



Resource Software Solution

JAVA BASIC

Lab Guides

Lab Guide 3: FOR

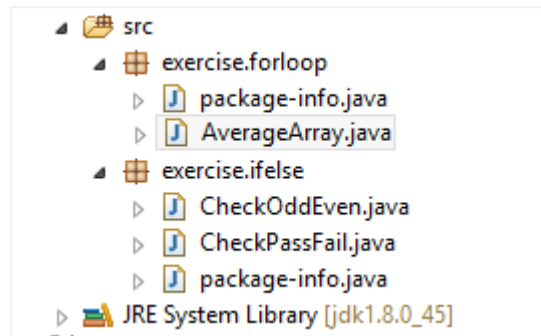
Specifications

Create a new package name **exercise.forloop**.

Write a program called **AverageArray** to initialization an int array, after that found average of this array and display the average.

Source code

Project structure:



- **AverageArray class**

```
AverageArray.java
1 package exercise.forloop;
2
3 public class AverageArray {
4
5     public static void main(String[] args) {
6
7         int[] intArray = { 12, 5, 9, 8, 11, 86 };
8         int sum = 0;
9         double average = 0.0d;
10        int length = intArray.length;
11
12        for (int i = 0; i < length; i++) {
13            sum += intArray[i]; // sum = sum + intArray[i]
14        }
15
16        average = sum / length;
17
18        System.out.println("Average of all element in intArray: " + average);
19    }
20
21 }
22
```

Output

```
Average of all element in intArray: 21.0
```

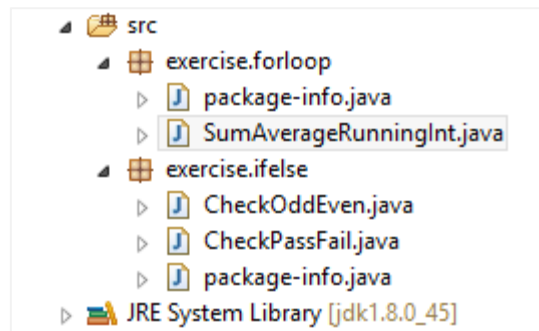
Lab Guide 4: FOR

Specifications

Write a program that sums the first 100 positive integers.

Source code

Project structure:



- **SumAverageRunningInt class**

```
1 package exercise.forloop;
2
3 public class SumAverageRunningInt {
4
5     public static void main(String[] args) {
6         // Define variables
7         int sum = 0; // The accumulated sum, init to 0
8         double average; // average in double
9         int lowerBound = 1;
10        int upperBound = 100;
11
12        // Use a for-loop to sum from lowerbound to upperbound
13        for (int number = lowerBound; number <= upperBound; ++number) {
14            // The loop index variable number = 1, 2, 3, ..., 99, 100
15            sum += number; // same as "sum = sum + number"
16        }
17        // Compute average in double. Beware that int / int produces int!
18        average = sum / upperBound;
19
20        // Print sum and average
21
22        System.out.println("Average of all 100 first numbers: " + average);
23    }
24 }
25
```

Output

```
Average of 100 first number: 50.0
```

Lab Guide 5: FOR

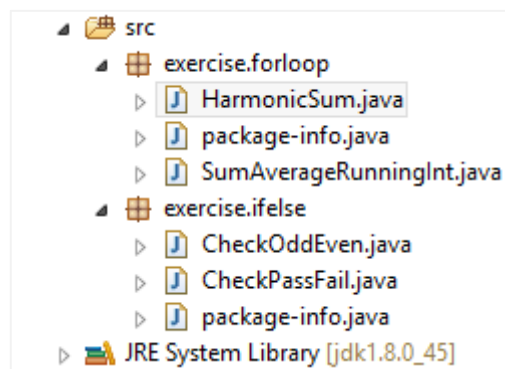
Specifications

Write a program called **HarmonicSum** to compute the sum of a harmonic series, as shown below, where $n=50000$. The program shall compute the sum from *left-to-right* as well as from the *right-to-left*. Are the two sums the same? Obtain the absolute difference between these two sums and explain the difference. Which sum is more accurate?.

$$Harmonic(n) = 1 + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{n}$$

Source code

Project structure:



- **HarmonicSum class**

```
HarmonicSum.java
1 package exercise.forloop;
2
3 /**
4  * Write a program called HarmonicSum to compute the sum of a harmonic series,
5  * as shown below, where n=50000. The program shall compute the sum from
6  * left-to-right as well as from the right-to-left. Obtain the difference
7  * between these two sums and explain the difference.
8  *
9  * Harmonic(n) = 1 + 1/2 + 1/3 + ... 1/n
10  */
11
12 public class HarmonicSum {
13
14     public static void main(String[] args) {
15
16         int n = 50000;
17         double sumL2R = printLeftToRightSum(n);
18         double sumR2L = printRightToLeftSum(n);
19         System.out.printf("Difference: %.15f", (sumL2R - sumR2L));
20         System.out.println();
21     }
22
23     /**
24      * Left-to-right harmonic sum.
25      *
26      * @param int n
27      */
28     private static double printLeftToRightSum(int n) {
29         double sum = 0.0;
30         for (int i = 1; i <= n; i++) {
31             sum += (double) 1 / i;
32         }
33         System.out.printf("Left-to-right harmonic sum %.15f", sum);
34         System.out.println();
35
36         return sum;
37     }
38 }
```

```
38
39 /**
40  * Right-to-left harmonic sum.
41  * @param n
42  * @return
43  */
44 private static double printRightToLeftSum(int n) {
45     double sum = 0.0;
46     for (int i = n; i >= 1; i--) {
47         sum += (double) 1 / i;
48     }
49     System.out.printf("Right-to-left harmonic sum %.15f", sum);
50     System.out.println();
51
52     return sum;
53 }
54 }
55
```

Output

```
Left-to-right harmonic sum 11.397003949278504
Right-to-left harmonic sum 11.397003949278519
Difference: -0.000000000000014
```

Lab Guide 6: WHILE

Specifications

Write a program called **Fibonacci** to print the first 20 Fibonacci numbers $F(n)$, where $F(n)=F(n-1)+F(n-2)$ and $F(1)=F(2)=1$. Also compute their average. The output shall look like:

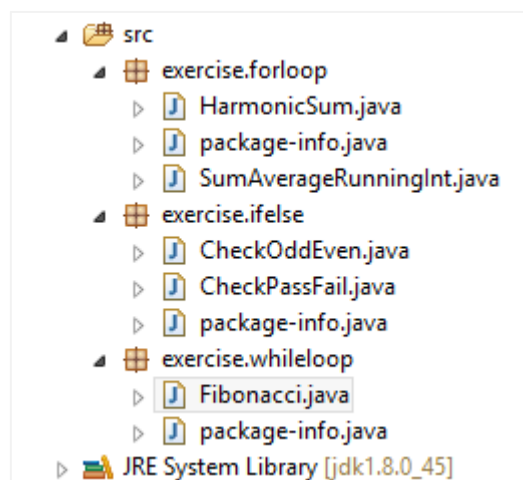
The first 20 Fibonacci numbers are:

1 1 2 3 5 8 13 21 34 55 89 144 233 377 610 987 1597 2584 4181 6765

The average is 885.5

Source code

Project structure:



- **Fibonacci class**

```
Fibonacci.java
1 package exercise.whileloop;
2
3 public class Fibonacci {
4
5     public static void main(String[] args) {
6
7         int n = 1; // The index n for F(n)
8         int fnMinus1 = 1; // F(n-1), init to F(2)
9         int fnMinus2 = 1; // F(n-2), init to F(1)
10        int nMax = 20; // maximum n, inclusive
11        int sum = fnMinus1 + fnMinus2; // Need sum to compute average
12        double average;
13
14        System.out.println("The first " + nMax + " Fibonacci numbers are:");
15
16        while (n <= nMax) {
17            System.out.print(fnMinus1 + " ");
18            // Compute F(n), print it and add to sum
19            // Increment the index n and shift the numbers for the next iteration
20
21            sum = fnMinus1 + fnMinus2;
22
23            fnMinus1 = fnMinus2;
24            fnMinus2 = sum;
25
26            ++n;
27        }
28
29        // Compute and display the average (=sum/nMax).
30        average = sum / nMax;
31        // Beware that int/int gives int.
32        System.out.println("\n The average is " + average);
33    }
34 }
35
```

Output:

The first 20 Fibonacci numbers are:

1 1 2 3 5 8 13 21 34 55 89 144 233 377 610 987 1597 2584 4181 6765

The average is 885.0

Bài tập 1: Mô tả bài toán: Viết chương trình yêu cầu người dùng nhập một số nguyên N. Hiển thị chuỗi số theo các yêu cầu sau:

- Yêu cầu 1: In dãy số 1, 2, 3, ..., N.
- Yêu cầu 2: In dãy số N, N-1, N-2, ... 1.
- Yêu cầu 3: In dãy số 1, 3, 5, ..., N.
- Yêu cầu 4: In dãy số 2, 4, 6, ..., N.

Bài tập 2: Viết chương trình yêu cầu người dùng nhập một số nguyên N. Thực hiện các phép tính sau:

- **Yêu cầu 1:** $S1 = \text{Giai thừa của số nguyên } N$.
- **Yêu cầu 2:** $S2 = 1! + 2! + 3! + \dots + N!$
- **Yêu cầu 3:** Đếm và in ra số lượng các số nguyên chia hết cả 3 và 7 trong phạm vi từ 1 – N

Bài tập 3: Viết chương trình hiển thị sơ đồ phòng chiếu theo hình sau:

run:

SƠ ĐỒ CHỖ NGỒI												
MÀN HÌNH												
A	[01]	[02]	[03]	[04]	[05]	[06]	[07]	[08]	[09]	[10]	[11]	[12]
B	[01]	[02]	[03]	[04]	[05]	[06]	[07]	[08]	[09]	[10]	[11]	[12]
C	[01]	[02]	[03]	[04]	[05]	[06]	[07]	[08]	[09]	[10]	[11]	[12]
D	[01]	[02]	[03]	[04]	[05]	[06]	[07]	[08]	[09]	[10]	[11]	[12]
E	[01]	[02]	[03]	[04]	[05]	[06]	[07]	[08]	[09]	[10]	[11]	[12]
F	[01]	[02]	[03]	[04]	[05]	[06]	[07]	[08]	[09]	[10]	[11]	[12]
G	[01]	[02]	[03]	[04]	[05]	[06]	[07]	[08]	[09]	[10]	[11]	[12]
H	[01]	[02]	[03]	[04]	[05]	[06]	[07]	[08]	[09]	[10]	[11]	[12]
I	[01]	[02]	[03]	[04]	[05]	[06]	[07]	[08]	[09]	[10]	[11]	[12]
J	[01]	[02]	[03]	[04]	[05]	[06]	[07]	[08]	[09]	[10]	[11]	[12]
K	[01]	[02]	[03]	[04]	[05]	[06]	[07]	[08]	[09]	[10]	[11]	[12]

Gợi ý: Sử dụng 2 vòng lặp lồng nhau

Bài tập 4: Thiết kế trò chơi *Hãy chọn giá đúng* với quy luật như sau:

- Xây dựng giá bán của một vật phẩm bằng cách tạo ngẫu nhiên một số bằng lệnh random.
- Yêu cầu người dùng đoán giá mua bằng cách nhập vào 1 số nguyên.
- Nếu giá mua > giá bán: Hãy xuất thông báo cho người dùng biết giá trị giá mua nhập vào cao hơn giá trị giá bán và yêu cầu người dùng đoán lại.
- Nếu giá mua < giá bán: Hãy xuất thông báo cho người dùng biết giá trị giá mua nhập vào thấp hơn giá trị giá bán và yêu cầu người dùng đoán lại.
- Ghi nhận số lần đoán sai của người dùng. Mỗi lần nhập sai trừ 01 điểm trong quỹ điểm 10 điểm.
- Nếu giá mua = giá bán: Xuất thông báo nhập đúng và xuất điểm và số lần đoán sai của người dùng

Gợi ý: Thao tác với lệnh ngẫu nhiên Random