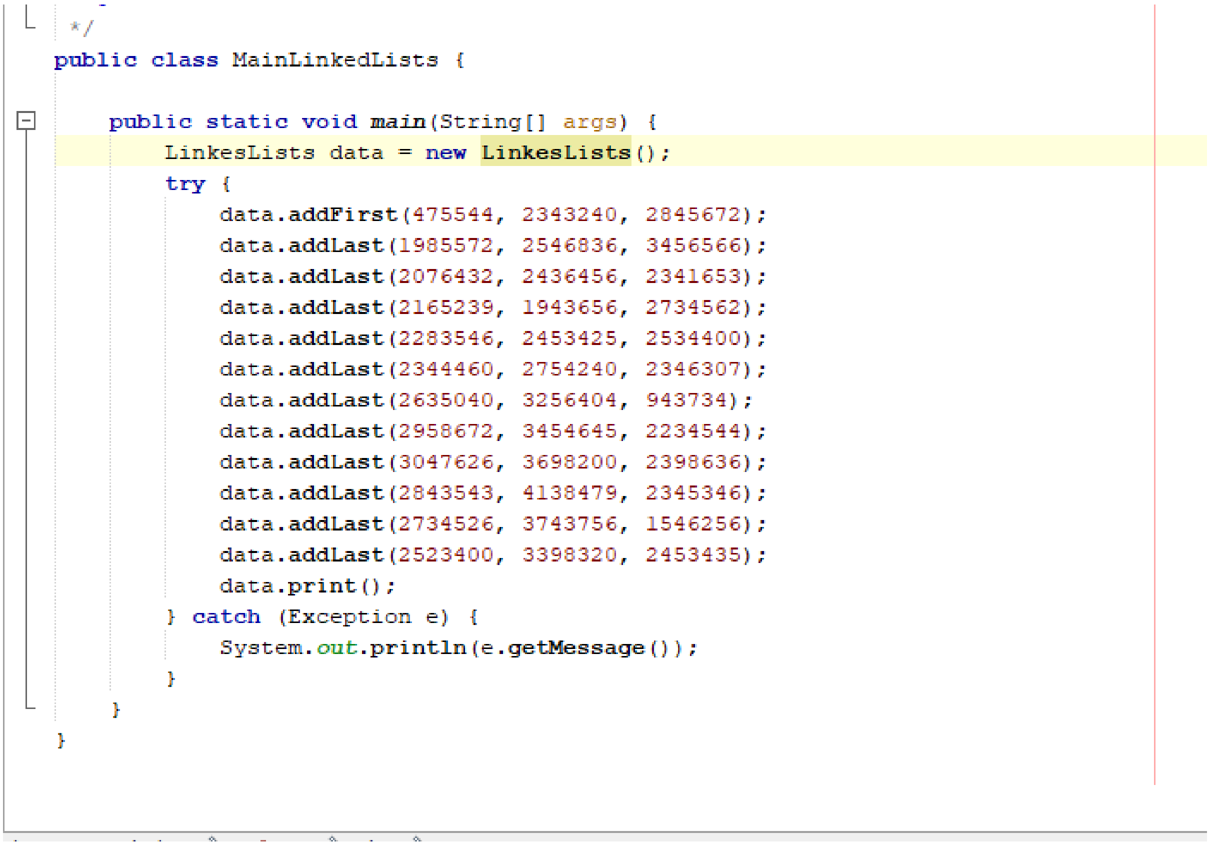
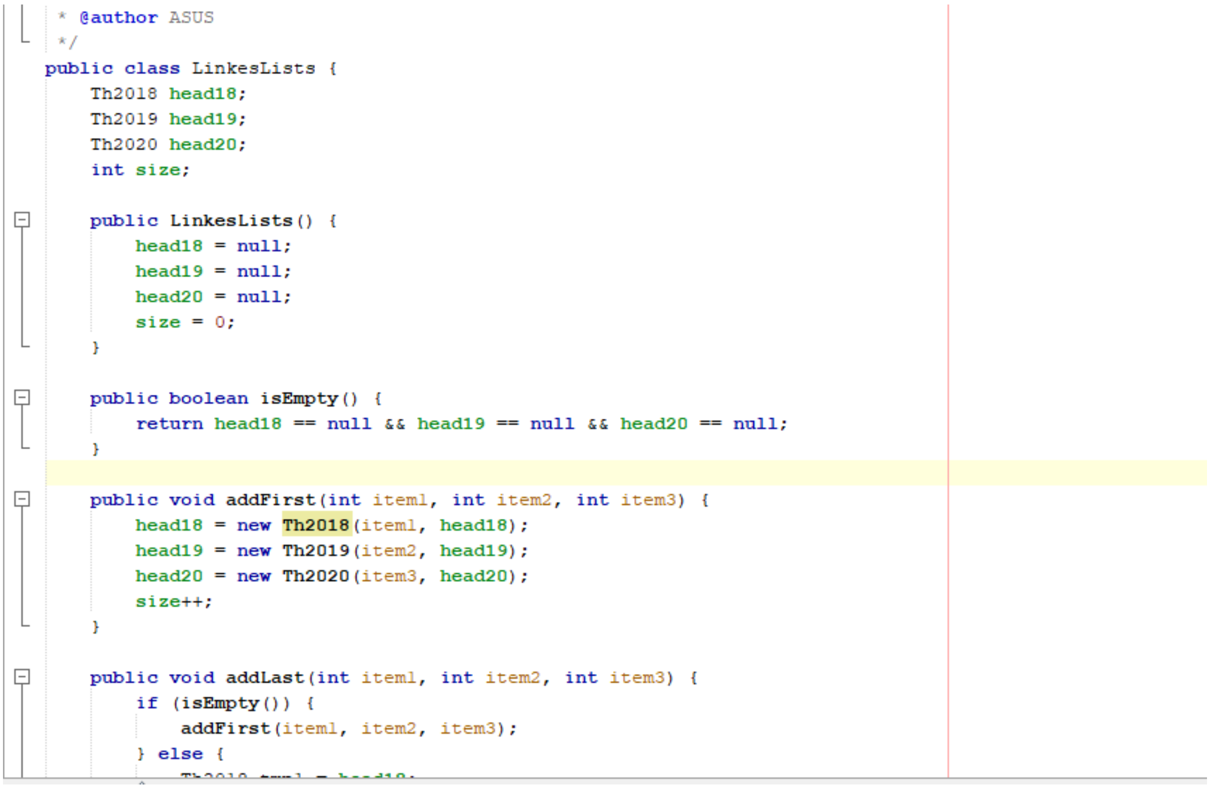
Laporan Quiz 2

1. Main Class

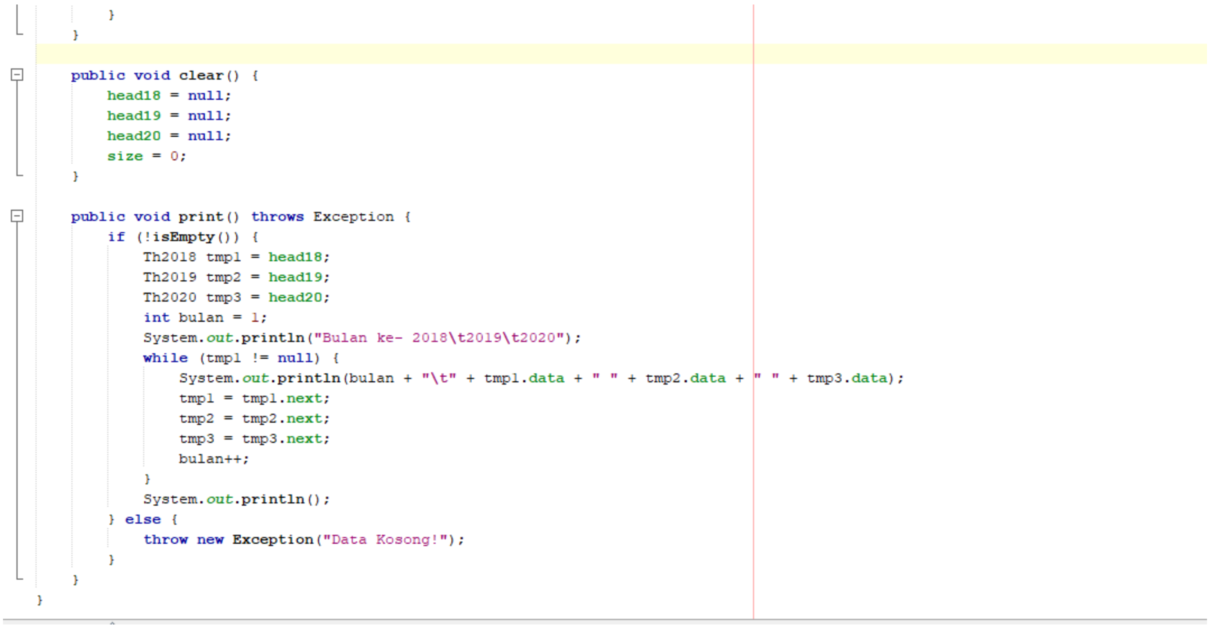


```
 */
public class MainLinkedLists {

    public static void main(String[] args) {
        LinkesLists data = new LinkesLists();
        try {
            data.addFirst(475544, 2343240, 2845672);
            data.addLast(1985572, 2546836, 3456566);
            data.addLast(2076432, 2436456, 2341653);
            data.addLast(2165239, 1943656, 2734562);
            data.addLast(2283546, 2453425, 2534400);
            data.addLast(2344460, 2754240, 2346307);
            data.addLast(2635040, 3256404, 943734);
            data.addLast(2958672, 3454645, 2234544);
            data.addLast(3047626, 3698200, 2398636);
            data.addLast(2843543, 4138479, 2345346);
            data.addLast(2734526, 3743756, 1546256);
            data.addLast(2523400, 3398320, 2453435);
            data.print();
        } catch (Exception e) {
            System.out.println(e.getMessage());
        }
    }
}
```

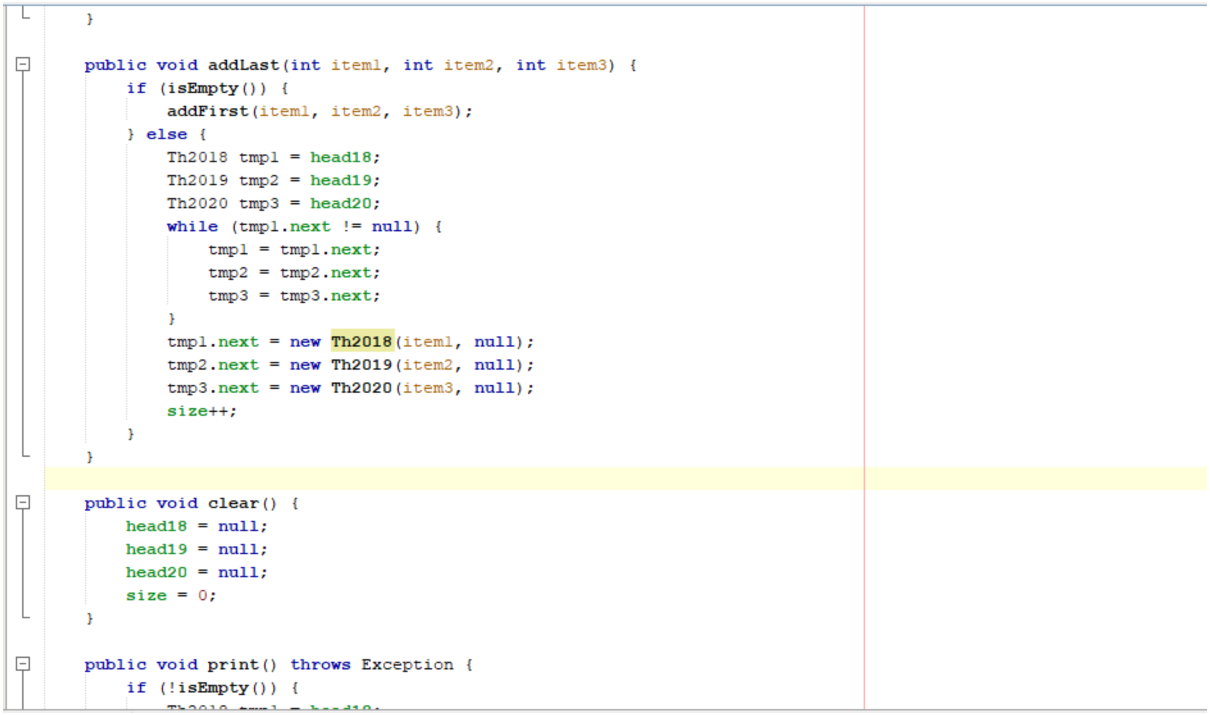
2. Linked List



```
 * @author ASUS
 */
public class LinkesLists {
    Th2018 head18;
    Th2019 head19;
    Th2020 head20;
    int size;

    public LinkesLists() {
        head18 = null;
        head19 = null;
        head20 = null;
        size = 0;
    }

    public boolean isEmpty() {
        return head18 == null && head19 == null && head20 == null;
    }

    public void addFirst(int item1, int item2, int item3) {
        head18 = new Th2018(item1, head18);
        head19 = new Th2019(item2, head19);
        head20 = new Th2020(item3, head20);
        size++;
    }

    public void addLast(int item1, int item2, int item3) {
        if (isEmpty()) {
            addFirst(item1, item2, item3);
        } else {
            Th2018 tmp1 = head18;
```

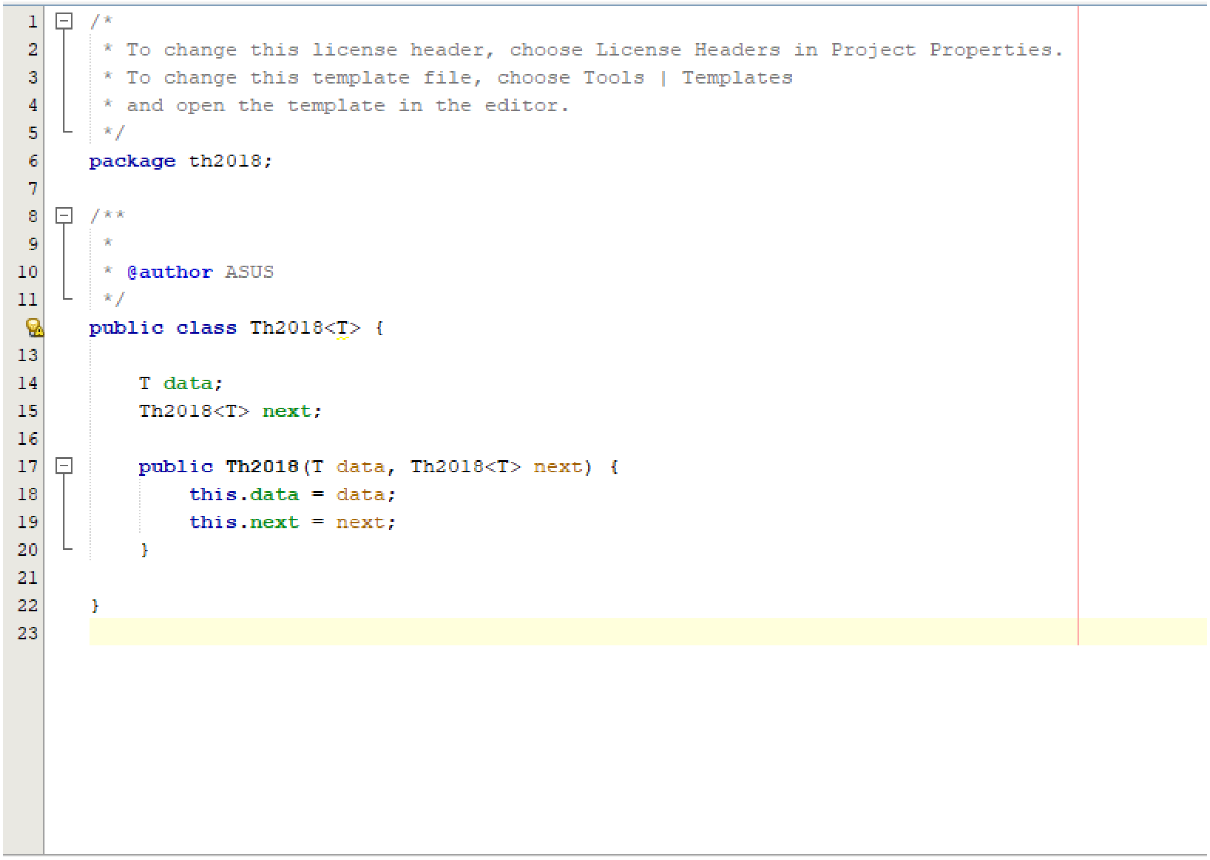


```
        }
    }

    public void clear() {
        head18 = null;
        head19 = null;
        head20 = null;
        size = 0;
    }

    public void print() throws Exception {
        if (!isEmpty()) {
            Th2018 tmp1 = head18;
            Th2019 tmp2 = head19;
            Th2020 tmp3 = head20;
            int bulan = 1;
            System.out.println("Bulan ke- 2018\t2019\t2020");
            while (tmp1 != null) {
                System.out.println(bulan + "\t" + tmp1.data + " " + tmp2.data + " " + tmp3.data);
                tmp1 = tmp1.next;
                tmp2 = tmp2.next;
                tmp3 = tmp3.next;
                bulan++;
            }
            System.out.println();
        } else {
            throw new Exception("Data Kosong!");
        }
    }
}
```



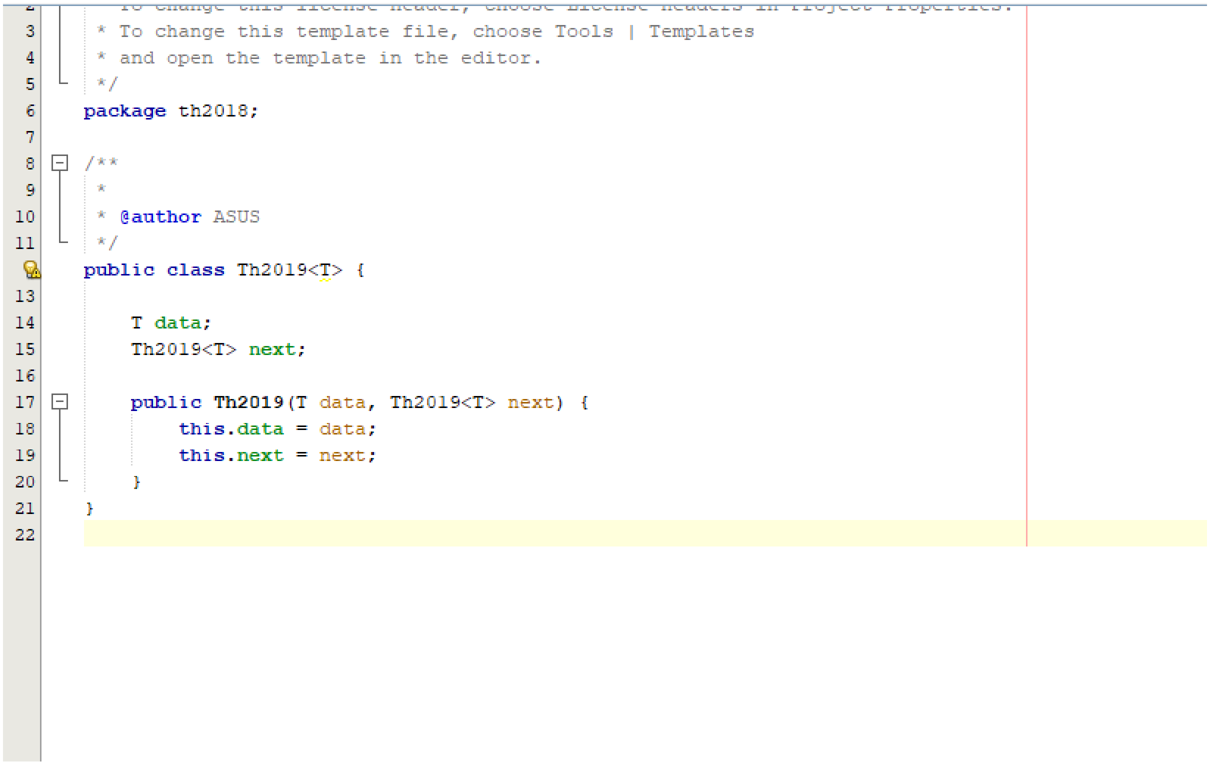
```
    }

    public void addLast(int item1, int item2, int item3) {
        if (isEmpty()) {
            addFirst(item1, item2, item3);
        } else {
            Th2018 tmp1 = head18;
            Th2019 tmp2 = head19;
            Th2020 tmp3 = head20;
            while (tmp1.next != null) {
                tmp1 = tmp1.next;
                tmp2 = tmp2.next;
                tmp3 = tmp3.next;
            }
            tmp1.next = new Th2018(item1, null);
            tmp2.next = new Th2019(item2, null);
            tmp3.next = new Th2020(item3, null);
            size++;
        }
    }

    public void clear() {
        head18 = null;
        head19 = null;
        head20 = null;
        size = 0;
    }

    public void print() throws Exception {
        if (!isEmpty()) {
            Th2018 tmp1 = head18;
```

3. Class Th 2018



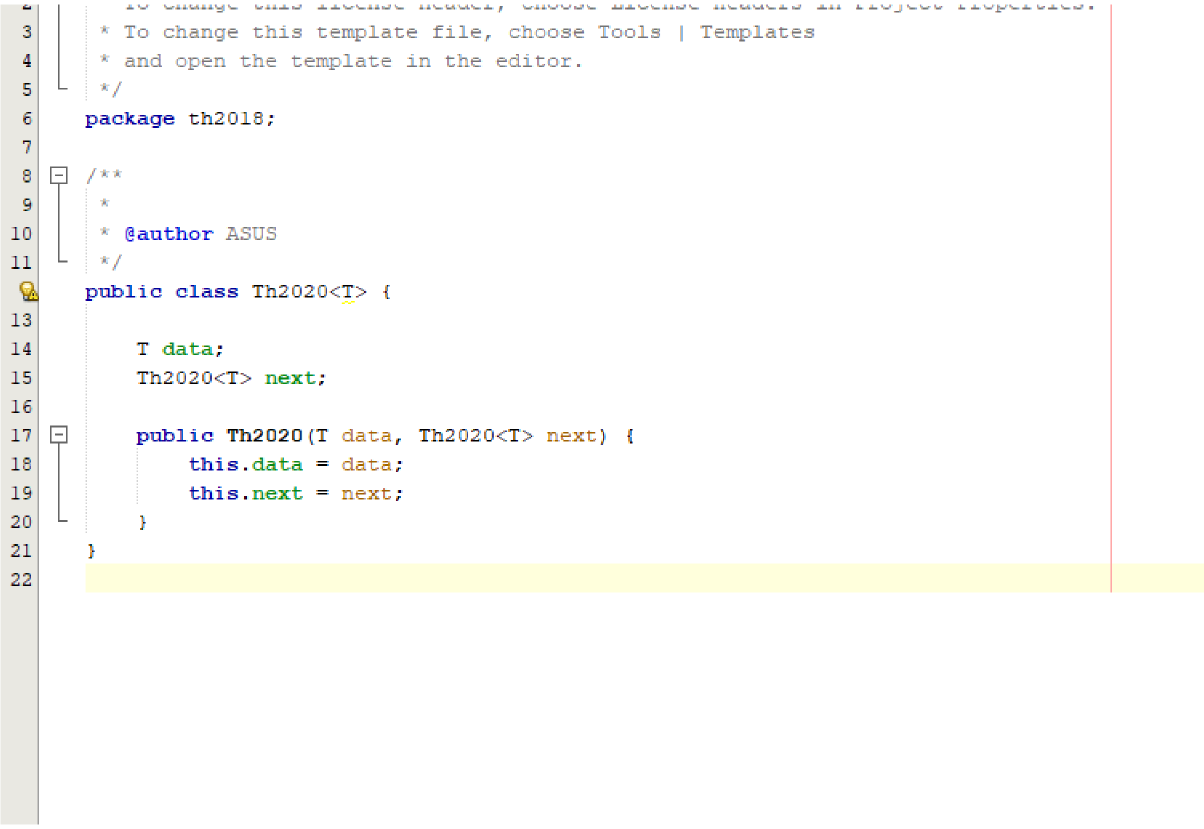
```java
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package th2018;

/**
 *
 * @author ASUS
 */
public class Th2018<T> {

    T data;
    Th2018<T> next;

    public Th2018(T data, Th2018<T> next) {
        this.data = data;
        this.next = next;
    }

}
```

4. Class Th 2019



```java
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package th2018;

/**
 *
 * @author ASUS
 */
public class Th2019<T> {

    T data;
    Th2019<T> next;

    public Th2019(T data, Th2019<T> next) {
        this.data = data;
        this.next = next;
    }
}
```

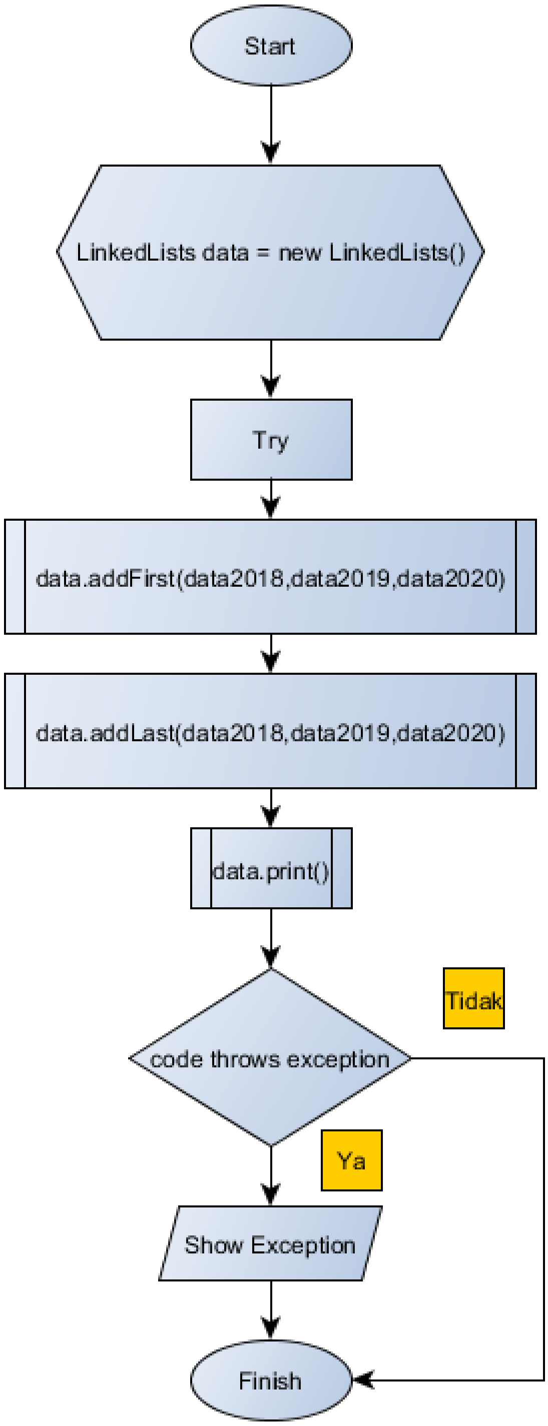
5. Class Th 2020



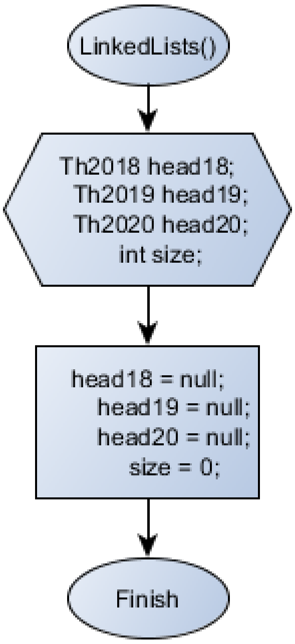
```
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package th2018;

/**
 *
 * @author ASUS
 */
public class Th2020<T> {

    T data;
    Th2020<T> next;

    public Th2020(T data, Th2020<T> next) {
        this.data = data;
        this.next = next;
    }
}

```

Flowchart

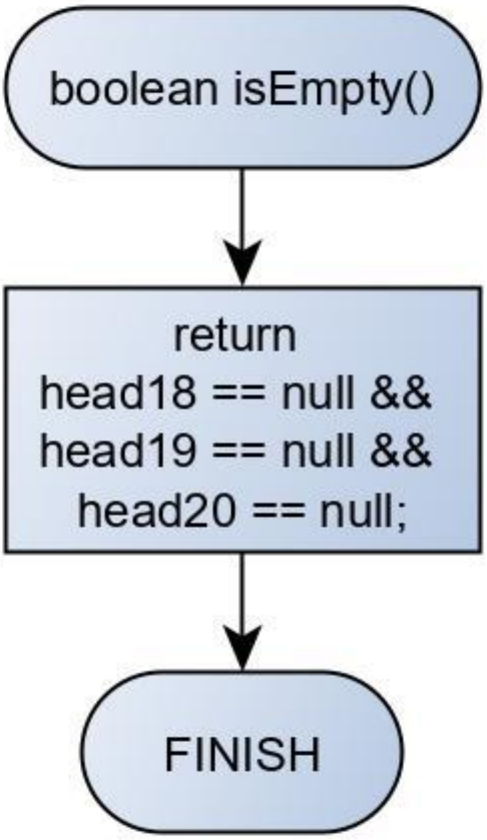
1. Main


Start
LinkedLists data = new LinkedLists()
Try
data.addFirst(data2018,data2019,data2020)
data.addLast(data2018,data2019,data2020)
data.print()
code throws exception
Tidak
Ya
Show Exception
Finish

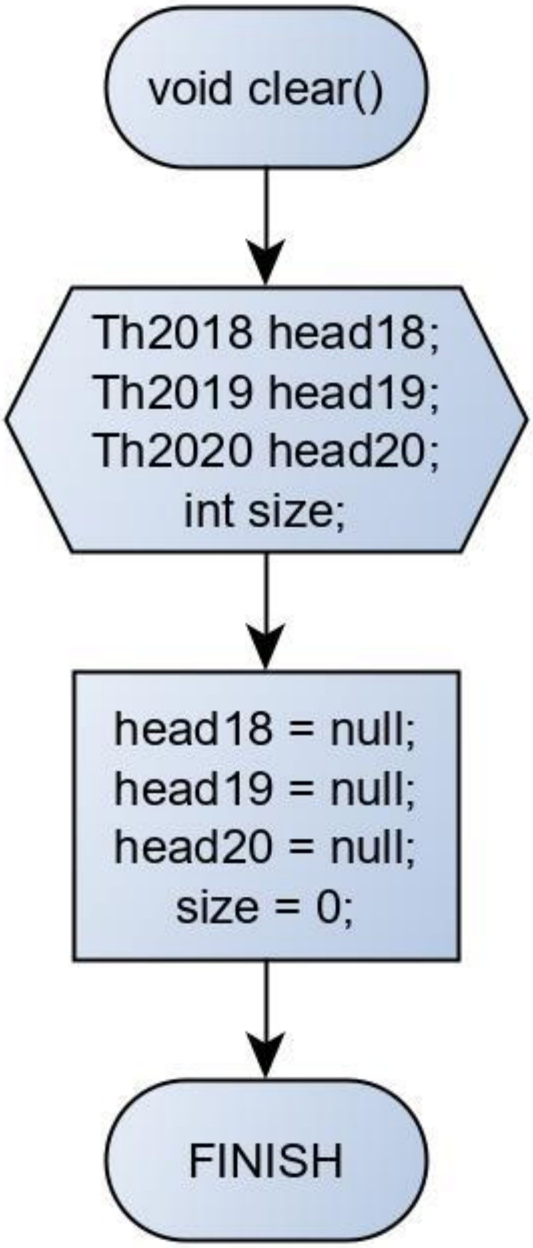
2. LinkedLists



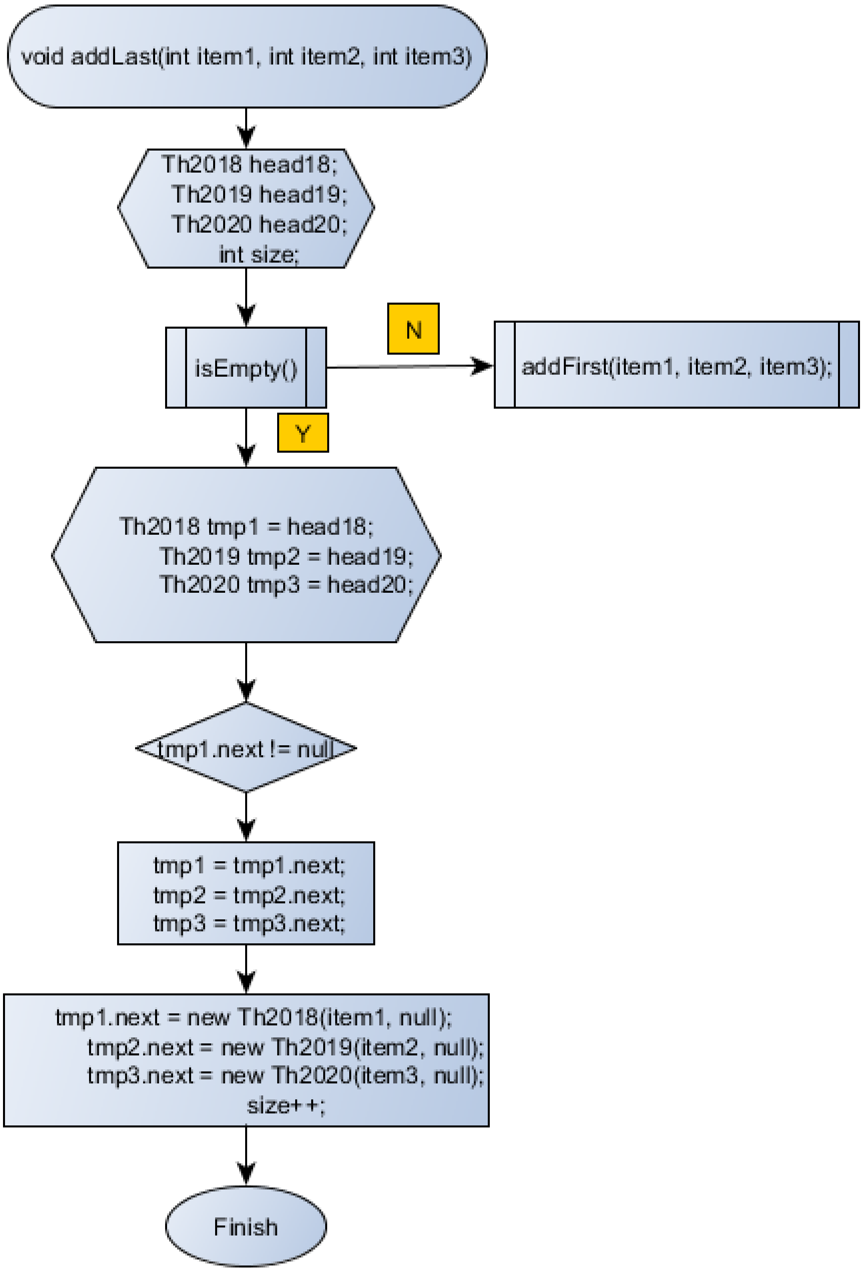
3. isEmpty



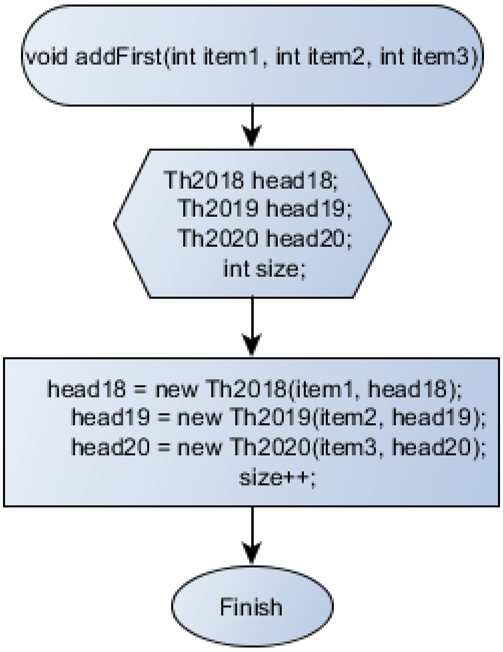
4. Clear


void clear()
Th2018 head18;
Th2019 head19;
Th2020 head20;
int size;
head18 = null;
head19 = null;
head20 = null;
size = 0;
FINISH

5. addLast


void addLast(int item1, int item2, int item3)
Th2018 head18;
Th2019 head19;
Th2020 head20;
int size;
isEmpty()
N
addFirst(item1, item2, item3);
Y
Th2018 tmp1 = head18;
Th2019 tmp2 = head19;
Th2020 tmp3 = head20;
tmp1.next != null
tmp1 = tmp1.next;
tmp2 = tmp2.next;
tmp3 = tmp3.next;
tmp1.next = new Th2018(item1, null);
tmp2.next = new Th2019(item2, null);
tmp3.next = new Th2020(item3, null);
size++;
Finish

6. addFirst


void addFirst(int item1, int item2, int item3)
Th2018 head18;
Th2019 head19;
Th2020 head20;
int size;
head18 = new Th2018(item1, head18);
head19 = new Th2019(item2, head19);
head20 = new Th2020(item3, head20);
size++;
Finish

Penjelasan

Pada kode program saya, saya menggunakan 5 class, yaitu class main, linkedList, dan 3 class untuk menyimpan node per tahun. Pada class main diinputkan data secara statis sesuai data yang diberikan di soal,

1. Menggunakan method addFirst untuk mengisi data pertama, pada LinkedLists data pertama disebut head, sehingga data yang baru langsung dimasukkan ke head.
2. Untuk memasukkan data selanjutnya, digunakan method addLast, yaitu memasukkan data ke node next, dilakukan pengecekan, apabila linkedlist sedang kosong, maka dipanggil method addFirst, jika tidak, maka semua data dilakukan proses traverse lalu data yang baru diinputkan diakhir perulangan traverse.
3. Print data, menggunakan teknik traverse juga, yaitu data akan disimpan sementara pada variabel tertentu lalu ditampilkan, dan variabel tersebut akan diganti dengan node next nya lalu isi variabel tersebut adalah data baru yang akan ditampilkan lagi, terus berulang seperti itu hingga variabel bernilai null atau tidak ada data lagi yang bisa ditampilkan.