

Tutorium Programmieren

Tut Nr.7: Pakete, Listen

Michael Friedrich | 10. / 12.12.2013

INSTITUT FÜR THEORETISCHE INFORMATIK



- 1 Anmerkungen
- 2 Pakete
- 3 Listen
 - Tutoriumsaufgabe
 - doppelt verkettet

Meldet euch möglichst bald an

Spätestens bis zum 23.12.2013

■ Vielen Dank nochmals

■ Verbesserungspunkte

- Fehler in den Folien → JA, aber stören idR den Hauptaussage der Folie nicht. Wir sind auch nur Menschen ;)
- Mehr zusammen coden
 - JAIN, Probleme dabei sind fehlende initiative
 - Jede Woche auseinandersetzen mit Umsetzungsschwierigkeiten
 - step-by-step durch die Lösung meiner Meinung nach besser als dauernd zuhören (Bsp. letztes Tut)

■ Vielen Dank nochmals

■ Verbesserungspunkte

- Fehler in den Folien → JA, aber stören idR den Hauptaussage der Folie nicht. Wir sind auch nur Menschen ;)
- Mehr zusammen coden
 - JAIN, Probleme dabei sind fehlende initiative
 - Jede Woche auseinandersetzen mit Umsetzungsschwierigkeiten
 - step-by-step durch die Lösung meiner Meinung nach besser als dauernd zuhören (Bsp. letztes Tut)

- Vielen Dank nochmals

- Verbesserungspunkte
 - Fehler in den Folien → JA, aber stören idR den Hauptaussage der Folie nicht. Wir sind auch nur Menschen ;)
 - Mehr zusammen coden
 - JAIN, Probleme dabei sind fehlende initiative
 - Jede Woche auseinandersetzen mit Umsetzungsschwierigkeiten
 - step-by-step durch die Lösung meiner Meinung nach besser als dauernd zuhören (Bsp. letztes Tut)

- Vielen Dank nochmals

- Verbesserungspunkte
 - Fehler in den Folien → JA, aber stören idR den Hauptaussage der Folie nicht. Wir sind auch nur Menschen ;)
 - Mehr zusammen coden
 - JAIN, Probleme dabei sind fehlende initiative
 - Jede Woche auseinandersetzen mit Umsetzungsschwierigkeiten
 - step-by-step durch die Lösung meiner Meinung nach besser als dauernd zuhören (Bsp. letztes Tut)

- Vielen Dank nochmals

- Verbesserungspunkte
 - Fehler in den Folien → JA, aber stören idR den Hauptaussage der Folie nicht. Wir sind auch nur Menschen ;)
 - Mehr zusammen coden
 - JAIN, Probleme dabei sind fehlende initiative
 - Jede Woche auseinandersetzen mit Umsetzungsschwierigkeiten
 - step-by-step durch die Lösung meiner Meinung nach besser als dauernd zuhören (Bsp. letztes Tut)

- Vielen Dank nochmals

- Verbesserungspunkte
 - Fehler in den Folien → JA, aber stören idR den Hauptaussage der Folie nicht. Wir sind auch nur Menschen ;)
 - Mehr zusammen coden
 - JAIN, Probleme dabei sind fehlende initiative
 - Jede Woche auseinandersetzen mit Umsetzungsschwierigkeiten
 - step-by-step durch die Lösung meiner Meinung nach besser als dauernd zuhören (Bsp. letztes Tut)

- Vielen Dank nochmals

- Verbesserungspunkte
 - Fehler in den Folien → JA, aber stören idR den Hauptaussage der Folie nicht. Wir sind auch nur Menschen ;)
 - Mehr zusammen coden
 - JAIN, Probleme dabei sind fehlende initiative
 - Jede Woche auseinandersetzen mit Umsetzungsschwierigkeiten
 - step-by-step durch die Lösung meiner Meinung nach besser als dauernd zuhören (Bsp. letztes Tut)

- Vielen Dank nochmals

- Verbesserungspunkte
 - Fehler in den Folien → JA, aber stören idR den Hauptaussage der Folie nicht. Wir sind auch nur Menschen ;)
 - Mehr zusammen coden
 - JAIN, Probleme dabei sind fehlende initiative
 - Jede Woche auseinandersetzen mit Umsetzungsschwierigkeiten
 - step-by-step durch die Lösung meiner Meinung nach besser als dauernd zuhören (Bsp. letztes Tut)

- Struktur, um Code zu ordnen
- bekanntes Beispiel: Libraries

Beispiel

```
package garage;  
  
public class Bike {  
    ...  
}
```

Wird in der Regel auf dem Übungsblatt vorgegeben.

- Struktur, um Code zu ordnen
- bekanntes Beispiel: Libraries

Beispiel

```
package garage;  
  
public class Bike {  
    ...  
}
```

Wird in der Regel auf dem Übungsblatt vorgegeben.

- Struktur, um Code zu ordnen
- bekanntes Beispiel: Libraries

Beispiel

```
package garage;  
  
public class Bike {  
    ...  
}
```

Wird in der Regel auf dem Übungsblatt vorgegeben.

- Struktur, um Code zu ordnen
- bekanntes Beispiel: Libraries

Beispiel

```
package garage;  
  
public class Bike {  
    ...  
}
```

Wird in der Regel auf dem Übungsblatt vorgegeben.

- Struktur, um Code zu ordnen
- bekanntes Beispiel: Libraries

Beispiel

```
package garage;  
  
public class Bike {  
    ...  
}
```

Wird in der Regel auf dem Übungsblatt vorgegeben.

- Basieren auf der Objektreferenz
- besteht aus Nodes mit Wert Referenz auf nächsten Node

Beispiel

```
class ListNode{  
    String content;  
    ListNode next;  
  
    ListNode(String x, ListNode n) {  
        this.content = x;  
        this.next = n;  
    }  
}
```

- Basieren auf der Objektreferenz
- besteht aus Nodes mit Wert Referenz auf nächsten Node

Beispiel

```
class ListNode{  
    String content;  
    ListNode next;  
  
    ListNode(String x, ListNode n) {  
        this.content = x;  
        this.next = n;  
    }  
}
```

- Basieren auf der Objektreferenz
- besteht aus Nodes mit Wert Referenz auf nächsten Node

Beispiel

```
class ListNode{  
    String content;  
    ListNode next;  
  
    ListNode(String x, ListNode n) {  
        this.content = x;  
        this.next = n;  
    }  
}
```

- Basieren auf der Objektreferenz
- besteht aus Nodes mit Wert Referenz auf nächsten Node

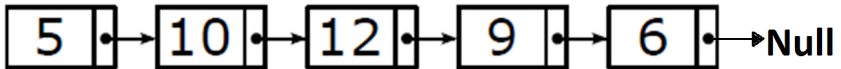
Beispiel

```
class ListNode{  
    String content;  
    ListNode next;  
  
    ListNode(String x, ListNode n) {  
        this.content = x;  
        this.next = n;  
    }  
}
```

- Basieren auf der Objektreferenz
- besteht aus Nodes mit Wert Referenz auf nächsten Node

Beispiel

```
class ListNode{  
    String content;  
    ListNode next;  
  
    ListNode(String x, ListNode n) {  
        this.content = x;  
        this.next = n;  
    }  
}
```



siehe pdf

Jemand eine Idee, wie das realisierbar ist?

Beispiel

```
class DNode {  
    Object content;  
    DNode next;  
    DNode prev;  
  
    DNode(content, DNode next, DNode prev) {  
        this.content = content;  
        this.next = next;  
        this.prev = prev;  
    }  
}
```

Jemand eine Idee, wie das realisierbar ist?

Beispiel

```
class DNode {  
    Object content;  
    DNode next;  
    DNode prev;  
  
    DNode(content, DNode next, DNode prev) {  
        this.content = content;  
        this.next = next;  
        this.prev = prev;  
    }  
}
```


Jemand eine Idee, wie das realisierbar ist?

Beispiel

```
class DNode {  
    Object content;  
    DNode next;  
    DNode prev;  
  
    DNode(content, DNode next, DNode prev) {  
        this.content = content;  
        this.next = next;  
        this.prev = prev;  
    }  
}
```