**Main :**

public class Class\_main {

public static void main(String[] args) {

SelectMode mode = new SelectMode() ;

mode.select() ;

}//end main

}//end class

**Dnode :**

public class DNode {

information info ;

DNode Llink,Rlink ;

}

**Information :**

public class information {

String name ;

String artist ;

String album ;

String length ;

public information( String name\_music, String artist\_name, String album\_name, String length\_song ) {

name = name\_music ;

artist = artist\_name ;

album = album\_name ;

length = length\_song ;

}

}

**Doubly linkedlist :**

public class DLL {

DNode head, tail, chack;

int count = 0;

void add(information item) {

DNode newnode = new DNode();

newnode.info = item;

if (count == 0) {

head = newnode;

tail = newnode;

count++;

} else {

tail.Rlink = newnode;

newnode.Llink = tail;

tail = newnode;

count++;

} // end if

System.out.println( "Add succeed");

}// end add

void front\_ins(information item, String ref) {

DNode pos = SearchData(ref);

DNode newnode = new DNode();

newnode.info = item;

if (pos != null) {

if (ref.equals(head.info.name)) {

newnode.Rlink = head;

head.Llink = newnode;

head = newnode;

count++;

} else {

newnode.Rlink = pos;

newnode.Llink = pos.Llink;

pos.Llink = newnode;

newnode.Llink.Rlink = newnode;

count++;

} // end if

System.out.println("insert succeed");

} else {

System.out.println("insert Failed") ;

return ;

} // end if

}// end mathod

void behind\_ins(information item, String ref) {

DNode pos = SearchData(ref);

DNode newnode = new DNode();

newnode.info = item;

if (pos != null) {

if (ref.equals(tail.info.name)) {

/\* behind head \*/

tail.Rlink = newnode;

newnode.Llink = tail;

tail = newnode;

count++;

} else {

newnode.Rlink = pos.Rlink;

newnode.Llink = pos;

pos.Rlink.Llink = newnode;

pos.Rlink = newnode;

count++;

} // end if

System.out.println("insert succeed");

} // end if

}// end mathod

void remove(String pos) {

// ลบหัว

DNode ref = SearchData(pos);

if (ref == null) {

System.out.println("Don't have data ! ");

return;

}

if (pos.equals(head.info.name)) {

// ลบหัว

if (count > 1) {

head = ref.Rlink;

head.Llink = null;

count--;

} else if (count == 1) {

head = null;

tail = null;

count--;

} // end if

} else if (pos.equals(tail.info.name)) {

// ลบท้าย

tail = ref.Llink;

tail.Rlink = null;

count--;

} else {

ref.Llink.Rlink = ref.Rlink;

ref.Rlink.Llink = ref.Llink;

ref.Rlink = null;

ref.Llink = null;

count--;

} // end if

System.out.println("Remove succeed");

}// end mathod

void show() {

chack = head;

int number = 0;

if (chack == null) {

System.out.println("Node don't have data");

return;

} // end if

System.out.println();

while (chack != null) {

number++;

System.out.println(number + ". Music name : " + chack.info.name);

chack = chack.Rlink;

} // end loop

System.out.println();

}// end mathod

void showAll() {

chack = head;

int number = 0;

if (chack == null) {

System.out.println("Node don't have data");

return;

} // end if

System.out.println();

while (chack != null) {

number++;

System.out.println(

number + ". " +

"Name: " + chack.info.name + " | " +

"Artist: " + chack.info.artist + " | " +

"Album: " + chack.info.album + " | " +

"Length: " + chack.info.length

) ;

chack = chack.Rlink;

} // end loop

System.out.println();

}// end mathod

DNode SearchData(String ref) {

DNode node;

node = head;

if (node == null) {

return null;

} // end if

for (int i = 0; i < count; i++) {

if (ref.equals(node.info.name)) {

return node;

}

node = node.Rlink;

} // end for

return null;

} // end mathod

}// end class

**SelectMode :**

import java.util.Scanner;

public class SelectMode {

void select() {

DLL D = new DLL() ;

Scanner data = new Scanner(System.in) ;

String name = "" ;

String artist = "" ;

String album = "" ;

String length = "" ;

boolean end\_pg = true ;

String mode ;

System.out.println();

while(end\_pg){

System.out.println(

"\n1 : Add Music to Playlist\n" +

"2 : Insert Music Before Selected Song\n" +

"3 : Insert Music After Selected Song\n" +

"4 : Remove Music\n" +

"5 : Show Playlist\n" +

"6 : Exit Program\n"

);

System.out.println();

System.out.println("Select Mode: " ) ;

mode = data.nextLine().trim() ;

if( mode.equals( "1" ) ) {

System.out.print("What name song will u add ? : ") ;

name = data.nextLine().trim() ;

System.out.print("What artist name ? : ") ;

artist = data.nextLine().trim() ;

System.out.print("What album name ? : ") ;

album = data.nextLine().trim() ;

System.out.print("How long is this song ? : ") ;

length = data.nextLine().trim() ;

information info = new information(name, artist, album, length ) ;

D.add(info) ;

} else if( mode.equals( "2" ) ){

System.out.print("What name song will u add ? : ") ;

name = data.nextLine().trim() ;

System.out.print("What artist name ? : ") ;

artist = data.nextLine().trim() ;

System.out.print("What album name ? : ") ;

album = data.nextLine().trim() ;

System.out.print("How long is this song ? : ") ;

length = data.nextLine().trim() ;

information info = new information(name, artist, album, length ) ;

String pos ;

D.show() ;

System.out.print("Insert song where ? : ") ;

pos = data.nextLine().trim() ;

D.front\_ins( info, pos ) ;

} else if( mode.equals( "3" ) ){

System.out.print("What name song will u add ? : ") ;

name = data.nextLine().trim() ;

System.out.print("What artist name ? : ") ;

artist = data.nextLine().trim() ;

System.out.print("What album name ? : ") ;

album = data.nextLine().trim() ;

System.out.print("How long is this song ? : ") ;

length = data.nextLine().trim() ;

information info = new information(name, artist, album, length ) ;

String pos ;

D.show() ;

System.out.print("Insert song where ? : ") ;

pos = data.nextLine().trim() ;

D.behind\_ins( info, pos ) ;

} else if( mode.equals( "4" ) ){

String pos ;

D.show( ) ;

System.out.print( "What music would you like to remove ? : ") ;

pos = data.nextLine( ).trim() ;

D.remove(pos) ;

D.show( ) ;

} else if( mode.equals("5") ){

D.showAll() ;

} else if( mode.equals("6") ) {

end\_pg = false ;

data.close() ;

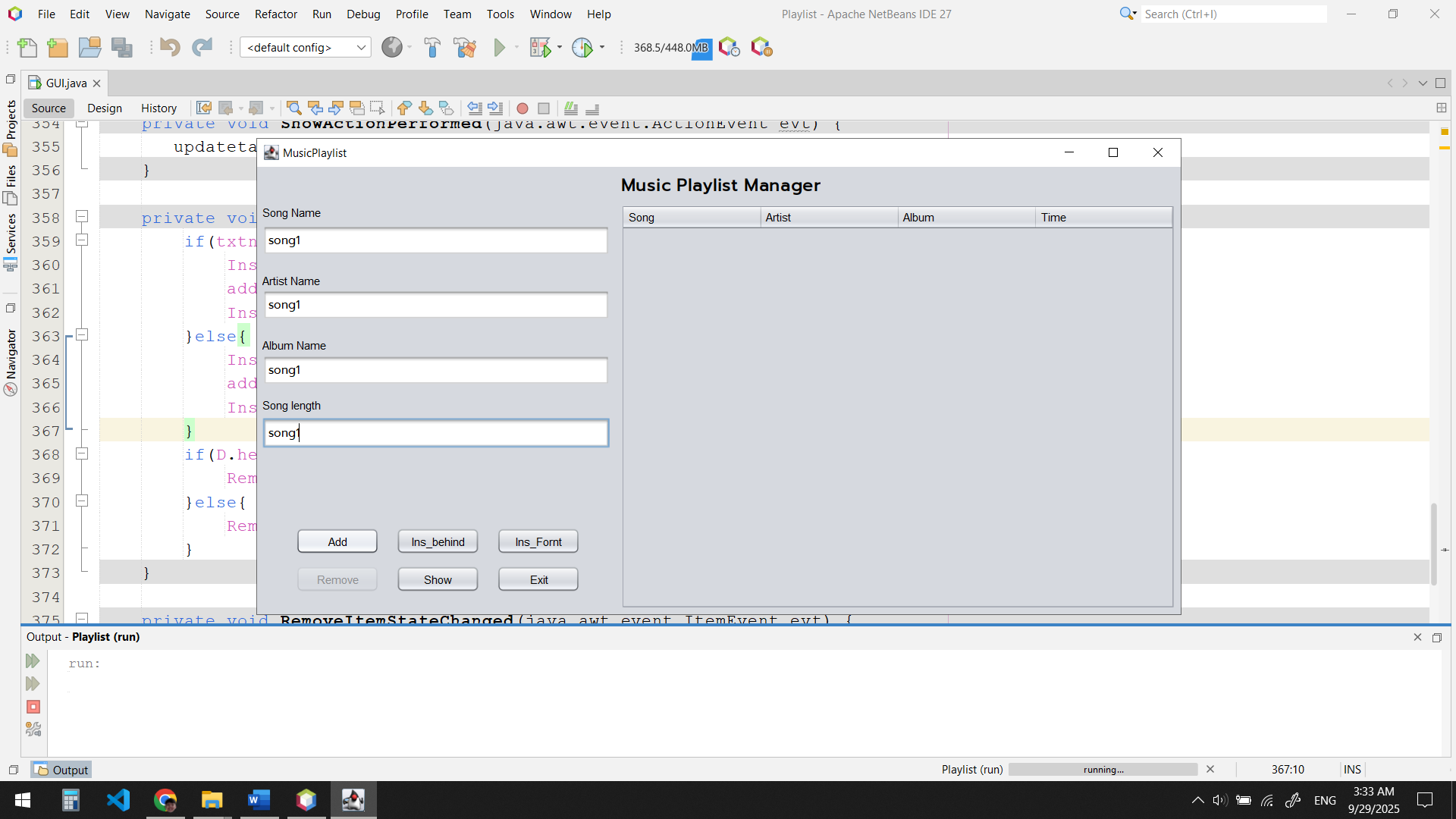
} //end if

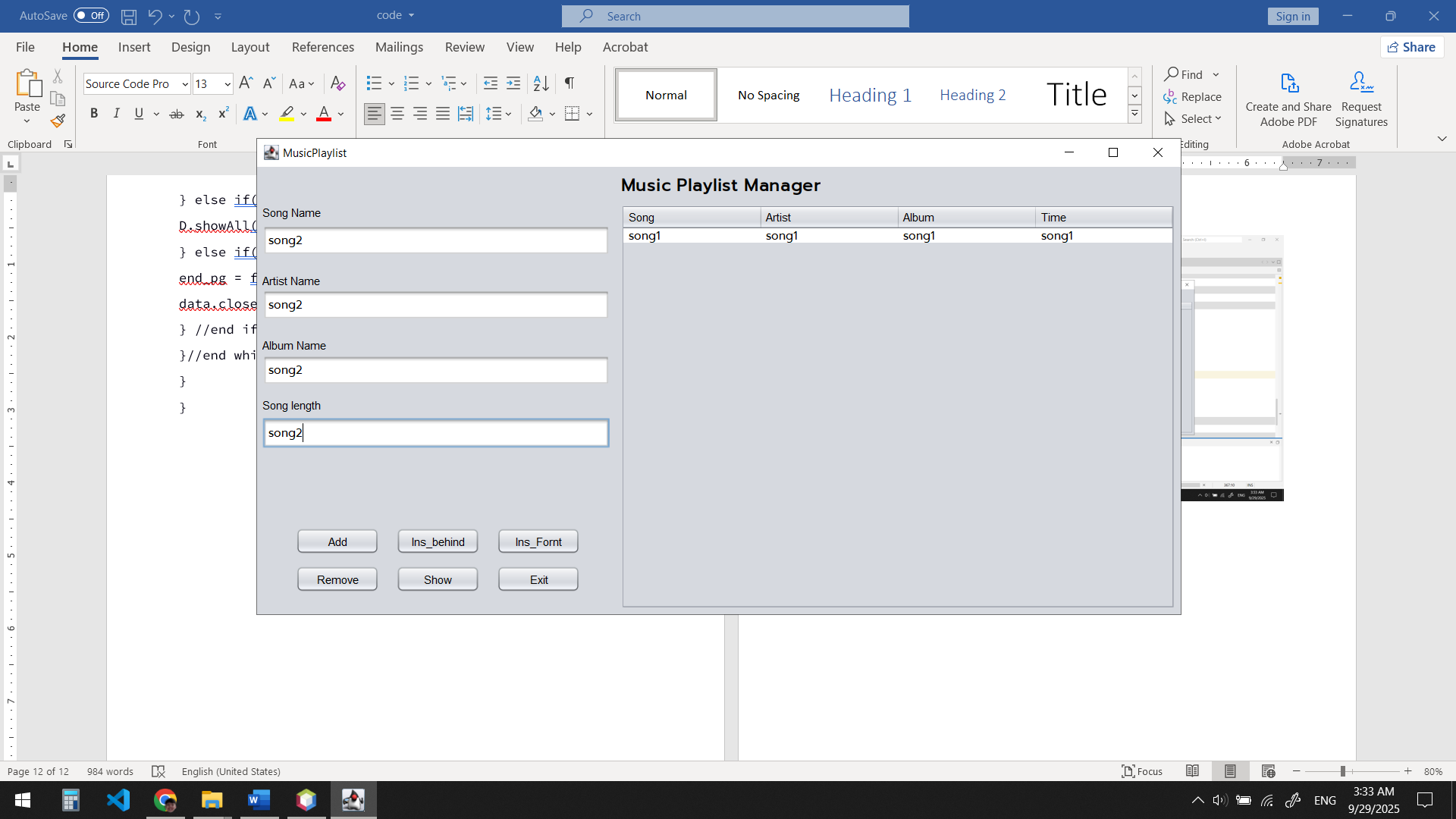
}//end while

}

}

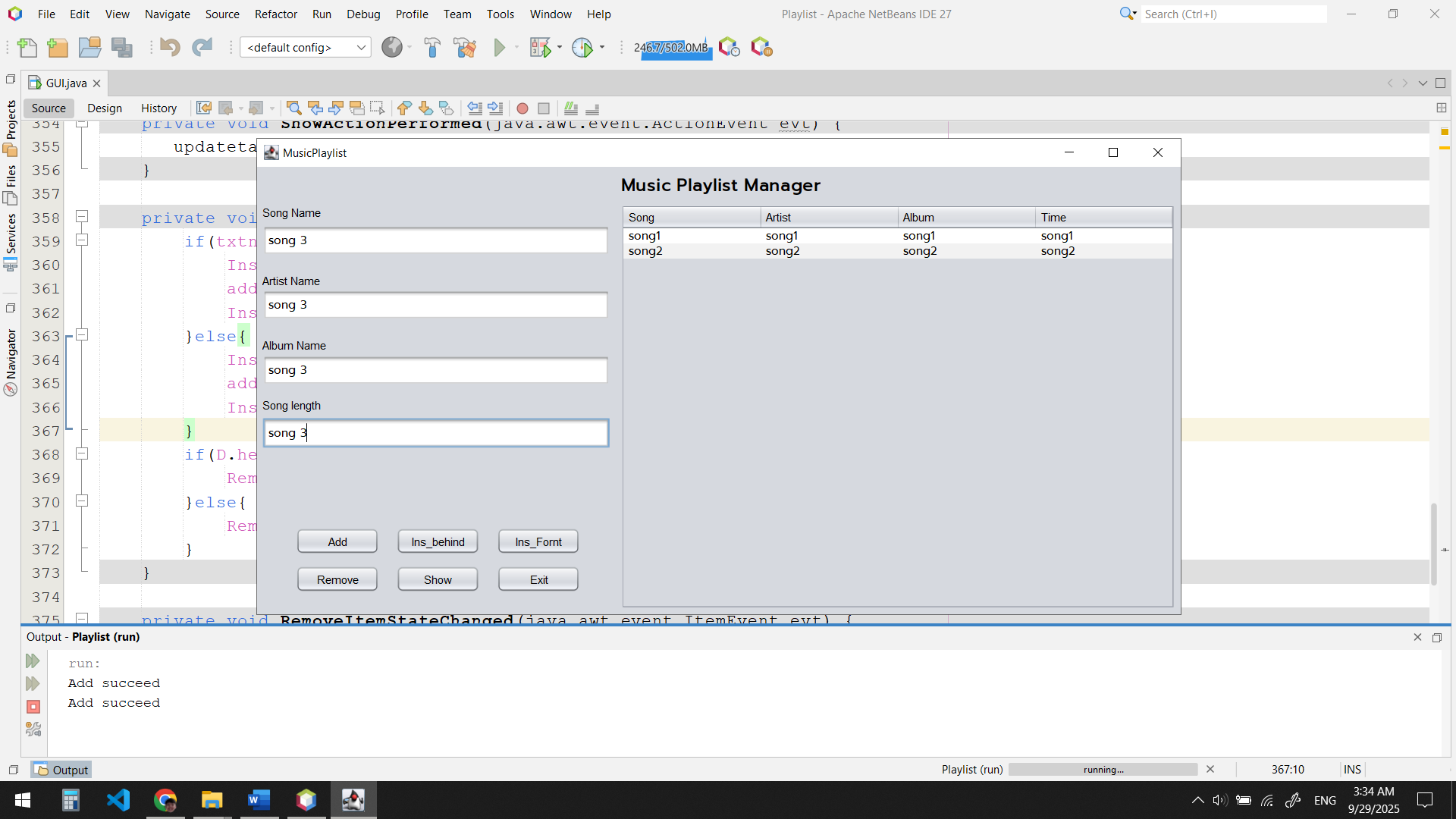
**Testcase :**

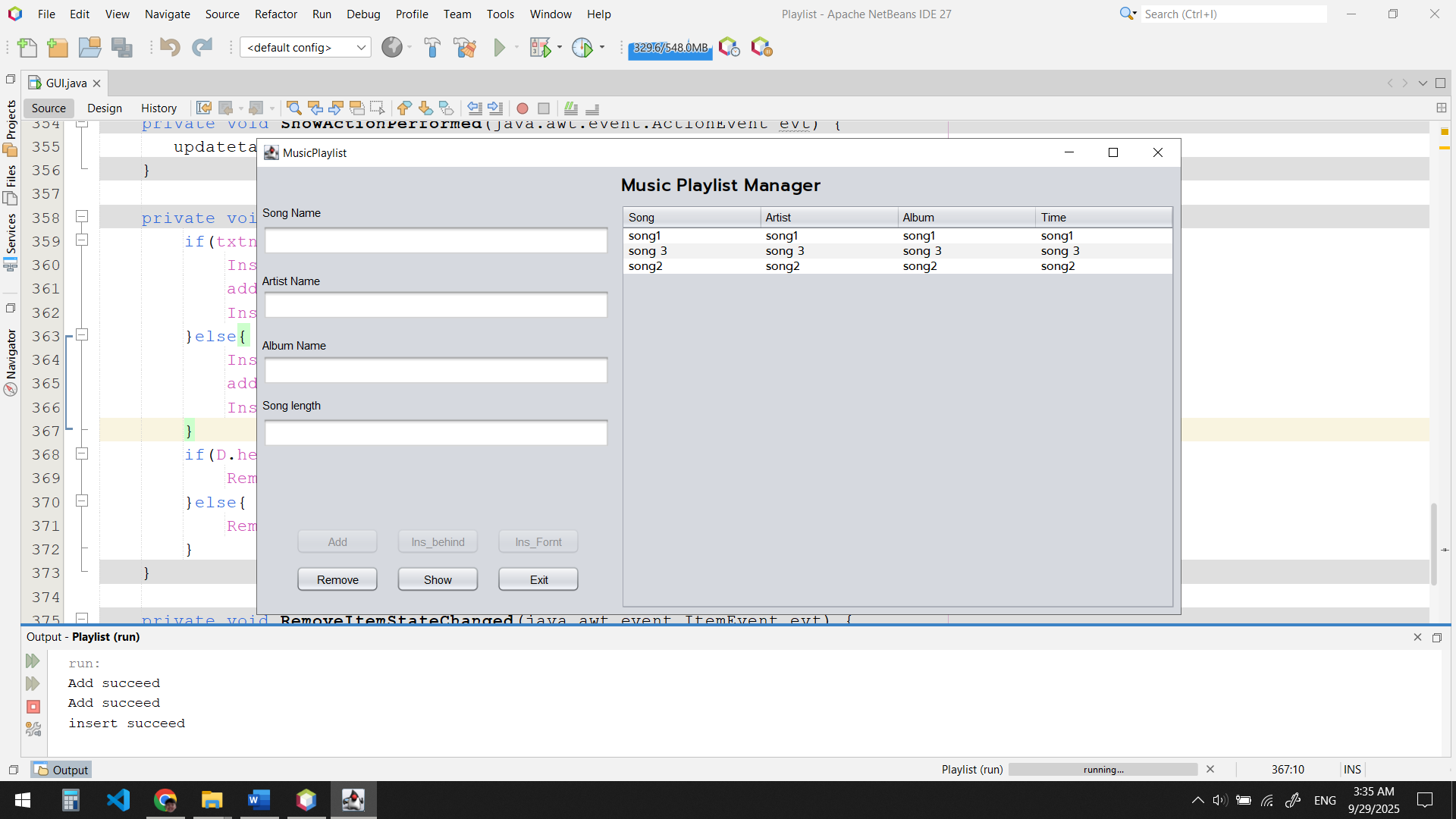
**Add :** 



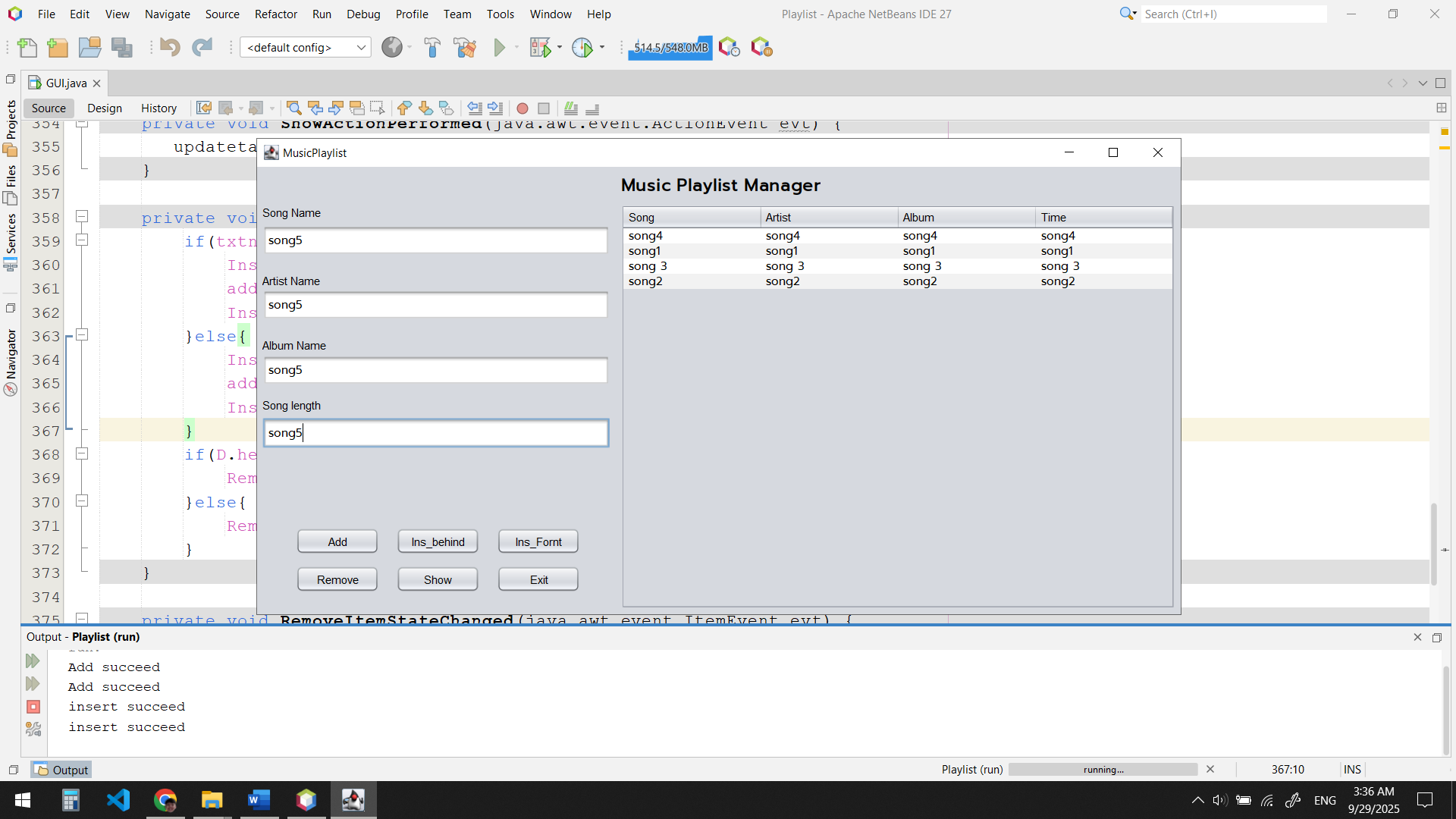
**Insert front:**

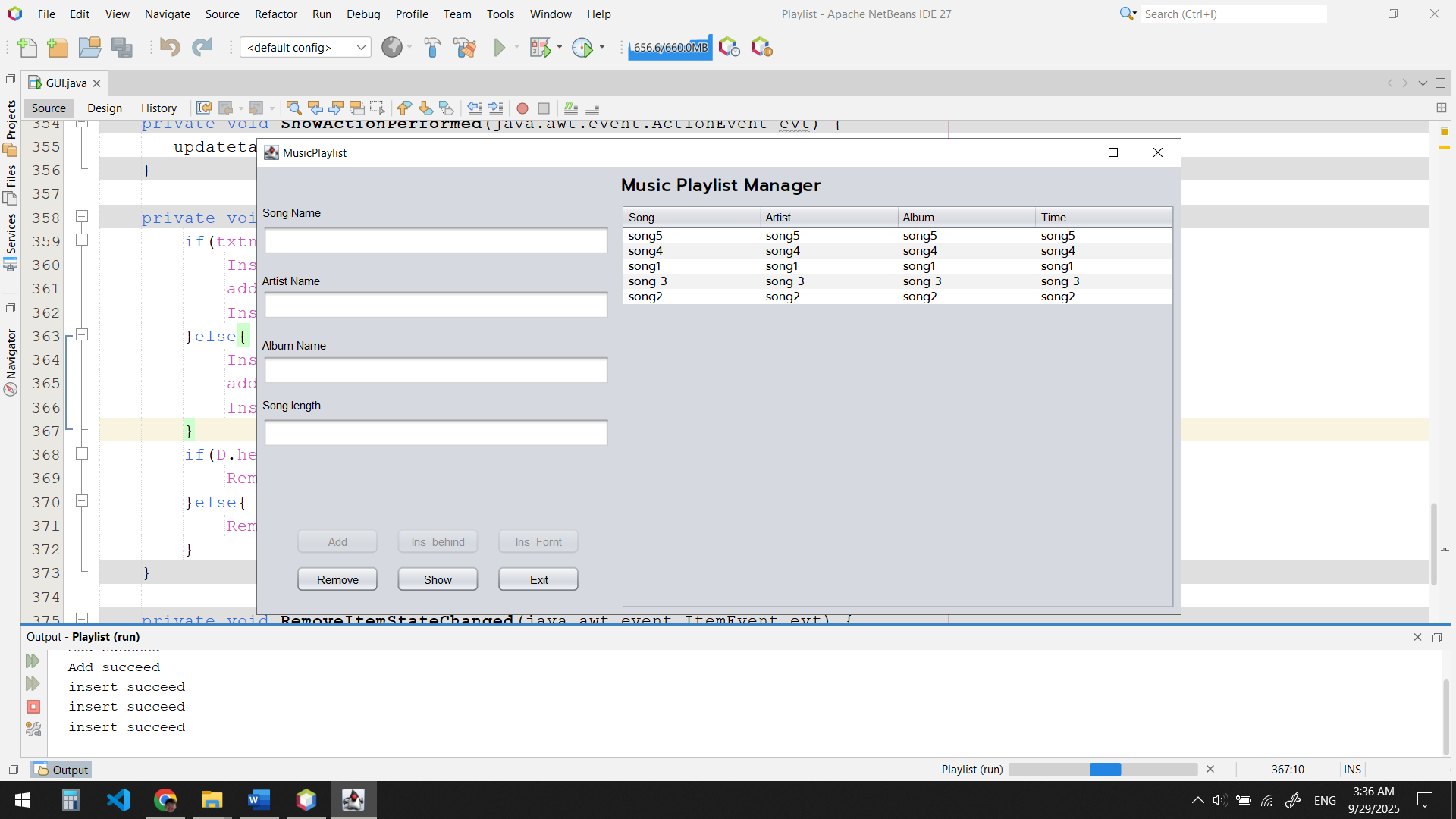
**1: insert front tail**





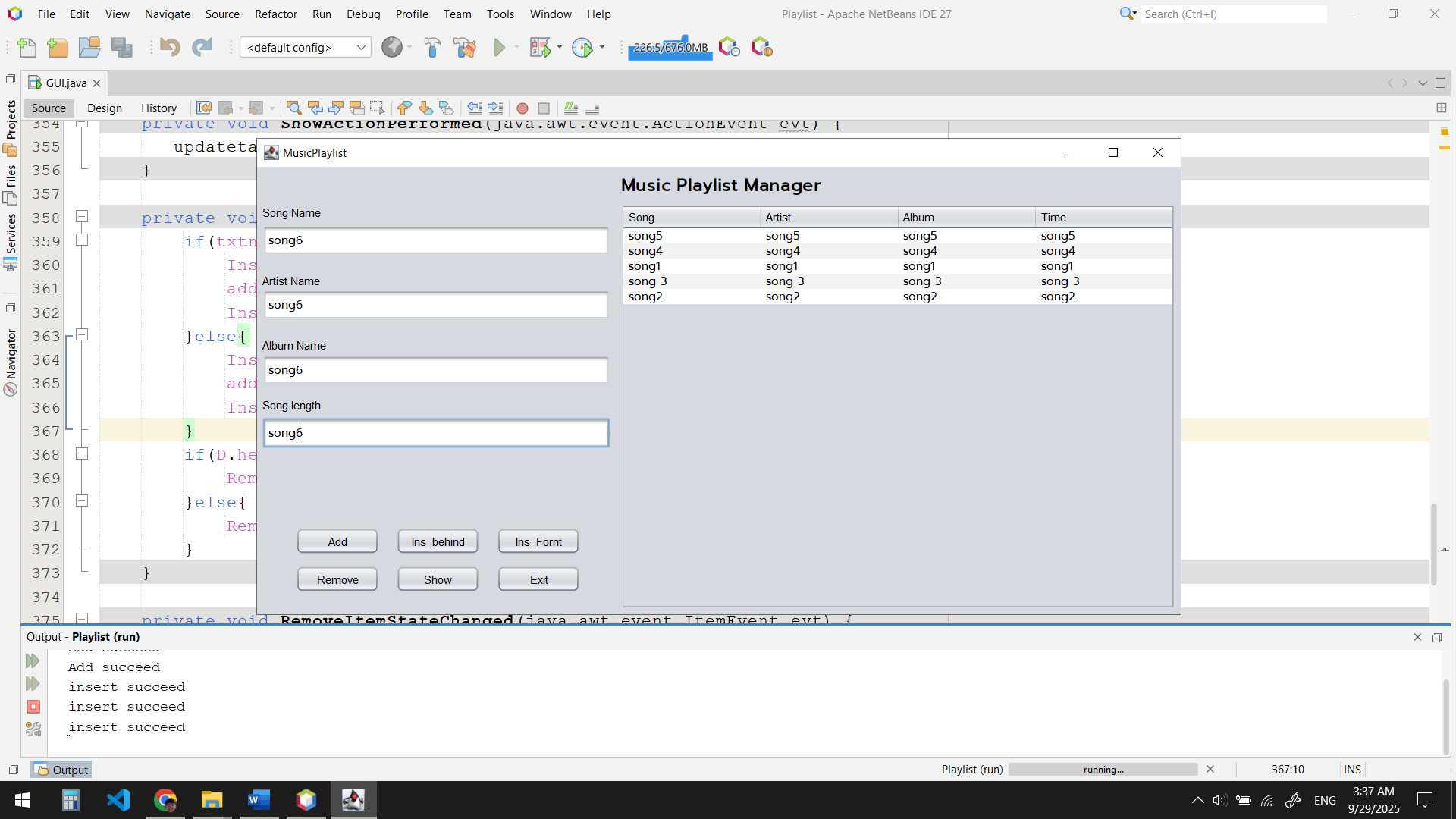
2: insert front head

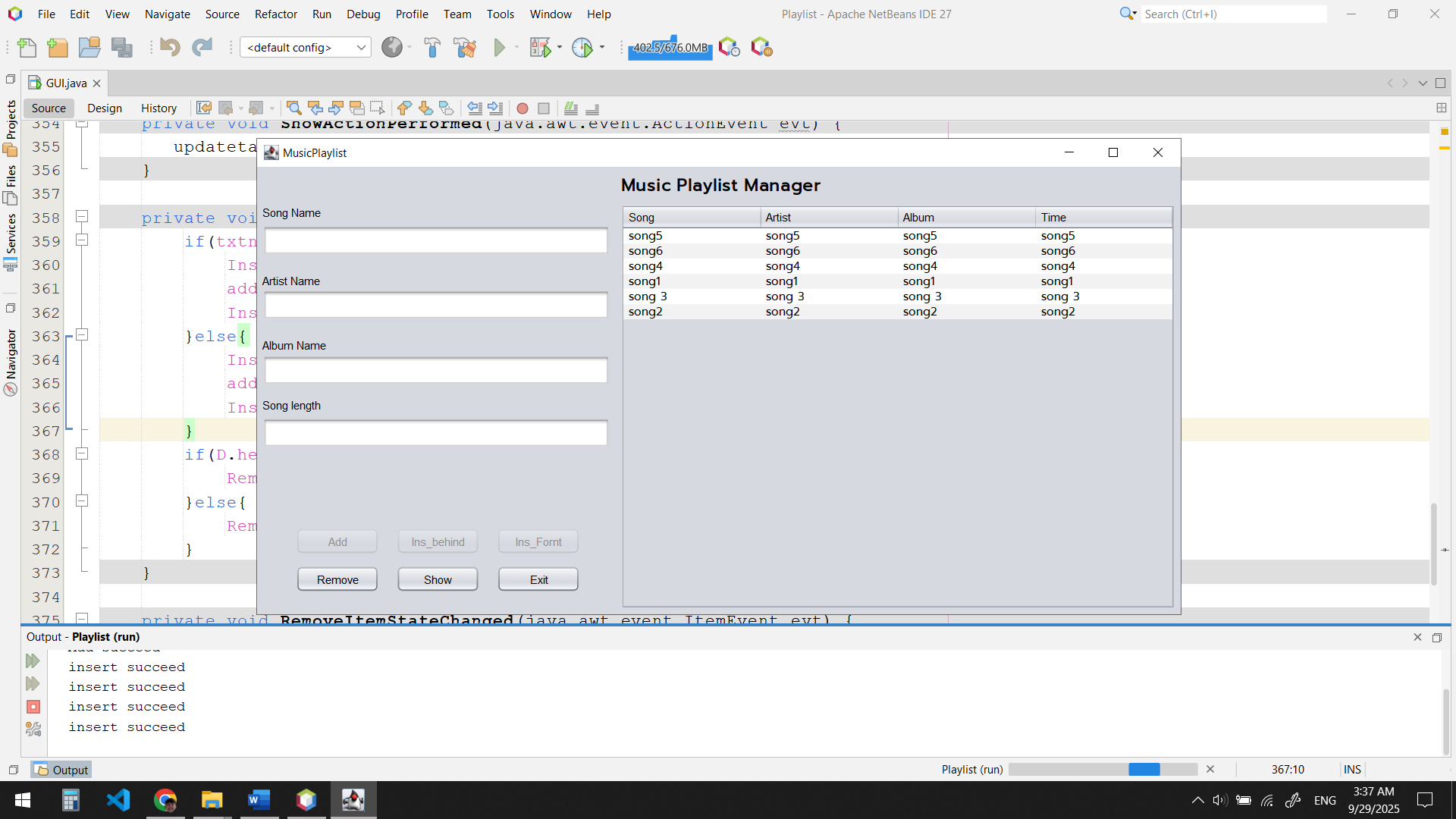




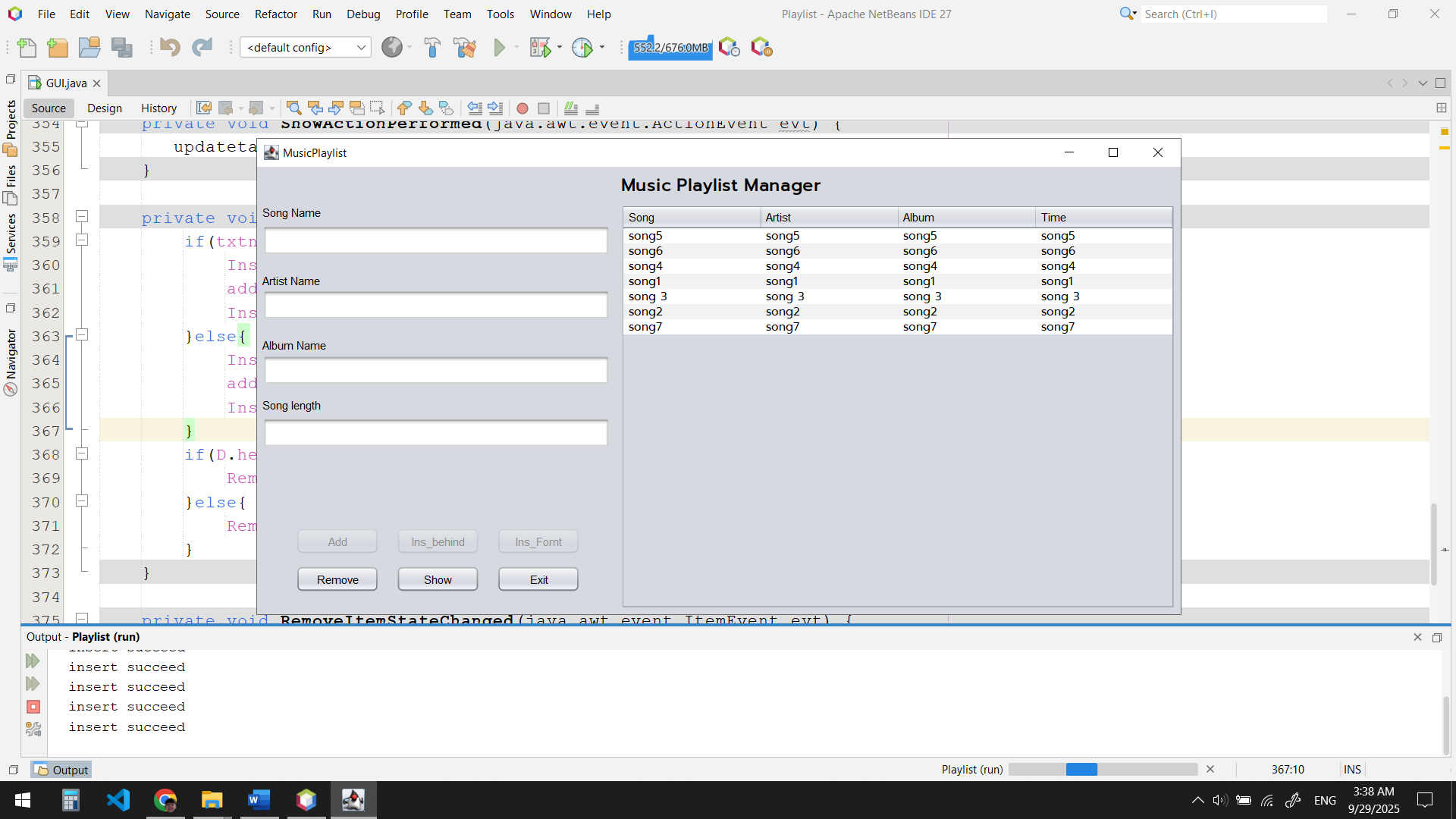
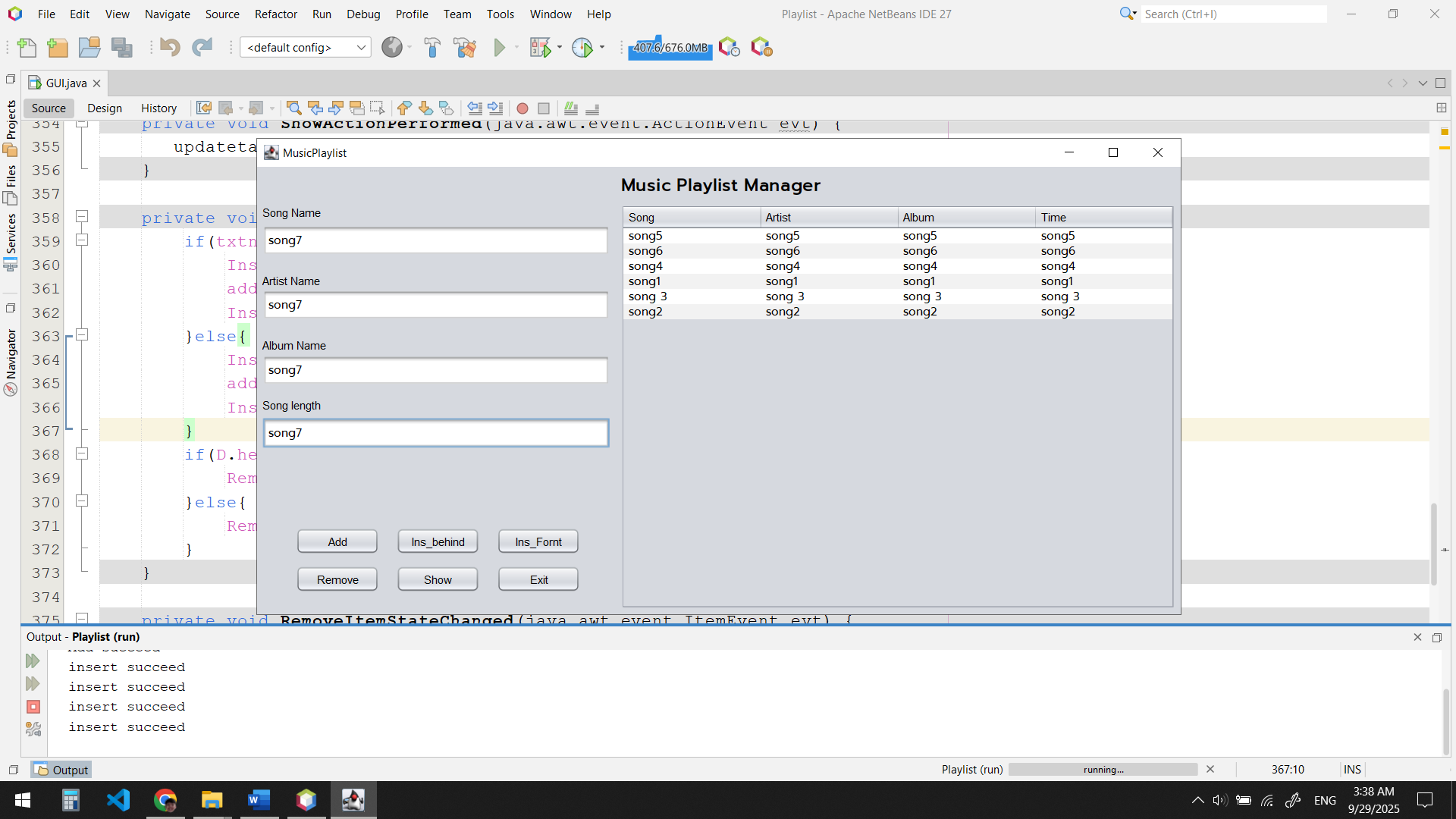
**Insert behind :**

**1: insert behind head**

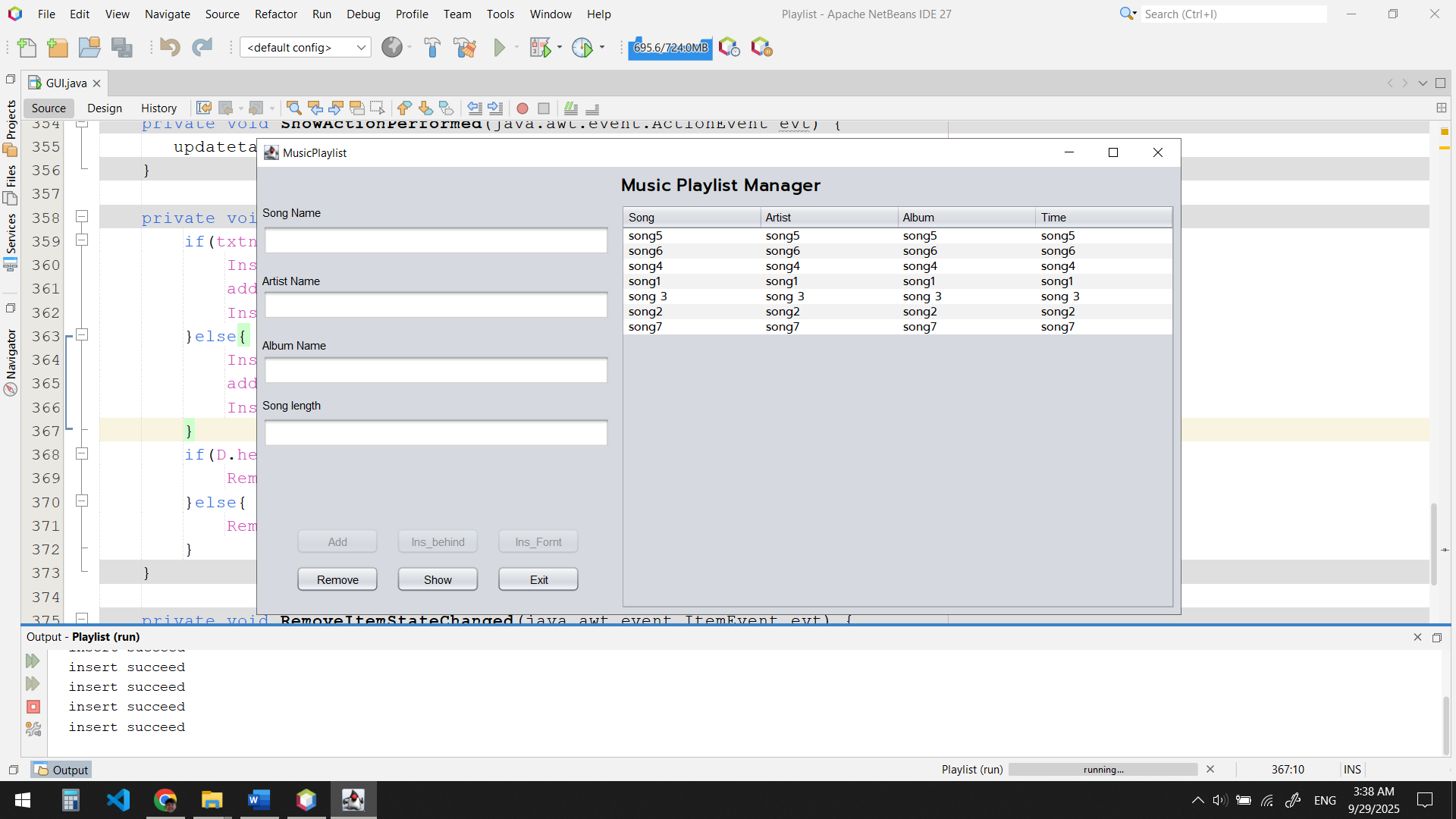


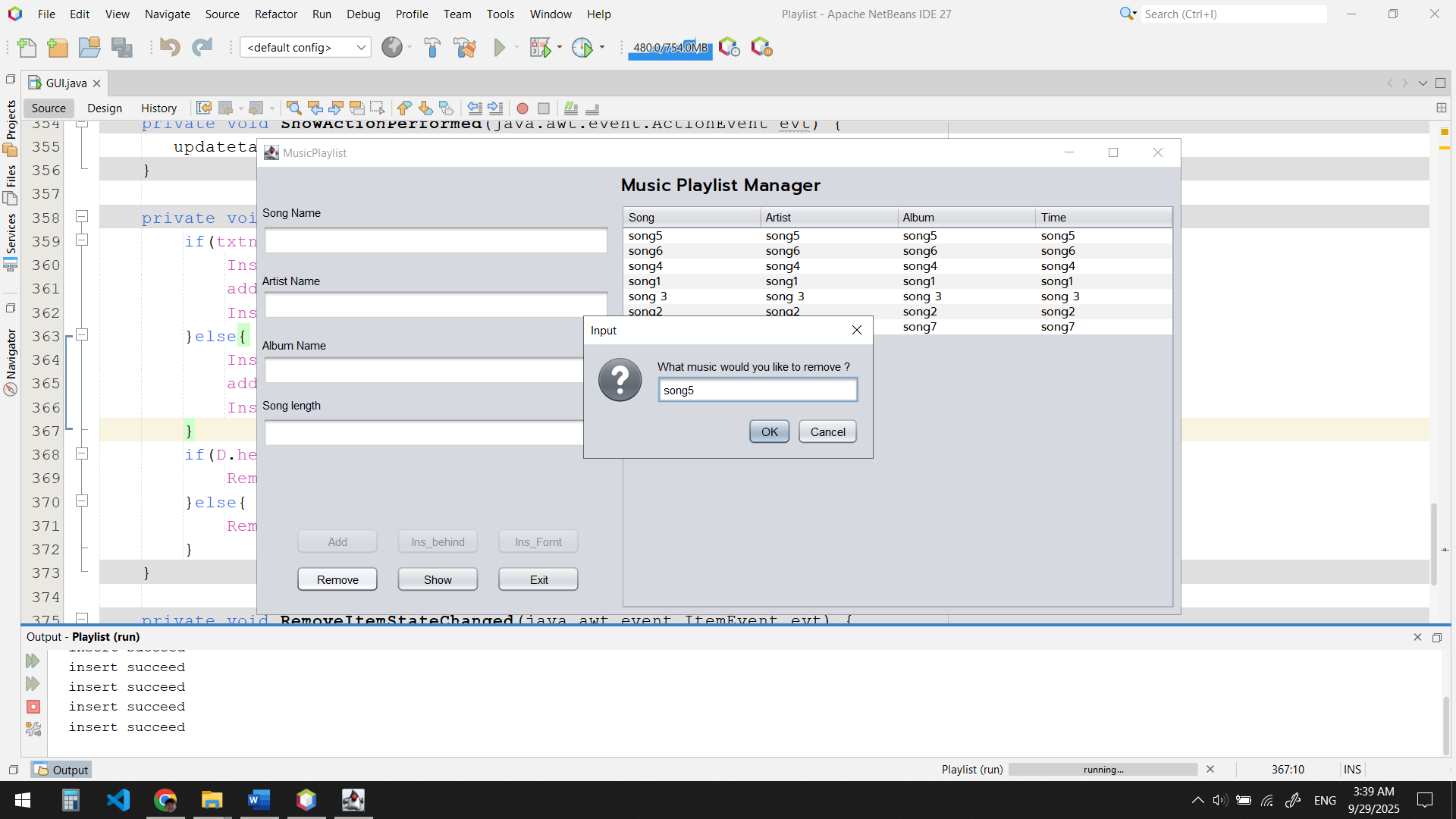


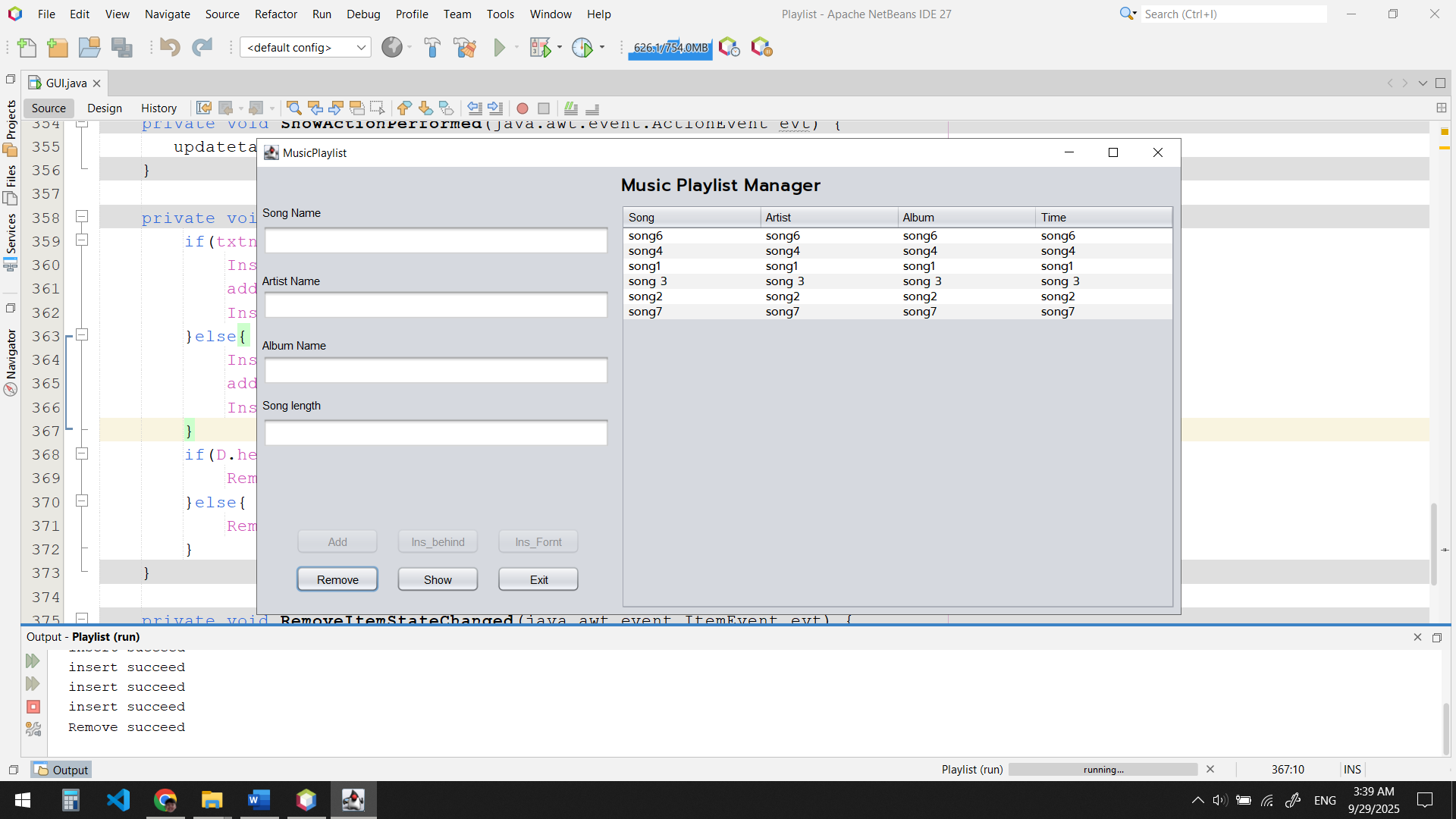
**2: Add behind tail**



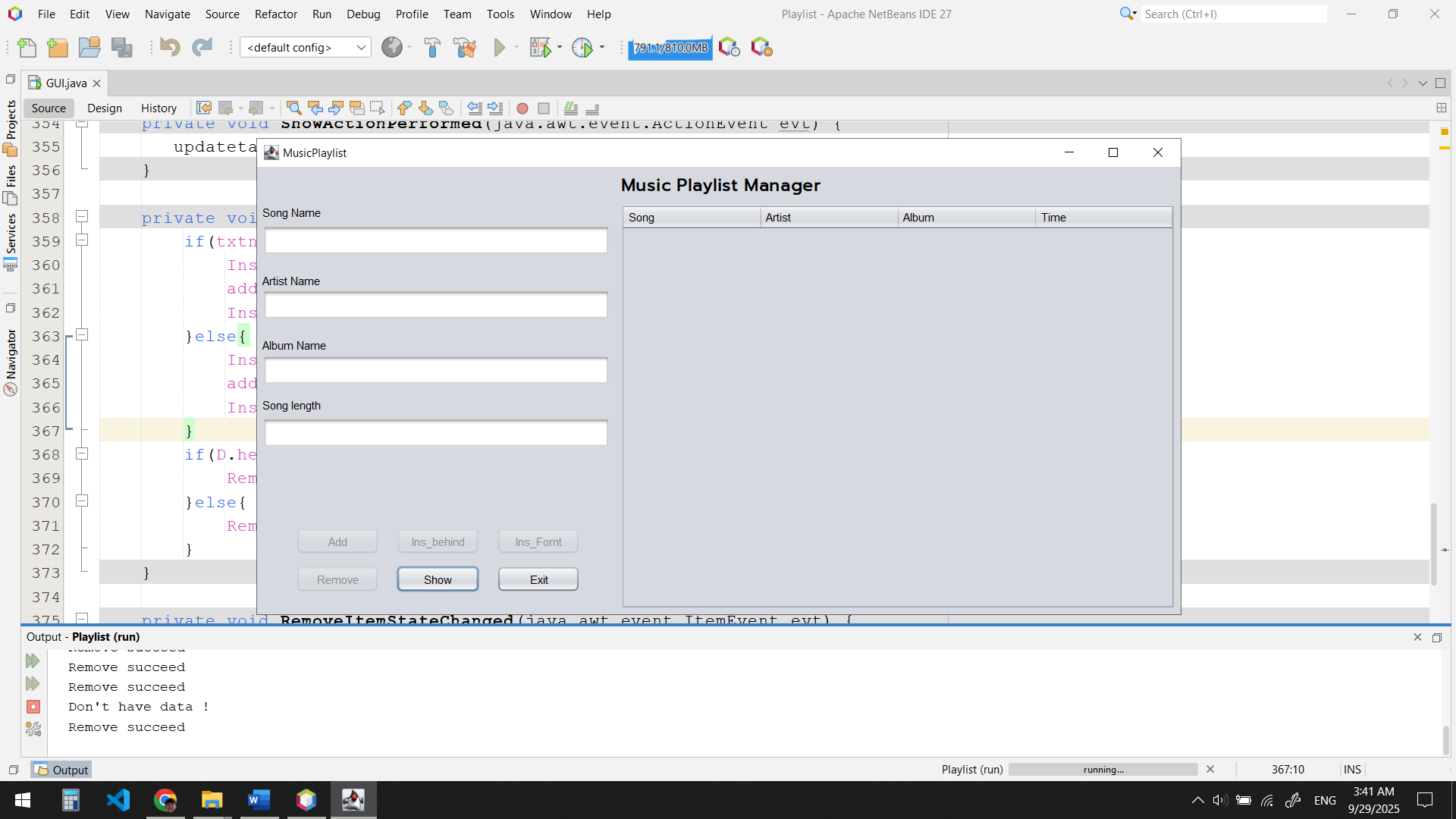
**Remove:**







**Don’t have data to remave :**



\*จะกดปุ่ม remove ไม่ได้ถ้าไม่มีข้องมูลใน list \*

**2: Can't find information**