

En stack kan implementeras som en länkad lista. Skapa en klass Stack<T> som kan lagra godtycklig datatyp (det vill säga en generisk Stack). Stacken ska kunna utföra push, pop och peek samt ska man kunna kolla om stacken är tom. Implementera således metoderna:

push

```
public void push(T item)
```

Lägger till ett element överst på stacken.

Parametrar

item-elementet som ska pushas till stacken

Throws

java.lang.NullPointerException- om item är null

pop

```
public T pop()
```

Tar bort det översta elementet på stacken och returnerar det.

Throws

java.util.EmptyStackException- om stacken är tom

Returns

Det översta elementet på stacken

peek

```
public T peek()
```

Kollar på och returnerar det översta objektet på stacken utan att ta bort det

Throws

java.util.EmptyStackException- om stacken är tom

Returns

Det översta elementet på stacken utan att stacken poppas

isEmpty

```
public boolean isEmpty()
```

Kollar om stacken är tom eller ej

Returns

true- om stacken är tom annars *false*

Skapa även klass `Node<E>` som representerar noderna som sparas på stacken. Klassen `Node<E>` ska ha följande metoder:

getNext

```
public Node<E> getNext()
```

Returnerar den nod som denna noden pekar på.

Returns

Nästa nod i nod-kedjan

setNext

```
public void setNext(Node<E> next)
```

Sätter nodens pekare till *next*

Parametrar

next- den nod som ska pekas på

getValue

```
public E getValue()
```

Returnerar värdet för denna noden

Returns

Värdet för denna noden

setValue

```
public void setValue(E value)
```

Sätter värdet på denna noden till *value*

Parametrar

value-värdet som ska sättas för denna noden

Tips:

För en stack behövs bara en referens *top* till den översta noden. Sedan kommer man åt de andra genom att följa *next-pekarna*