

A03 DoubleLinkedList

Gruppe 3 / Gruppe C

Klassendiagramm

A02_DoubleLinkedList

Queue <T>

```
private Node<T> first
private Node<T> last
private Node<T> current

public void add(T a)
public void reset()
public void resetToLast()
public Node<T> getFirst()
public Node<T> getLast()
public T next()
public T previous()
public void moveNext()
public void movePrevious()
public T getCurrent() throws CurrentNotSetException
public T get(int pos)
public void remove(int pos)
public void removeCurrent() throws CurrentNotSetException
public void insertAfterCurrentAndMove(T a) throws CurrentNotSetException
```

Node<T>

```
private final T data
private Node<T> next
private Node<T> previous

public Node(T data)
public T getData()
public Node<T> getNext()
public void setNext(Node<T> next)
public Node<T> getPrevious()
public void setPrevious(Node<T> next)
```

<<throws>>

CurrentNotSetException

```
private static final long serialVersionUID
```

Legende:

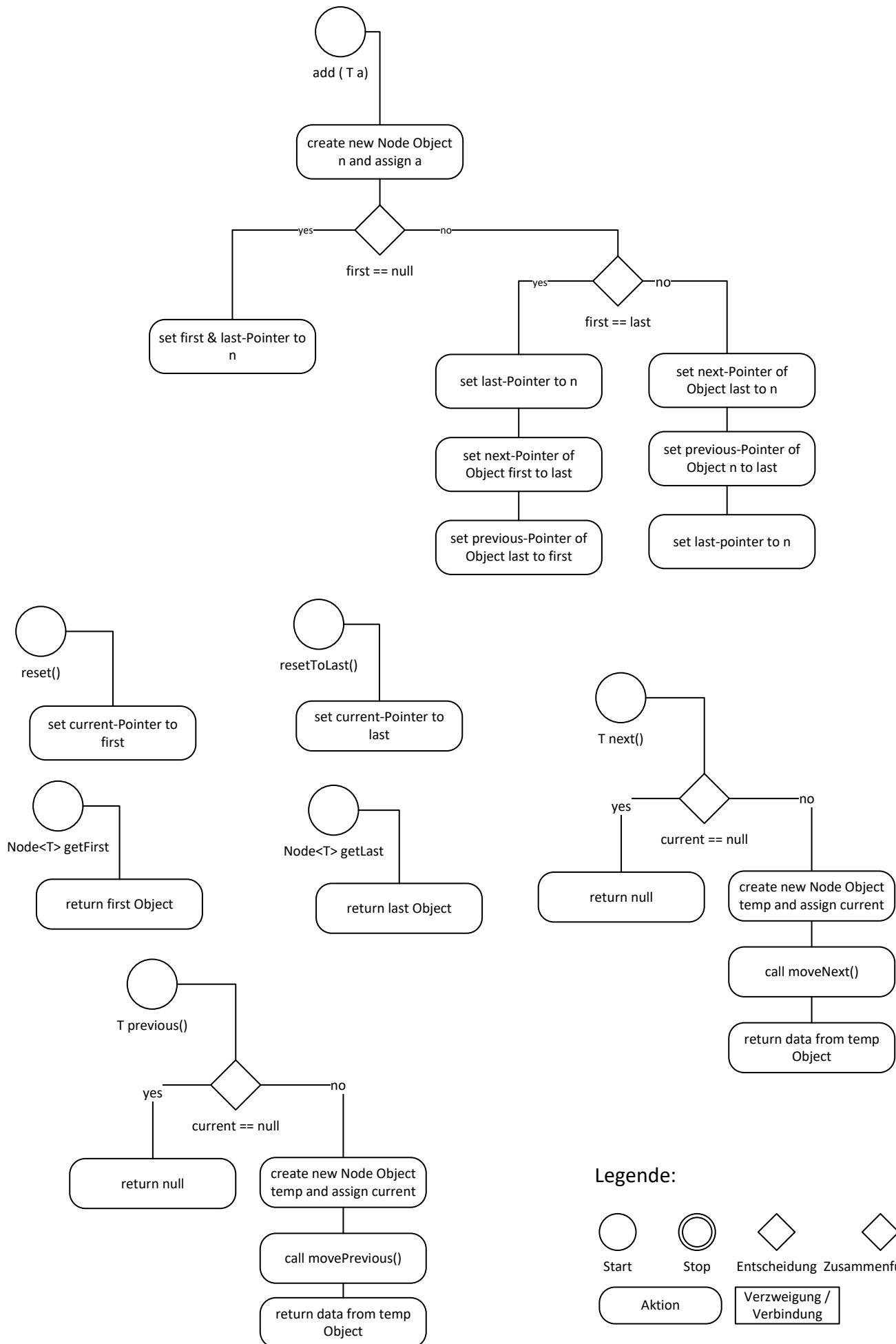
Paketname

Klassenname

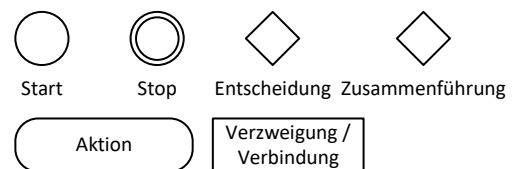
Attribute

Methoden

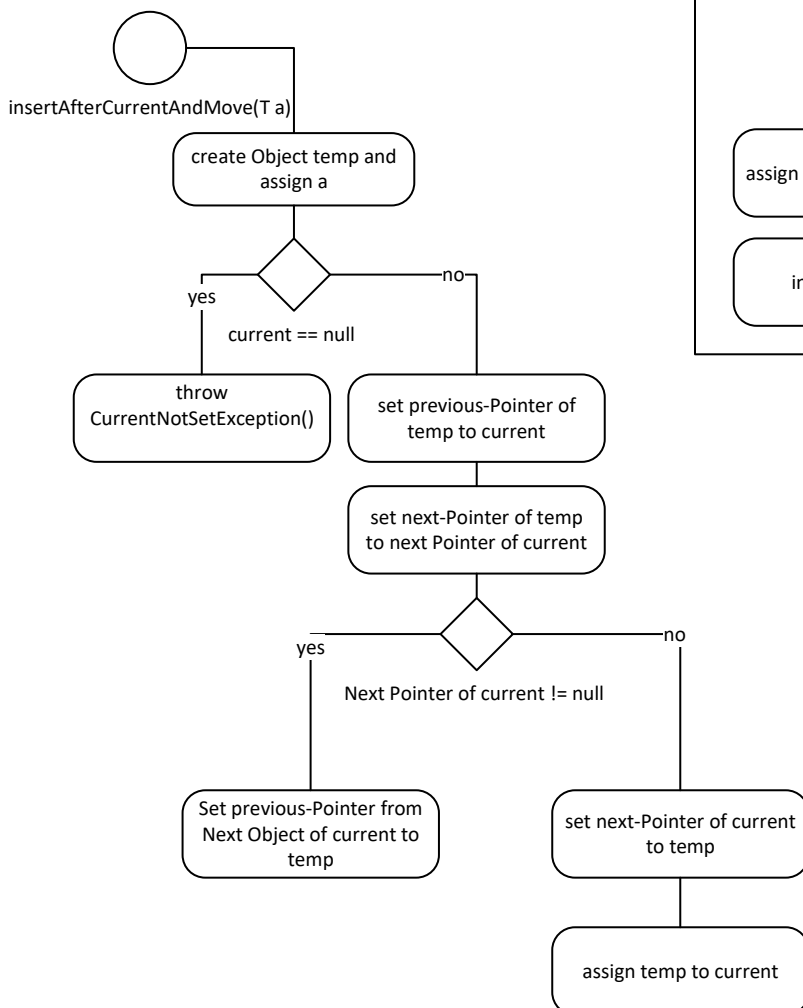
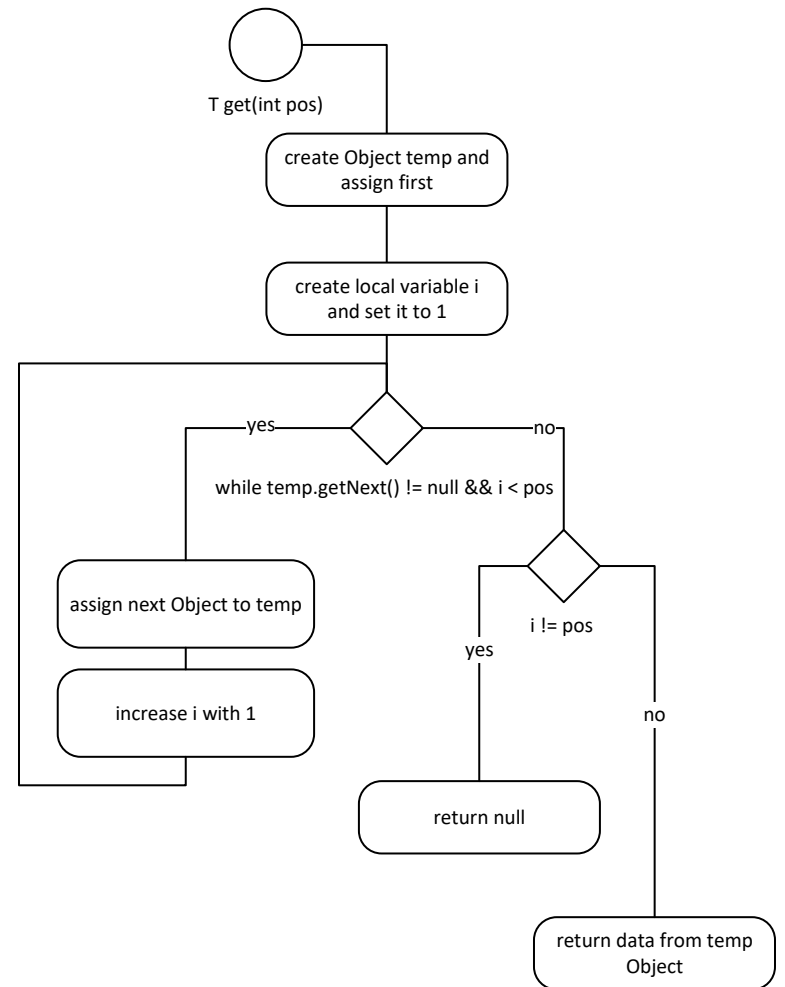
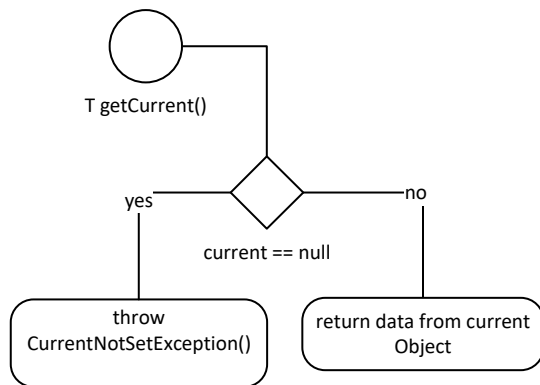
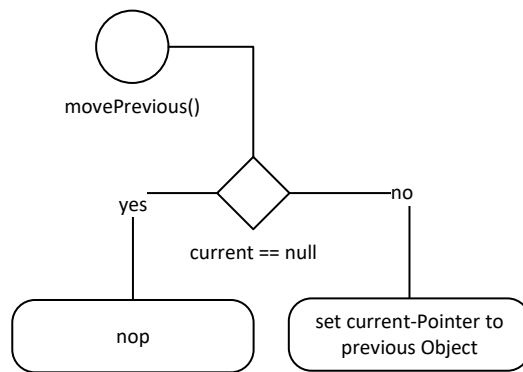
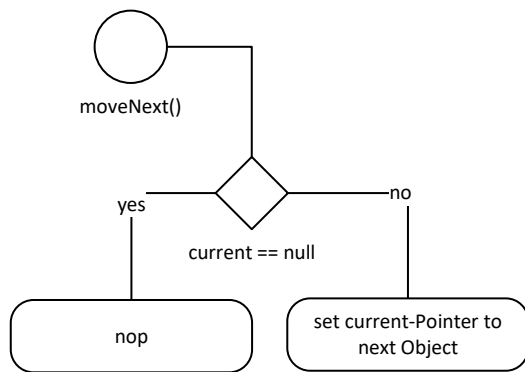
FlowChart 1/4



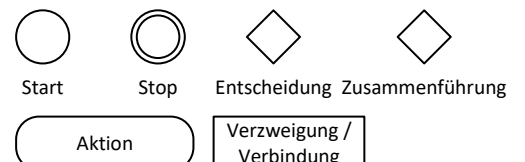
Legende:



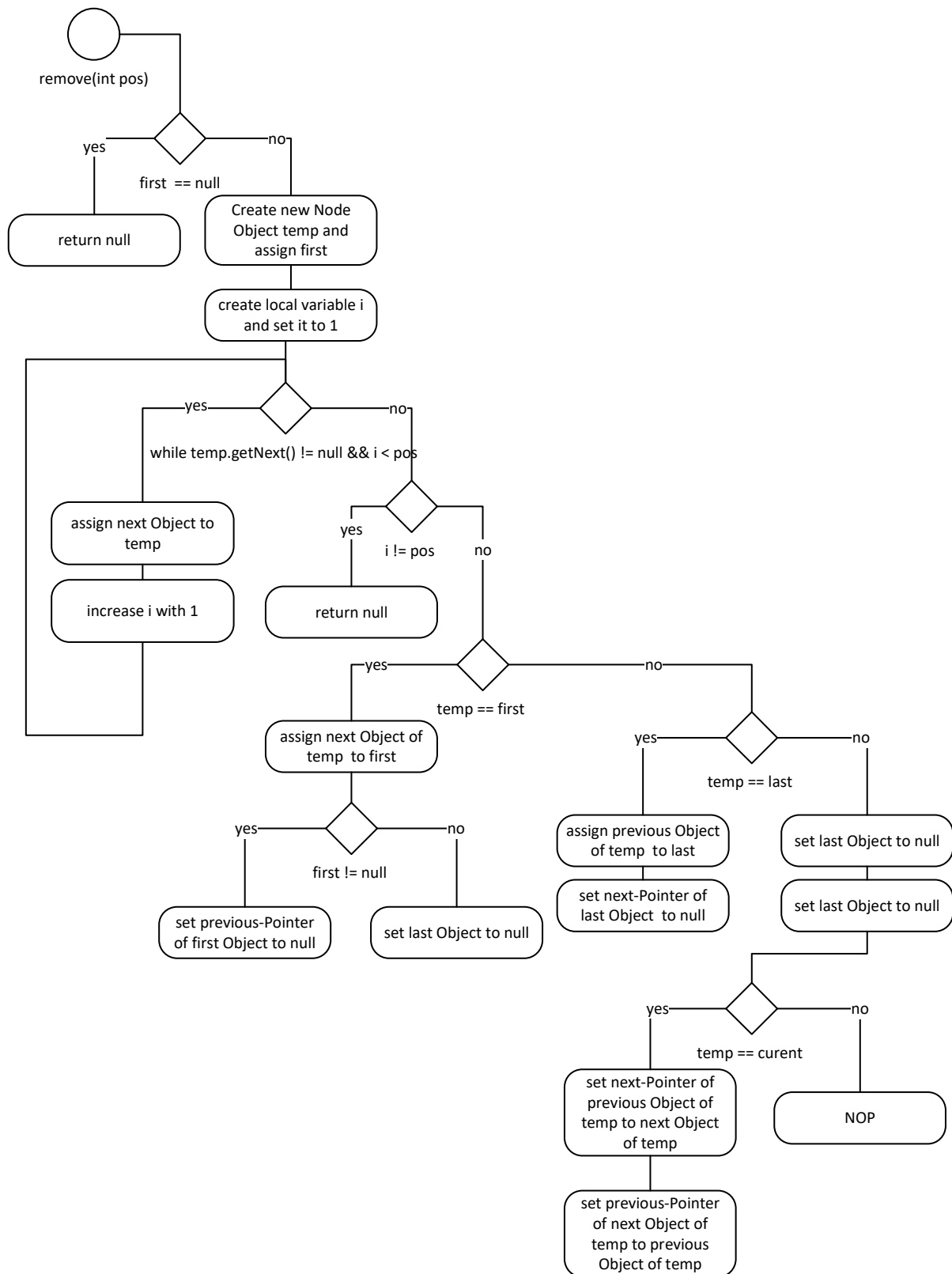
FlowChart 2/4



Legende:



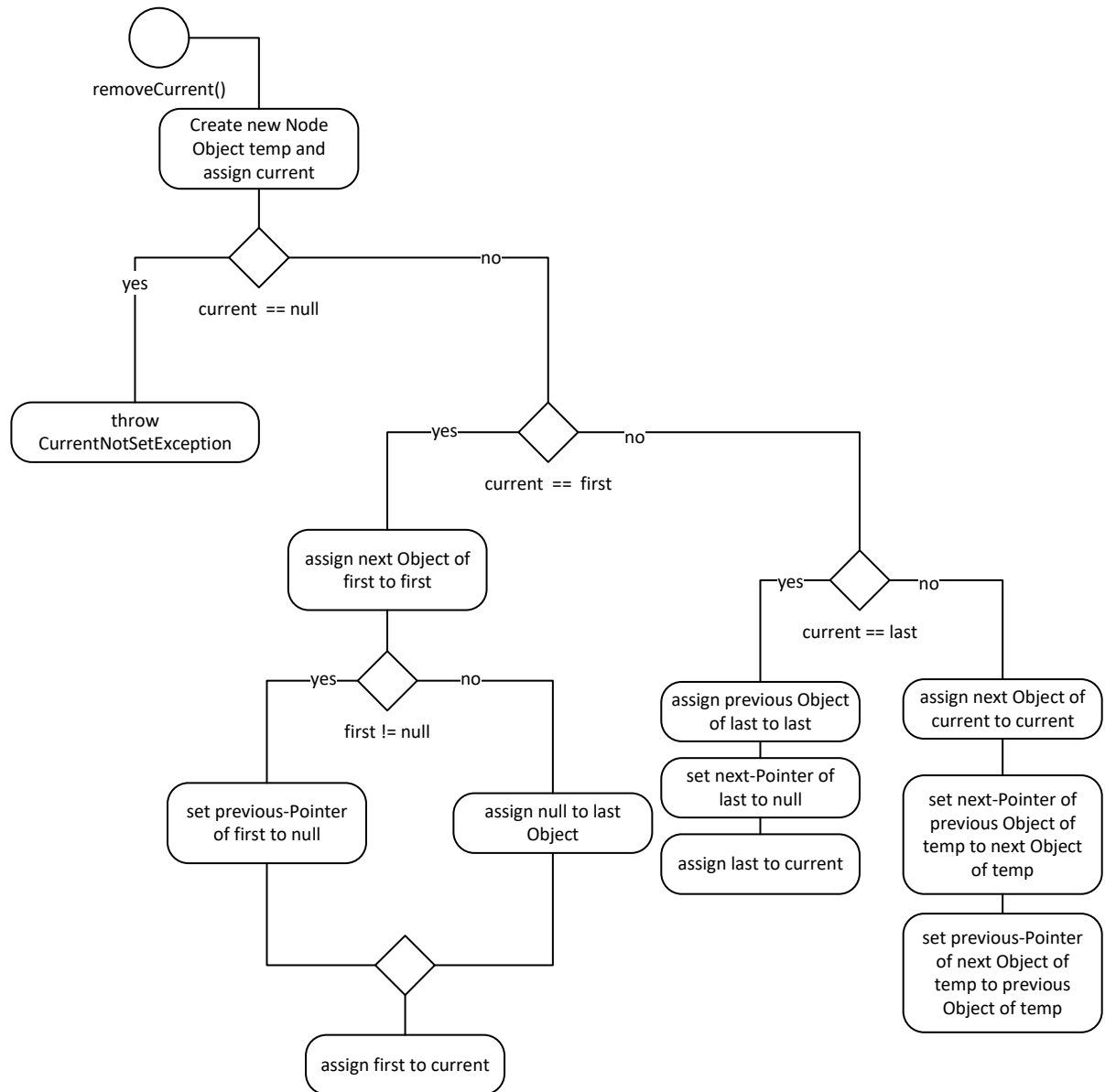
FlowChart 3/4



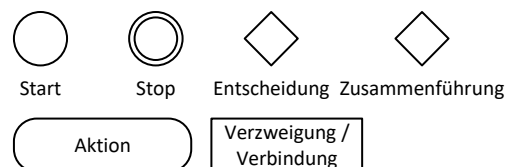
Legende:



FlowChart 4/4



Legende:



Klasse DoubleLinkedList

Methode add(T a) // füge der Liste ein Element hinzu

Erzeuge ein neues Objekt Node n und weise a zu //Instanzieren

Wenn first == null dann // Erstes Element in der Liste

setze first & last auf n

sonst Wenn first==first dann // Wenn ein Element vorhanden ist

setze last mit n

setze den next Pointer von first auf last

setze den previous Pointer von last auf first

Sonst //Element anhängen wenn first und last vorhanden

setze den next Pointer von last auf n

setzte den previous Pointer von n auf last

setzte last mit n

Wenn_ende

Methodenende

Methode reset() // Aktuelles Element auf erstes Element der Liste setzten

setze current auf first

Methodenende

Methode resetToLast() // Aktuelles Element auf letztes Element der Liste setzen

setze current auf last

Methodenende

Methode Node<T> getFirst() // Erstes Objekt der Liste zurückgeben

Return first

Methodenende

Methode Node<T> getLast() // Letztes Objekt der Liste zurückgeben

Return last

Methodenende

Methode T next() // Daten des aktuellen Objekts zurückgeben und current Pointer auf das nachfolgende Element setzen.

Wenn current == null dann gib null zurück Wenn_ende

Neues Node Objekt instanzieren und current zuweisen

moveNext() Funktion aufrufen //setzt den current Pointer auf das nächste Objekt

Daten des Aktuellen Objekts retournieren

Methodenende

Methode T previous() // Daten des aktuellen Objekts zurückgeben und current Pointer auf das vorhergehende Element setzen.

Wenn current == null dann gib null zurück Wenn_ende

Neues Node Objekt instanzieren und current zuweisen

movePrevious() Funktion aufrufen //setzt den current Pointer auf das vorhergehende Objekt

Daten des Aktuellen Objekts retournieren

Methodenende

PseudoCode

Methode moveNext() // current Pointer auf nächstes Element setzten

Wenn current != null dann current ist nächstes Element von current

Methodenende

Methode movePrevious() // current Pointer auf das vorhergehende Element setzten

Wenn current != null dann current ist vorhergehendes Element von current

Methodenende

Methode T getCurrent() // Daten vom aktuellen current retourneren

Wenn current != null dann retourniere null sonst retourniere current data Wenn_End

Methodenende

Methode T get(int pos) // gibt Daten an bestimmter position zurück

Erzeuge neues Node Objekt temp und weise first zu; //starte mit erstem Objekt in der Liste

Lokale Variable i mit 1 initialisieren

Solange next Objekt und pos nicht erreicht //kopiere nächstes Objekt solange bis pos erreicht

speichere next Objekt auf temp

erhöhe Zähler i

Wenn Zähler ungleich pos dann gib null zurück Wenn_End //wenn pos nicht erreicht werden kann gib

Null zurück

Gib daten vom Objekt temp zurück

Methodenende

Methode remove(int pos) // löscht Objekt von current an bestimmter position

Erzeuge neues Node Objekt temp und weise first zu; //starte mit erstem Objekt in der Liste

Lokale Variable i mit 1 initialisieren

Solange next Objekt und pos nicht erreicht //kopiere nächstes Objekt solange bis pos erreicht

speichere next Objekt auf temp

erhöhe Zähler i

Wenn Zähler ungleich pos dann beende die Methode Wenn_End //wenn pos nicht erreicht werden kann

ist kein Element vorhanden -> NOP

Wenn temp == first dann //Zu löschende Element ist das Erste in der Liste

weise first, Objekt nach first zu

Wenn first != null dann // abfrage Ob noch ein Objekt außer first in der Liste

setzte previous Pointer von first auf Null

Sonst

setzte last Objekt auf null

Wenn_End

Sonst Wenn temp == last dann //Zu löschende Element ist das Letzte in der Liste

weise last das Objekt vor temp zu

setzte den next Pointer von last nul

Sonst

Setzte den next Pointer vom Objekt vor temp auf das Element nach temp

Setzte den previous Pointer vom Objekt nach temp auf das Element vor temp

Wenn_End

Wenn temp == current dann setze current null Wenn_End //falls zu löschendes Objekt current ist dann Pointer löschen

Methodenende

PseudoCode

Methode removeCurrent() // löscht Objekt auf das current zeigt

Erzeuge neues Node Objekt temp und weise current zu

Wenn Current == null dann wirf Exception Wenn_Ende

Wenn temp == first dann

 weise first, Objekt nach first zu

 Wenn first != null dann

 setzte previous Pointer von first auf Null

 Sonst

 setzte last Objekt auf null

 Wenn_Ende

 weise dem current Pointer first zu

Sonst Wenn temp == last dann

 weise last das Objekt vor last zu

 setzte den next Pointer von last nul

 weise dem current Pointer last zu

Sonst

 weise dem current Pointer das Objekt nach Current zu

 Setzte den next Pointer vom Objekt vor temp auf das Element nach temp

 Setzte den previous Pointer vom Objekt nach temp auf das Element vor temp

Wenn_Ende

Methodenende

Methode insertAfterCurrentAndRemove(T a) // Ifügt ein Objekt nach current ein

Erzeuge neues Node Objekt temp und instanziiere es mit a

Wenn Current == null dann wirf Exception Wenn_Ende

Setze den previous Pointer von temp auf current

Setze den next Pointer von temp auf Objekt nach current

Wenn current.getNext() != null dann

 setze den Previous Pointer vom Element nach current auf temp

Wenn_Ende

Setze next Pointer von current auf temp

Weise current temp zu

Methodenende