제어문

콘솔에서 입력받기

❖콘솔에서 문자열 입력을 받을 때 1.5 이전 버전에서는 java.io.BufferedReader, java.io.InputStreamReader 2개의 클래스와 System.in 객체를 이용

❖System.in 은 표준 입력장치를 의미하는데 일반적으로 키보드를 의미

❖InputStreamReader는 키보드와 프로그램을 연결시켜주는 역할을 수행

❖BufferedReader는 키보드의 입력을 저장하는 버퍼의 역할을 수행

❖BufferedReader 객체가 readLine()을 호출하면 줄 단위로 읽어서 리턴

```
BufferedReader in = new BufferedReader(new InputStreamReader(System.in));
String str = in.readLine();
```

❖ 입력받은 내용을 정수나 실수로 변경

Integer.parseInt(문자열), Double.parseDouble(문자열)

실습(ConsoleInput.java)

```
이름을 입력하세요: 아이린
나이를 입력하세요: 30
이름:아이린나이:30
```

```
import java.io.BufferedReader;
import java.io.InputStreamReader;

class ConsoleInput {
	public static void main(String[] args) {
		String name = "noname";
		int age = 0;
		BufferedReader in = new BufferedReader(new InputStreamReader(System.in));
		try {
			System.out.print("이름을 입력하세요: ");
			name = in.readLine();
			System.out.print("나이를 입력하세요: ");
			age = Integer.parseInt(in.readLine());
		} catch (Exception e) {
			System.out.println("입력 오류:"+ e.getMessage());
		}
		System.out.println("이름:" + name + "₩t나이:" + age);
	}
}
```

콘솔에서 입력받기

- Scanner 클래스
  - ✓ java.util.Scanner 클래스
  - ✓ Scanner 객체 생성

```
Scanner a = new Scanner(System.in);
```

  - ✓ Scanner에서 키 입력 받기
    - Scanner는 입력되는 키 값을 공백으로 구분되는 아이템 단위로 읽음
    - 공백 문자 :'₩t',' ₩f',' ₩r',' ',' ₩n'

```
Scanner scanner = new Scanner(System.in);
String name = scanner.next();                  //문자열
String addr = scanner.next();                  // 문자열
int age = scanner.nextInt();                   // 정수
double weight = scanner.nextDouble();          // 실수
```

콘솔에서 입력받기

생성자	설명
Scanner(File Source)	파일로부터 입력
Scanner(InputStream Source)	네트워크나 키보드(System.in)로부터 입력
Scanner(Readable Source)	다음 아이템을 찾아 byte로 변환하여 반환
Scanner(String Source)	문자열로부터 입력

메소드	설명
String next()	다음 아이템을 찾아 문자열로 반환
boolean nextBoolean()	다음 아이템을 찾아 boolean으로 변환하여 반환
byte nextByte()	다음 아이템을 찾아 byte로 변환하여 반환
double nextDouble()	다음 아이템을 찾아 double로 변환하여 반환
float nextFloat()	다음 아이템을 찾아 float로 변환하여 반환
int nextInt()	다음 아이템을 찾아 int로 변환하여 반환
long nextLong()	다음 아이템을 찾아 long으로 변환하여 반환
short nextShort()	다음 아이템을 찾아 short로 변환하여 반환
String nextLine()	한 라인 전체('₩n' 포함)를 문자열 타입으로 반환
boolean hasNextLine()	입력된 라인이 있는지 여부를 리턴

ScannerInput.java

```
import java.util.Scanner;

public class ScannerInput
{
	public static void main(String[] args) {
		System.out.print("임의의 숫자를 입력하세요: ");
		//키보드로부터 입력받는 Scanner 객체를 생성
		Scanner scanner = new Scanner(System.in);
		//구분 문자를 누를 때까지의 내용을 정수로 변환해서 n에 대입
		//실수면 nextFloat 이나 nextDouble 문자열이면 next 로 변환
		int n = scanner.nextInt();
		System.out.println("입력한 정수:" + n);
		scanner.close();
	}
}
```

```
임의의 숫자를 입력하세요: 30
입력한 정수:30
```

VariousInput.java

```
이름을 입력하세요: 박수영
안녕하세요 박수영님
나이를 입력하세요: 25
박수영님의 나이는 25세 입니다.
```

```
import java.util.Scanner;

public class VariousInput {
	public static void main(String[] args) {
		Scanner keyboard = new Scanner(System.in);

		System.out.print("이름을 입력하세요: ");
		String name = keyboard.nextLine();
		System.out.println("안녕하세요 " + name + '님');

		System.out.print("나이를 입력하세요: ");
		int age = keyboard.nextInt();
		System.out.println(name + "님의 나이는 " + age + "세 입니다.");

		keyboard.close();
	}
}
```

콘솔 입력의 문제점

```
이름을 입력하세요: 예리
나이를 입력하세요: 22
직업을 입력하세요: 예리님의 나이는 22세이며 직업은 입니다.
```

```
import java.util.Scanner;

public class InputProblem {
	public static void main(String[] args) {
		Scanner keyboard = new Scanner(System.in);

		System.out.print("이름을 입력하세요: ");
		String name = keyboard.nextLine();
		System.out.print("나이를 입력하세요: ");
		int age = keyboard.nextInt();

		//구분문자를 없애기 위해서 문자열 입력 메소드를 한 번 호출
		//keyboard.nextLine();

		//숫자 다음에 문자열을 입력받아서 변환하고자 하면 Enter 와 같은 구분 문자를 받음
		System.out.print("직업을 입력하세요: ");
		String job = keyboard.nextLine();

		System.out.println(name + "님의 나이는 " + age + "세이며 직업은 " + job +"입니다.");

		keyboard.close();
	}
}
```

1. 제어문(ControlStatement)

❖자바 인터프리터는 기본적으로 왼쪽에서 오른쪽으로 위에서 아래로 프로그램을 읽어서 해석

❖제어문은 프로그램의 흐름을 변경하는 명령어

❖제어구조: 프로그램의 흐름

- ✓ 순차 처리: 위에서 아래로
- ✓ 선택 처리 – 분기(Branch)
  - 조건 또는 값에 의한 분기: if, if~else, switch 등
  - 무조건 분기: goto(자바에서는 사용할 수 없는 예약어)
- ✓ 반복(Loop): for, while, do~while
- ✓ 기타: break, continue, return…

❖ 분기나 반복에서 분기에 따른 처리 내용이나 반복 조건에 따른 처리 내용은 {처리 내용} 처럼 작성하는데 처리 내용이 한 줄 이라면 { }를 생략 할 수 있음 – 코드의 가독성을 높이기 위해서 { }를 생략하지 않는 것을 권장

2. 선택 처리 - 조건 분기

❖단일 if문

✓ 형식

```
if (조건) {
        문장;
}
        문장2;
```

✓ 조건의 결과가 true이면 { }안의 문장을 수행한 후 { } 바깥쪽 문장을 처리하고 조건이 거짓이면 { }안의 문장을 수행하지 않고 { } 바깥쪽 문장을 처리

✓ 조건은 반드시 ( )에 입력해야 하고 조건문 다음에 처리해야 할 문장이 여러 줄이라면 반드시 { }를 해야 하고 한 줄이면 { }를 생략가능

✓ 조건은 boolean 타입으로 결과가 리턴되는 연산식, 변수, 메소드 호출, 상수 모두 가능

실습(TestIf.java)

```
import java.util.Scanner;

public class TestIf {
	public static void main(String args[]) {
		int jumsu = 0;
		Scanner input = new Scanner(System.in);
		System.out.print("점수를 입력하세요:");
		jumsu = input.nextInt();
		if (jumsu >= 60) {
			System.out.println("당신은 합격입니다");
		}
		System.out.println("당신의 점수는" + jumsu + "입니다");
		input.close();
	}
}
```

```
점수를 입력하세요:67
당신은 합격입니다
당신의 점수는67입니다
```

```
점수를 입력하세요:53
당신의 점수는53입니다
```

2. 선택 처리 - 조건 분기

❖if ~ else문

✓ 형식

```
if(조건){
        처리내용1;
}else{
        처리내용2;
}
```

✓ ( )의 결과가 true이면 처리내용1을 수행하고 false이면 처리내용2을 수행

✓ if는 단독으로 사용이 가능하지만 else는 단독 사용이 안됨

실습(TestIfElse.java)

```
import java.util.Scanner;

public class TestIfElse {
	public static void main(String args[]) {
		int jumsu = 0;
		Scanner input = new Scanner(System.in);
		System.out.print("점수를 입력하세요:");
		jumsu = input.nextInt();
		if (jumsu >= 60) {
			System.out.println("당신은 합격입니다");
		} else {
			System.out.println("당신은 불합격입니다");
		}
		System.out.println("당신의 점수는" + jumsu + "입니다");
		input.close();
	}
}
```

```
점수 입력:97
합격
당신의 점수는97입니다
```

```
점수 입력:59
불합격
당신의 점수는59입니다
```

실습(LoginTest1.java)

```
import java.util.Scanner;

public class LoginTest1 {
	public static void main(String[] args) {
		String id = null;
		String password = null;
		Scanner input = new Scanner(System.in);
		System.out.print("아이디를 입력하세요: ");
		id = input.next();
		System.out.print("비밀번호를 입력하세요: ");
		password = input.next();

		if (id.equals("root") && password.equals("system")) {
			System.out.println("로그인 성공");
		} else {
			System.out.println("로그인 실패");
		}
		input.close();
	}
}
```

2. 선택 처리 - 조건 분기

❖다중 if ~ else if문

```
if (조건1)
        { 처리내용1; }
else if (조건2)
        { 처리내용2; }
else
        { 처리내용3; }
```

❖조건1이 true이면 처리내용1을 수행하고 제어문을 빠져나가고 그렇지 않으면 조건2를 검사해서 true이면 처리내용2를 처리하고 빠져나가며 false 이면 else 블럭에 있는 처리내용3을 처리

❖else if는 개수에 상관없이 작성 가능

❖else는 생략 가능

❖if 문이 중첩되는 경우 else if와 else는 가장 가까운 if와 쌍으로 연결

실습(ElseIfTest.java)

```
import java.util.Scanner;

public class TestIfElseif {
	public static void main(String args[]) {
		int jumsu = 0;
		Scanner input = new Scanner(System.in);
		System.out.print("점수를 입력하세요:");
		jumsu = input.nextInt();
		if (jumsu >= 90 && jumsu <= 100) {
			System.out.println("당신은 학점은 A입니다.");
		} else if (jumsu >= 80 && jumsu < 90) {
			System.out.println("당신은 학점은 B입니다.");
		} else if (jumsu >= 70 && jumsu < 80) {
			System.out.println("당신은 학점은 C입니다.");
		} else if (jumsu >= 60 && jumsu < 70) {
			System.out.println("당신은 학점은 D입니다.");
		} else if (jumsu >= 0 && jumsu < 60) {
			System.out.println("당신은 학점은 F입니다.");
		} else {
			System.out.println("잘못된 점수입니다.");
		}
		input.close();
	}
}
```

```
점수를 입력하세요:88
당신은 학점은 B입니다.

점수를 입력하세요:120
잘못된 점수입니다.
```

실습(LoginTest2.java)

```
import java.util.Scanner;
public class LoginTest2 {
	public static void main(String[] args) {
		String id = null;
		String password = null;
		Scanner input = new Scanner(System.in);
		System.out.print("아이디를 입력하세요: ");
		id = input.next();
		System.out.print("비밀번호를 입력하세요: ");
		password = input.next();

		if (id.equals("root") && password.equals("system")) {
			System.out.println("로그인 성공");
		} else if (id.equals("root")) {
			System.out.println("비밀번호가 틀렸습니다.");
		} else {
			System.out.println("없는 아이디 입니다.");
		}
		input.close();
	}
}
```

```
아이디를 입력하세요: root
비밀번호를 입력하세요: system
로그인 성공

아이디를 입력하세요: root
비밀번호를 입력하세요: 1234
비밀번호가 틀렸습니다.

아이디를 입력하세요: ggangpae1
비밀번호를 입력하세요: system
없는 아이디 입니다.
```

2. 선택 처리 – 값에 의한 분기

❖switch-case

✓ 형식

```
switch(수식 또는 변수 또는 메소드의 리턴 값)
{
    case 값1:
        처리내용1;
        break;
    case 값2:
        처리내용2;
        break;
    default:
        처리내용;
        break;
}
```

2. 선택 처리 – 값에 의한 분기

❖switch-case

- ✓ 규칙
  - case의 값 – 한 개의 값 만 사용
  - 처리 내용이 여러 개 – 중괄호({ }) 사용 안 함
  - break - switch 블록을 빠져 나오게 되며 break가 없으면 다음 case의 처리도 수행 : break를 만날때까지 모두 처리
  - default 문 – case문에서 일치하는 값이 없을 경우 수행
  - case에 사용하는 값은 정수 또는 문자열(1.7에서 추가) 데이터만 가능

실습(TestSwitch1)

```
import java.util.Scanner;

import java.util.Scanner;

public class TestSwitch1 {

	public static void main(String[] args) {
		// 키보드로부터 입력받기 위한 객체를 생성
		Scanner sc = new Scanner(System.in);
		// 메뉴 입력받기
		System.out.print("메뉴 입력(1-한식 2-중식 3-분식 기타-일식):");
		int menu = sc.nextInt();

		//읽기 전용의 변수를 생성
		final int KOREA = 1;
		final int CHINA = 2;
		final int SNACKBAR = 3;
```

```
메뉴 입력(1-한식 2-중식 3-분식 기타-일식):1
한식

메뉴 입력(1-한식 2-중식 3-분식 기타-일식):3
분식

메뉴 입력(1-한식 2-중식 3-분식 기타-일식):7
일식
```

실습(TestSwitch1)

```
//menu의 값에 따른 분기
		switch(menu) {
		case KOREA:
				System.out.println("한식");
				break;
		case CHINA:
				System.out.println("중식");
				break;
		case SNACKBAR:
				System.out.println("분식");
				break;
		default:
				System.out.println("일식");
				break;
		}

		sc.close();
	}
}
```

실습(TestSwitch2)

```
import java.util.Scanner;

class SwitchTest2 {
	public static void main(String args[]) {
		int jumsu = 0;
		Scanner input = new Scanner(System.in);

		System.out.print("점수를 입력하세요(0-100): ");
		jumsu = input.nextInt();
		System.out.println("점수:" + jumsu);


		char grade;
```

```
점수를 입력하세요(0-100): 87
점수:87
입력된 점수 87점은  B학점 입니다.

점수를 입력하세요(0-100): 176
점수:176
잘못된 점수 입력입니다
```

실습(TestSwitch2)

```
switch (jumsu / 10) {
case 10:
case 9:
		grade = 'A';
		System.out.println("입력된 점수 " + jumsu + "점은  " + grade + "학점 입니
다.");
		break;
case 8:
		grade = 'B';
		System.out.println("입력된 점수 " + jumsu + "점은  " + grade + "학점 입니
다.");
		break;
case 7:
		grade = 'C';
		System.out.println("입력된 점수 " + jumsu + "점은  " + grade + "학점 입니
다.");
		break;
```

실습(TestSwitch2)

```
        case 6:
                grade = 'D';
                System.out.println("입력된 점수 " + jumsu + "점은  " + grade + "학점 입니
다.");
                break;
        case 5:
        case 4:
        case 3:
        case 2:
        case 1:
        case 0:
                grade = 'F';
                System.out.println("입력된 점수 " + jumsu + "점은  " + grade + "학점 입니
다.");
                break;
        default:
                System.out.println("잘못된 점수 입력입니다");
        }
```

실습(TestSwitch2)

```
            input.close();
        }
    }
```

3. 반복(Loop)

❖while 문

✓ 형식

```
while(조건식)
{
   처리 내용;
}
```

✓ 조건식이 true일 경우: block({ })내의 처리 내용을 실행하고 조건으로 이동

✓ 조건식이 false일 경우: block({ })내의 처리 내용을 실행하지 않고 빠져 나옴.

✓ 조건식은 boolean이 나오는 식

실습(TestWhile)

```
public class TestWhile {

	public static void main(String[] args) {
		int idx=0;
		//while(idx < 10)
		while(idx != 10){
			idx = idx + 1;
			System.out.println("idx=" + idx);
		}
	}
}
```

```
idx=1
idx=2
idx=3
idx=4
idx=5
idx=6
idx=7
idx=8
idx=9
idx=10
```

3. 반복(Loop)

❖do while 문

✓ 형식

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

✓ 적어도 한번은 무조건 block({ }) 내의 문장을 실행
✓ 조건식이 true일 경우 block({ }) 내의 처음으로 이동해서 문장을 수행
✓ 조건식이 false일 경우 block 바깥으로 이동
✓ while(조건식) 뒤에 반드시 세미콜론(;)을 입력

실습(TestDoWhile)

```
public class TestDoWhile {

	public static void main(String[] args) {
		int idx=10;

		do {
			idx = idx - 1;
			System.out.println("idx=" + idx);
		}while(idx > 0);
	}
}
```

```
idx=9
idx=8
idx=7
idx=6
idx=5
idx=4
idx=3
idx=2
idx=1
idx=0
```

3. 반복(Loop)

❖**for 문**

✓ 형식

```
for (초기식; 조건식; 두 번째 반복부터 수행되는 식)
{
    문장;
    문장;
}
```

✓ 초기식- 맨 처음 한번만 수행하는 식으로 수행하고 조건식으로 이동

✓ 조건식의 결과가 false가 되면 for 블록을 빠져 나가게 되고 그렇지 않으면 괄호 안의 내용을 수행

✓ 세번째 식은 한 번 수행이 끝나고 이후부터 반복적으로 수행하는 식

✓ 모든 식은 생략 가능하고 하나의 식에 2개 이상의 명령문을 입력해도 되는데 이 때는 , 로 구분해서 입력

실습(TestFor)

```
public class TestFor {

	public static void main(String[] args) {
		for(int idx=1;idx<=5;idx++)
		{
			System.out.println("pageNo=" + idx);
		}
	}
}
```

```
pageNo=1
pageNo=2
pageNo=3
pageNo=4
pageNo=5
```

실습(TestSum.java)

```
public class TestSum {
	public static void main(String[] args) {
		//1부터 10까지의 합 구하기
		int i;
		int sum = 0;
		for(i=1; i<= 10; i=i+1){
			sum = sum + i;
		}
		System.out.println("합계:" + sum);

		sum = 0;
		for(i=1; i<= 10; sum=sum+i, i=i+1);
		System.out.println("합계:" + sum);
	}
}
```

```
합계:55
합계:55
```

실습(TestFloatSum)

```
public class ModPattern {
	public static void main(String[] args) {
		int i = 0;
		for(;;) {
			try {
				i = i % 4;
				if(i==0)
					System.out.println("라투");
				else if(i==1)
					System.out.println("오미크론");
				else if(i==2)
					System.out.println("다크스펙터");
				else
					System.out.println("오미크론");
				Thread.sleep(1000);
				i++;
			} catch (InterruptedException e) {
				e.printStackTrace();
			}
		}
	}
}
```

```
라투
오미크론
다크스펙터
오미크론
라투
오미크론
다크스펙터
오미크론
라투
오미크론
```

실습(TestLoop)

```
public class TestLoop {

	public static void main(String[] args) {
		//while은
		//한번만 수행할 내용 -> while(조건) -> { }내에 반복수행할 내용
		int i = 0;
		while(i<3){
				System.out.println("Hello Java");
				i = i + 1;
		}
		System.out.println("=======================");
		//do~while은
		//한번만 수행할 내용 -> do{반복 수행할 내용} -> while(조건);
		i = 0;
		do {
				System.out.println("Hello Java");
				i = i + 1;
		}while(i<3);
		System.out.println("=======================");
```

```
Hello Java
Hello Java
Hello Java
==================
Hello Java
Hello Java
Hello Java
==================
Hello Java
Hello Java
Hello Java
```

실습(TestLoop)

```
		//for
		//for(한번만 수행할 내용; 조건; 두번째부터 수행할 내용){반복 수행할 내용}
		for(i=0; i<3; i=i+1) {
			System.out.println("Hello Java");
		}
	}
}
```

3. 반복(Loop)

❖무한 반복 만들기

- ✓ while(true){ }
- ✓ do{ }while(true)
- ✓ for(; ;){ }

4. 기타 제어문

❖제어문은 중첩해서 사용 가능

❖제어문이 중첩되면 바깥쪽 제어문에서 시작해서 안쪽의 제어문을 전부 수행하고 바깥쪽의 제어문의 마지막 부분으로 가서 작업을 다시 수행하거나 종료

❖break: switch 문이나 반복문에서 탈출하는 제어문

❖continue: 반복문에서 이하 문장을 수행하지 않고 다음 반복으로 건너 뛰는 제어문으로 for 문인 경우는 증감식으로 이동하고 while 문이면 맨 위 조건 비교식으로 이동하며 do – while인 경우는 {의 시작 부분으로 이동

❖break와 continue는 대부분 반복문에서 if와 함께 사용

❖반복문의 경우 라벨을 설정한 후 break나 continue 뒤에 라벨 이름을 입력해서 특정 라벨의 반복문을 중지시키거나 반복을 계속 수행할 수 있도록 할 수 있음

실습(TestBreakContinue)

```
public class TestBreakContinue {
	public static void main(String args[]) {
		int sum = 0;
		for (int i = 1; i <= 10; i++) {
			//첫번째 3의 배수를 만나면 반복문 종료
			if (i % 3 == 0) {
				break;
			}
			sum = sum + i;
		}
		System.out.println("Break Sum의 합:" + sum);

		sum = 0;
		for (int i = 1; i <= 10; i++) {
			//3의 배수인 경우는 아래 문장을 수행하지 않고 반복문의 최상단으로 이동
			if (i % 3 == 0) {
				continue;
			}
			sum = sum + i;
		}
		System.out.println("Continue Sum의 합:" + sum);
	}
}
```

```
Break Sum의 합:3
Continue Sum의 합:37
```

실습(Gugudan)

[2]단
2∗1=2 2∗2=4 2∗3=6 2∗4=8 2∗5=10 2∗6=12 2∗7=14 2∗8=16 2∗9=18
[3]단
3∗1=3 3∗2=6 3∗3=9 3∗4=12 3∗5=15 3∗6=18 3∗7=21 3∗8=24 3∗9=27
[4]단
4∗1=4 4∗2=8 4∗3=12 4∗4=16 4∗5=20 4∗6=24 4∗7=28 4∗8=32 4∗9=36
[5]단
5∗1=5 5∗2=10 5∗3=15 5∗4=20 5∗5=25 5∗6=30 5∗7=35 5∗8=40 5∗9=45
[6]단
6∗1=6 6∗2=12 6∗3=18 6∗4=24 6∗5=30 6∗6=36 6∗7=42 6∗8=48 6∗9=54
[7]단
7∗1=7 7∗2=14 7∗3=21 7∗4=28 7∗5=35 7∗6=42 7∗7=49 7∗8=56 7∗9=63
[8]단
8∗1=8 8∗2=16 8∗3=24 8∗4=32 8∗5=40 8∗6=48 8∗7=56 8∗8=64 8∗9=72
[9]단
9∗1=9 9∗2=18 9∗3=27 9∗4=36 9∗5=45 9∗6=54 9∗7=63 9∗8=72 9∗9=81

실습(Gugudan)

```
public class Gugudan
{
	public static void main(String args[])
	{
		for(int dan=2; dan<10; dan++)
		{
			System.out.println("[" + dan + "]단");
			for(int su=1; su<10; su++)
			{
				System.out.print(dan + "*" + su + "=" + (dan*su) + " ");
			}
			System.out.println();
		}
	}
}
```

실습(BreakOuter)

```
public class BreakOuter {
	public static void main(String[] args) {
		outLoop: for (int i = 0; i < 3; i++) {
			for (int j = 0; j < 3; j++) {
				System.out.println("i:" + i + "-j:" + j);
				if (i == 1 && j == 2)
			//break는 기본적으로 가까운 반복문을 빠져나가지만 break 다음에 레이블을
			//사용하면 레이블에 해당하는 반복문을 빠져 나갑니다.
					break outLoop;
			}
		}
	}
}
```

```
i:0-j:0
i:0-j:1
i:0-j:2
i:1-j:0
i:1-j:1
i:1-j:2
```

실습(ContinueOuter)

```
public class ContinueOuter {
	public static void main(String[] args) {
		outLoop: for (int i = 0; i < 3; i++) {
			for (int j = 0; j < 5; j++) {
				System.out.println("i:" + i + "-j:" + j);
				if (j == 2) {
					continue outLoop;
				}
			}
		}
	}
}
```

```
i:0-j:0
i:0-j:1
i:0-j:2
i:1-j:0
i:1-j:1
i:1-j:2
i:2-j:0
i:2-j:1
i:2-j:2
```

실습(Star)

```
public class Star {
	public static void main(String args[]) {
		for (int i = 1; i <= 5; i++) {
			for (int j = 1; j <= 5; j++) {
				System.out.print("*");
			}
			System.out.println();
		}
	}
}
```

```
*****
*****
*****
*****
*****
```

실습(Switching)

```
public class Switching {
	public static void main(String args[]) {
		boolean flag = false;
		for (int i = 1; i <= 10; i++) {
			if (flag == false) {
				System.out.println("On");
			} else {
				System.out.println("Off");
			}
			flag = !flag;
		}
	}
}
```

```
On
Off
On
Off
On
Off
On
Off
On
Off
```

연습문제

❖ 아래와 같이 출력하는 프로그램을 작성해보세요.

```
*
**
***
****
*****
```

```
*
***
*****
*******
*********
```

```
*****
****
***
**
*
```

```
*
**
***
**
*
```

```
  *
 ***
*****
 ***
  *
```

연습문제

❖ 1부터 9까지의 정수를 입력 받아서 입력된 단의 구구단을 출력해보세요

1부터 9까지의 임의의 숫자를 입력하세요: 3

3 * 1 = 3
3 * 2 = 6
3 * 3 = 9
3 * 4 = 12
3 * 5 = 15
3 * 6 = 18
3 * 7 = 21
3 * 8 = 24
3 * 9 = 27

연습문제

❖ 입력된 년도가 윤년인지 아닌 지 판별해주는 프로그램을 작성해보세요
  ✓ 윤년은 4의 배수이고 100배수는 아니거나 400의 배수이면 윤년

년도를 입력하세요: 2012　　년도를 입력하세요: 2017

2012년은 윤년입니다.　　2017년은 윤년이 아닙니다.

❖ 2보다 큰 양의 정수를 입력 받아서 소수 인지 판별하는 프로그램을 작성하시오.
  ✓ 소수는 1과 자기 자신으로만 나누어 떨어지는 숫자

❖ 2부터 1000까지 에서 완전수의 개수를 구하시오
  ✓ 완전수는 자신을 제외한 약수의 합이 자신과 동일한 수
  ✓ 6 -> 1,2,3,6이 약수인데 6을 제외한 수의 합의 6이므로 완전수
  ✓ 7 -> 1,7이 약수인데 7을 제외한 수의 합이 1이므로 완전수가 아님