class Multiple {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        System.out.println("1부터 100까지 구하고 싶은 배수를 입력하세요.");
        Scanner scanner = new Scanner(System.in);
        int num = Integer.parseInt(scanner.nextLine());
        int i = 1;

        System.out.printf("[%d의 배수]\n", num);
        while(i <= 100) {
            if(i%num == 0)
                System.out.println(i);
            i++;
        }
    }
}
```

실습(4)

[Ex13] Matrix.java

```
package chap04.control;

public class Matrix {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        int i = 0;
        while(i < 10) {
            int j = 0;
            while(j < 10) {
                System.out.printf("[%d, %d]", i, j);
                j++;
            }
            System.out.println();
            i++;
        }
    }
}
```


실습(5)

[Ex14] Factorial.java

```
package chap04.control;

import java.util.Scanner;

public class Factorial {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        System.out.println("숫자를 입력하세요.");
        Scanner sc = new Scanner(System.in);
        int num = sc.nextInt();

        int i = 1;
        int result = 1;
        do {
            result *= i;
            i++;
        } while (i <= num);

        System.out.printf("%d! 의 결과는 %d 입니다.", num, result);
    }
}
```