01.First JAVA

```
public class First_Java {
    신규 *
    public static void main(String[] args) {
        System.out.println("First JAVA");
        System.out.println("추가된 메세지입니다.");
    }
}
```

가장 맨처음으로 만들었던 클래스다.
public static void main(String[] args){} 자바 실행시 가장 먼저 동작하고 사용되는 함수

System.out.println(); 는 출력문으로 ()안에 “First JAVA”를 출력시켜준다.

02.For문

```
public class ForTest {
    신규 *
    public static void main(String[] args) {
        for(int i = 0; i < 10; i++){
            System.out.println(i);
        }
    }
}
```

For문의 가장 간단한 식을 만들어 봤다.

for(int i = 0; i < 10; i++){} = for(초기화식; 조건식; 증감식){} 이런순으로 들어간다.
이를 풀어보면

int i = 0; = ‘ int(상수) **i**는 **0**이다.’
i < 10; = **‘0 < 10’**
i++ = ‘ **i**를 **1**증가 시키고, 다시 조건문으로 가서 거짓이 될 때까지 반복한다.’

i를 **‘**System.out.println(i);**’** 로 출력해보면 0~9까지의 숫자를 출력해준다.

03.배열 For문

```
public class ArrayTest {
    신규 *
    public static void main(String[] args) {
        final int START = 0;
        int[] numberArray = {1,2,3,4,5};
        for (int i = START; i < numberArray.length; i++){
            System.out.println("출력 값: " + numberArray[i]);
        }
        for (int num : numberArray) {
            System.out.println("출력: " + num);
        }
    }
}
```

배열For문이다.
기본적으로 For문 과 동일하나 'int[] numberArray = {1,2,3,4,5};'이 들어간다.

int[] numberArray = {1,2,3,4,5}; ='int numberArray의 []안에 {1,2,3,4,5}을 넣는다'.

그아래 For문은 for (int i = START; i < numberArray.length; i++)을 풀어보면

int i = START 는 int i = 0이고,
i < numberArray.length 는 numberArray이 {1,2,3,4,5}를 갖고
.length는 배열의 크기를 갖게 해주는 속성으로
numberArray의 {1,2,3,4,5}에게 (1~5) 배열을 줘서
조건식을 바꿀필요없이 만들어준다.
i++는 For문과 동일

이 식을 System.out.println("출력 값: " + numberArray[i]);으로 넣으면
출력 값: 1
출력 값: 2
출력 값: 3
출력 값: 4
출력 값: 5
로 출력된다.

아래 For문은 for (int num : numberArray)
int num : numberArray 은 num = numberArray으로 numberArray 을 줄이기 위해서 사용한것같다.
출력값은 위에 식과 동일하다.

04.Heap과Stack +주석

```java
public class HeapStackTest {
    신규 *
    public static void main(String[] args) {
        final int Start = 0;
        final int ALOC_ARRAY_NUMBER = 5;

        int[] numberArray = new int[ALOC_ARRAY_NUMBER];
        for (int i = Start; i < ALOC_ARRAY_NUMBER; i++) {
  /*     int i = Start; = int i의 초기값 = Start = 0 이다.
         ㄴ즉 i는 Start 값인데 그 Start 값은 0이기에 i는 0의 값을 갖는다.

       i < ALOC_ARRAY_NUMBER; = int i < ALOC_ARRAY_NUMBER = 5
       이므로 0~4까지가 나온다.

       i++  = 증감 즉 i는 0~4까지 1씩 늘어난다.
*/
            numberArray [i] = i+1;
            System.out.println("1." + numberArray[i]);
            //System.out.println() = 출력하고 한칸 내림

            System.out.print("2." + numberArray[i]);
            System.out.println("");
            //System.out.print() = 그 줄에 출력함(덧붙여지는 느낌)

            System.out.printf("3.%d\n",numberArray[i]);
            //System.out.printf() = format 함수를 사용할 수 있다.
            //ㄴ%d = (int), %s = (String), %f = (float,double)

            System.out.printf("3.numberArray[%d] = %d\n", i,numberArray[i]);
            /*System.out.printf()이므로 format 을 사용했고
             ()안의 "numberArray[%d] = %d\n", i,numberArray[i]는 나눠서 보면된다
            1. "numberArray[%d] = %d\n" || 2. i,numberArray[i]이고
             앞에 %d = i, %d = numberArray[i]를 대입하면된다.*/
            for (int num : numberArray) {
                System.out.println("numberArray elem " + num );
            }
        }
    }
}
```

Stack 는정적으로 할당된 메모리 영역 Ex) Primitive타입 (boolean, char, int…)
Heap는 동적으로 할당된 메모리 영역 Ex) Stack에 사용되는 것을 제외한것

05.While문

```java
public class WhileTest {
    신규 *
    public static void main(String[] args) {
        int idx = 0;
        char ch = 'A';

        while (idx < 10) {
            System.out.println("idx: " + idx + ", 안녕: " + (char)(ch + idx));
            idx++;
        }
    }
}
```

While문이다.
For문과 비슷하며

For문: 반복횟수가 정해진경우, 배열과 함께 주로 사용됨
While문: 무한루프나 특정조건에 만족할때까지 반복해야할경우 사용됨

위의 식을보면
while (idx < 10) {System.out.println("idx: " + idx + ", 안녕: " + (char)(ch + idx));
idx++;
으로
while (idx < 10) 는 (0<10)이고,
("idx: " + idx + ", 안녕: " + (char)(ch + idx) 는 idx 에 0, (char)(ch + idx) A + 0을 char로 만드므로 “idx” + 0, “안녕” + A(char 0)이라고 볼 수 있다.
idx++; 는 증감식이기에 실행값은

idx: 0, 안녕: A
idx: 1, 안녕: B
idx: 2, 안녕: C
idx: 3, 안녕: D
idx: 4, 안녕: E
idx: 5, 안녕: F
idx: 6, 안녕: G
idx: 7, 안녕: H
idx: 8, 안녕: I
idx: 9, 안녕: J

로 나온다.

06.Boolean

```
final class Led{
    3개 사용 위치
    private boolean isTurnOn;
    1개 사용 위치  신규 *
    public Led() {
        this.isTurnOn = false;
        System.out.println("생성자 호출");
    }
    2개 사용 위치  신규 *
    public boolean isTurnOn() {
        System.out.println("Getter");
        return isTurnOn;
    }
    1개 사용 위치  신규 *
    public void isTurnOn(boolean turnOn) {
        System.out.println("Setter");
        isTurnOn = turnOn;
    }
}
0개의 사용위치  신규 *
public class LectureClassTest {
    신규 *
    public static void main(String[] args) {
        System.out.println("생성자 호출전");
        Led led = new Led();
        System.out.println("현재 전구 상태: " + (led.isTurnOn()? "켜짐" : "꺼짐"));
        led.isTurnOn(true);
        System.out.println("생성자 호출후");
        System.out.println("현재 전구 상태: " + (led.isTurnOn()? "켜짐" : "꺼짐"));
    }
}
```

boolean은 가볍게 말해 참, 거짓을 판단하는 함수이며 위에 식과 같이 사용된다.
final class Led는 새로운 Led 클래스를 만든것이다.

private boolean isTurnOn; = isTurnOn의 참,거짓 함수를 만든것이고

public Led() {this.isTurnOn = false; System.out.println("생성자 호출")} = “생성자”

public boolean isTurnOn() {System.out.println("Getter"); return isTurnOn;} = “Getter”

public void isTurnOn(boolean turnOn) {
System.out.println("Setter"); isTurnOn = turnOn;} = “Setter”

Led led = new Led(); 는 다른 곳의 클래스를 불러오는 식이다.
ClassName 원하는이름 = new ClassName();
나머지는 System.out.println(); 대로 행동하며 (led.isTurnOn()? "켜짐" : "꺼짐") =
led.isTurnOn의 상태가 true인지 false인지 알려준다.

07.If문

```
public class IfTest {
    신규 *
    public static void main(String[] args) {
        final int PERMIT_AGE = 18;
        final int inputAge = 19;
        final int PERMIT_KIDS = 13;

        if (PERMIT_AGE < inputAge) {
            System.out.println("입장 가능합니다.");
        }else {
        System.out.println("입장 불가능합니다.");
        }

        System.out.println("");

        if (PERMIT_AGE < inputAge) {
            System.out.println("성인용입니다.");
        }
        if (PERMIT_KIDS >= inputAge) {
            System.out.println("아동입니다.");
        }
    }
}
```

If문이다.
if (PERMIT_AGE < inputAge){System.out.println("입장 가능합니다.");}
= '만약 18 < 19 면 입장 가능합니다.'
else {System.out.println("입장 불가능합니다.");}
= '그렇지않다면 입장 불가능합니다'
if문은 true혹은 false로 값이 나오며 연산식이나 boolean변수가 올 수 있다.

08.switch문

```java
import java.util.Scanner;

0개의 사용위치  신규 *
public class SwitchTest {
    신규 *
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        boolean isLoop = true;

        while (isLoop) {

            System.out.println("숫자를 입력하세요: ");
            int inputNumber = sc.nextInt();

            switch (inputNumber) {
                case 0:
                    System.out.println("종료");
                    isLoop = false;
                    break;
                case 1:
                    System.out.println("입금!");
                    break;
                case 2:
                    System.out.println("출금!");
                    break;
                case 3:
                    System.out.println("조회!");
                    break;
                default:
                    System.out.println("그런명령은 존재하지 않습니다.");
                    break;
            }
        }
    }
}
```

Switch문
Scanner함수는 출력문에 입력할 수 있게 해준다.

Scanner sc = new Scanner(System.in); = Scanner를 sc로 만들어준다.

boolean isLoop = true; = boolean함수를 사용하고 isLoop 는 true다.

while (isLoop) = 만약 참이면

System.out.println("숫자를 입력하세요: ");

int inputNumber = sc.nextInt(); ‘ inputNumber는 숫자값을 입력받습니다.’
switch (inputNumber) = switch (숫자값을 입력받습니다.)

case 0: , System.out.println("종료"); , isLoop = false; , break;
 = 숫자값:0, 종료 , 만약 거짓이면 , 다음으로 넘어간다.

case 1: , System.out.println(" 입금!, 출금!, 조회! "); , break;
 = 숫자값 1 , 입금! , 다음으로 넘어간다.

case 2: , System.out.println(" 입금!, 출금!, 조회! "); , break;
 = 숫자값 2 , 출금! , 다음으로 넘어간다.

case 3: , System.out.println(" 입금!, 출금!, 조회! "); , break;
 = 숫자값 3 , 조회! , 다음으로 넘어간다.

 default: , System.out.println("그런명령은 존재하지 않습니다."); , break;
 = 만약 다른 숫자값이면 , 그런명령은 존재하지 않습니다. , 다음으로 넘어간다.

그외 복습한 코딩

Dice.1

```java
public class DIce {
    신규 *
    public static void main(String[] args){
        final int MIN = 1;
        final int MAX = 6;

        final int diceNumber1 = (int)(Math.random()*(MAX - MIN +1)) + MIN;
        final int diceNumber2 = (int)(Math.random()*(MAX - MIN +1)) + MIN;

        final int diceSum = (diceNumber1 + diceNumber2);

        final int GAME_WINNER_CHECK = 4;

        System.out.println("첫번째주사위: " + diceNumber1);
        System.out.println("두번째주사위: " + diceNumber2);
        System.out.println("두 주사위의 합: " + diceSum);

        if (diceSum % GAME_WINNER_CHECK == 0) {
            System.out.println("승리!");
        }else {
            System.out.println("패배!");
        }
    }
}
```

Dice.2

```java
import java.util.Arrays;

2개 사용 위치  신규 *
final class Dice1{
    2개 사용 위치
    final private int MIN = 1;
    1개 사용 위치
    final private int MAX = 6;
    1개 사용 위치
    final private int WIN_DECISION1 = 3;
    1개 사용 위치
    final private int WIN_DECISION2 = 4;

    1개 사용 위치
    final int MAX_DICE = 4;
    5개 사용 위치
    final int[] diceNumArray = new int[MAX_DICE];
    4개 사용 위치
    final int totalScore;

    1개 사용 위치  신규 *
    public Dice1(){
        int diceNumberSum = 0;
        for (int i = 0; i < diceNumArray.length; i++) {
            diceNumArray[i] = (int)(Math.random()*(MAX - MIN + 1)) +MIN;
            diceNumberSum += diceNumArray[i];
            System.out.println("diceNumArray: " + diceNumArray[i]);
        }
        this.totalScore = diceNumberSum;
    }
    1개 사용 위치  신규 *
    public Boolean checkWin () {
        if ((totalScore % WIN_DECISION1 == 0)||(totalScore % WIN_DECISION2 == 0)) {
            return true;
        }
        return false;
    }
    신규 *
    @Override
    public String toString() { return "주사위값:" + Arrays.toString(diceNumArray) +'\n' + "주사위 총합: " + totalScore; }
}
0개의 사용위치  신규 *
public class DiceTest {
    신규 *
    public static void main(String[] args) {
        Dice1 dice = new Dice1();
        System.out.println(dice);
        System.out.println(dice.checkWin()? "승리!" : "패배!");
    }
}
```

Member.1

```java
public class IfTest {
    신규 *
    public static void main(String[] args) {
        final int PERMIT_AGE = 18;
        final int inputAge = 19;
        final int PERMIT_KIDS = 13;

        if (PERMIT_AGE < inputAge) {
            System.out.println("입장 가능합니다.");
        }else {
        System.out.println("입장 불가능합니다.");
        }

        System.out.println("");

        if (PERMIT_AGE < inputAge) {
            System.out.println("성인용입니다.");
        }
        if (PERMIT_KIDS >= inputAge) {
            System.out.println("아동입니다.");
        }
    }
}
```

Member.2

```java
import java.util.Scanner;

2개 사용 위치  신규 *
class Member2 {
    3개 사용 위치
    private String Name;
    2개 사용 위치
    private int Age;
    2개 사용 위치
    private String Email;
    2개 사용 위치
    private int PassWord;
    4개 사용 위치
    private Scanner scanner =new Scanner(System.in);
    1개 사용 위치  신규 *
    public Member2(){
        System.out.println("이름을 작성해주세요: ");
        Name = scanner.nextLine();
        System.out.println("나이을 작성해주세요: ");
        Age = scanner.nextInt();
        System.out.println("이메일을 작성해주세요: ");
        Email = scanner.next();
        System.out.println("비밀번호을 작성해주세요: ");
        PassWord = scanner.nextInt();
    }
    신규 *
    @Override
    public String toString() {
        return  Name + "회원님의 정보는" + '\n' +
                "이름: " + Name + '\n' +
                "나이: " + Age + '\n' +
                "이메일: " + Email + '\n' +
                "비밀번호: " + PassWord + " 입니다.";
    }
}
0개의 사용위치  신규 *
public class Person2 {
    신규 *
    public static void main(String[] args) {
        Member2 member = new Member2();
        System.out.println(member);
    }
}
```

한 주간 배우면서 아직은 배운것도 나오고 해서 이해하면서 할 수 있으나 이후에 코딩이 늘어나면 이해할수 있을지 모르겠습니다.