Examples

Week 7

ArrayLength.java

```
Enter 5 intergers >> 2 3 45 67 5 ↵
Average: 24.4
```

```
import java.util.Scanner;

public class ArrayLength {
    public static void main(String[] args) {
        int intArray[] = new int[5];
        int sum=0;

        Scanner scanner = new Scanner(System.in);
        System.out.print( "Enter " + intArray.length + " intergers >> ");
        for(int i=0; i<intArray.length; i++)
            intArray[i] = scanner.nextInt();

        for(int i=0; i<intArray.length; i++)
            sum += intArray[i];

        System.out.println( "Average: " + (double)sum/intArray.length );
        scanner.close();
    }
}
```

Foreach.java

```
public class Foreach
{
    enum Week { MONDAY, TUESDAY, WENDSEDAY, THURSDAY,
                                    FRIDAY, SATURDAY, SUNDAY }

    public static void main( String[] args )
    {
        int[] num = {1,2,3,4,5};
        int sum = 0;
        for(int k : num)
            sum += k;
        System.out.println( "Sum: " + sum);

        String names[] = { "apple", "pear", "banana", "cherry",
                                          "strawberry", "grapes" };
        for( String s : names )
            System.out.print( s + " " );
        System.out.println();
```

Foreach.java

```
        for( Week day : Week.values() )
            System.out.print( day + " " );
        System.out.println();
    }
}
```

```
Sum: 15
apple pear banana cherry strawberry grapes
MONDAY TUESDAY WENDSEDAY THURSDAY FRIDAY SATURDAY SUNDAY
```

ScoreAverage.java

```
public class ScoreAverage
{
    public static void main(String[] args)
    {
        double score[][] = { {3.3, 3.4},
                             {3.5, 3.6},
                             {3.7, 4.0},
                             {4.1, 4.2} };
        double sum=0;
        for( int year=0;  year<score.length;  year++)
            for( int term=0;  term<score[year].length;  term++ )
                sum += score[year][term];

        int n = score.length;
        int m = score[0].length;
        System.out.println("Average: " + sum/(n*m) );
    }
}
```

```
Average: 3.725
```

IrregularArray.java

```
public class IrregularArray
{
    public static void main (String[] args)
    {
        int intArray[][] = new int[4][];
        intArray[0] = new int[3];
        intArray[1] = new int[2];
        intArray[2] = new int[3];
        intArray[3] = new int[2];

        for (int i = 0; i < intArray.length; i++)
            for (int j = 0; j < intArray[i].length; j++)
                intArray[i][j] = (i+1)*10 + j;

        for (int i = 0; i < intArray.length; i++) {
            for (int j = 0; j < intArray[i].length; j++)
                System.out.print(intArray[i][j]+" ");
            System.out.println();
        }
    }
}
```

```
10 11 12
20 21
30 31 32
40 41
```

ReturnArray.java

```
public class ReturnArray
{
    static int[] makeArray()
    {
        int temp[] = new int[4];
        for (int i=0; i<temp.length; i++)
            temp[i] = i;
        return temp;
    }

    public static void main (String[] args)
    {
        int intArray[];
        intArray = makeArray();
        for (int i=0; i<intArray.length; i++)
            System.out.print(intArray[i] + " ");
        System.out.println();
    }
}
```

```
0 1 2 3
```

Calc.java

```
public class Calc
{
    public static void main (String[] args)
    {
        double sum = 0.0;

        for (int i=0; i<args.length; i++)
            sum += Double.parseDouble(args[i]);

        System.out.println("sum = " + sum);
    }
}
```

```
$ java Calc 2 20.5 88.1
sum = 110.6
```

BarChart.java

```
public class BarChart
{
    public static void main( String[] args )
    {
        int[] data = { 5, 2, 18, 9, 6, 11, 4, 5, 13, 7 };

        for( int n : data )
        {
            System.out.printf( "%2d ", n );
            for( int i=0; i < n; i++)
                System.out.print( '*' );
            System.out.println();
        }
    }
}
```

```
 5 *****
 2 **
18 ******************
 9 *********
 6 ******
11 ***********
 4 ****
 5 *****
13 *************
 7 *******
```

TestAverage.java

```
public class TestAverage
{
    public static void main( String[] args )
    {
        int[][] score = { { 87, 96, 70 },
                          { 68, 87, 90 },
                          { 94, 100, 90 },
                          { 100, 81, 82 },
                          { 83, 65, 85 },
                          { 78, 87, 65 },
                          { 85, 75, 83 },
                          { 91, 94, 100 },
                          { 76, 72, 84 },
                          { 87, 93, 73 } };
```

```
 87    96    70  84.3
 68    87    90  81.7
 94   100    90  94.7
100    81    82  87.7
 83    65    85  77.7
 78    87    65  76.7
 85    75    83  81.0
 91    94   100  95.0
 76    72    84  77.3
 87    93    73  84.3
84.9  85.0  82.2
```

TestAverage.java

```
        for ( int student=0; student < score.length; student++ )
        {
            for( int test=0; test < score[student].length; test++ )
                System.out.printf( "%4d  ", score[student][test] );
            System.out.printf( "%.1f\n", getAverage( score[student] ));
        }

        for( int test=0; test < score[0].length; test++ )
            System.out.printf( "%4.1f  ", getTestAverage( score, test ) );
        System.out.println();
    }
```

TestAverage.java

```
    public static double getAverage( int[] arr )
    {
        double m = 0.0;
        for( int d : arr )
            m += d;
        return m / arr.length;
    }

    public static double getTestAverage( int[][] arr, int test )
    {
        double m = 0.0;
        for( int student=0; student < arr.length; student++ )
            m += arr[student][test];
        return m / arr.length;
    }
}
```

Ex07_1.java

- 병렬패리티
- 아래와 같이 8bit X 8bit 블록 데이터가 2차원 배열의 형태로 주어진다. 오른쪽 그림과 같이 가로 방향 패리티 비트와 세로 방향 패리티 비트를 추가하여 출력하시오.(단, 짝수 패리티를 적용한다.)

```
int[][] input = { { 1, 0, 1, 0, 1, 1, 1, 1 },
                  { 1, 0, 0, 0, 0, 0, 1, 1 },
                  { 0, 1, 0, 0, 0, 0, 0, 0 },
                  { 1, 1, 1, 1, 0, 0, 0, 0 },
                  { 1, 0, 1, 1, 1, 0, 0, 1 },
                  { 0, 0, 0, 0, 0, 1, 1, 1 },
                  { 1, 1, 1, 1, 1, 1, 1, 1 },
                  { 0, 1, 1, 1, 1, 0, 0, 0 } };
```