

# SBB



```
class Einheit {
    String typ;
    Einheit folgend;
    Einheit(String t, Einheit f){
        typ = t; folgend = f;
    }
}
```

## Was gibt folgender Code aus?

```
Einheit e = new Einheit ("Lok",
                        new Einheit ("W1",
                                    new Einheit("W2", null)));
e = new Einheit("W3", e);
e.folgend.typ = "Speisewagen";
while (e != null){
    Out.print(e.typ + " ");
    e = e.folgend;
}
```

- (1)Lok Speisewagen W1 W2
- (2)W3 Speisewagen W2 W1
- (3)W3 Speisewagen W1 W2
- (4)Lok W1 Speisewagen W2
- (5)Lok W2 Speisewagen W1
- (6)W1 W2 W3 Speisewagen Lok
- (7)Gar nichts
- (8)Terminiert nicht

# SBB



```
class Einheit {
    String typ;
    Einheit folgend;
    Einheit(String t, Einheit f){
        typ = t; folgend = f;
    }
}
```

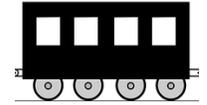
## Was gibt folgender Code aus?

```
Einheit e = new Einheit ("Lok",
                        new Einheit ("W1",
                                    new Einheit("W2", null)));
e = new Einheit("W3", e);
e.folgend.typ = "Speisewagen";
while (e != null){
    Out.print(e.typ + " ");
    e = e.folgend;
}
```

# SBB



```
class Einheit {
    String typ;
    Einheit folgend;
    Einheit(String t, Einheit f){
        typ = t; folgend = f;
    }
}
```



W2

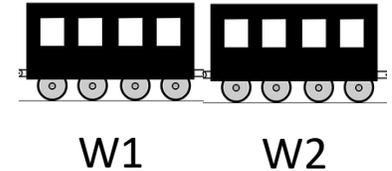
## Was gibt folgender Code aus?

```
Einheit e = new Einheit ("Lok",
                        new Einheit ("W1",
                                    new Einheit("W2", null)));
e = new Einheit("W3", e);
e.folgend.typ = "Speisewagen";
while (e != null){
    Out.print(e.typ + " ");
    e = e.folgend;
}
```

# SBB



```
class Einheit {
    String typ;
    Einheit folgend;
    Einheit(String t, Einheit f){
        typ = t; folgend = f;
    }
}
```



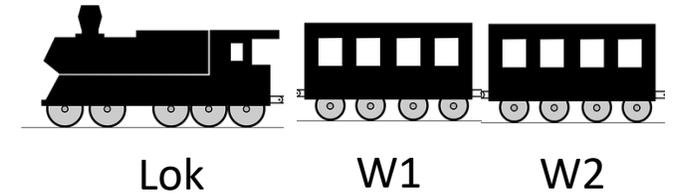
## Was gibt folgender Code aus?

```
Einheit e = new Einheit ("Lok",
                        new Einheit ("W1",
                                      new Einheit("W2", null)));
e = new Einheit("W3", e);
e.folgend.typ = "Speisewagen";
while (e != null){
    Out.print(e.typ + " ");
    e = e.folgend;
}
```

# SBB



```
class Einheit {
    String typ;
    Einheit folgend;
    Einheit(String t, Einheit f){
        typ = t; folgend = f;
    }
}
```



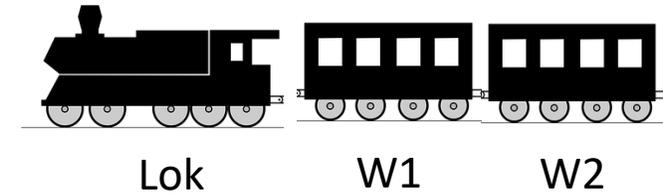
## Was gibt folgender Code aus?

```
Einheit e = new Einheit ("Lok",
                        new Einheit ("W1",
                                    new Einheit("W2", null)));
e = new Einheit("W3", e);
e.folgend.typ = "Speisewagen";
while (e != null){
    Out.print(e.typ + " ");
    e = e.folgend;
}
```

# SBB



```
class Einheit {
    String typ;
    Einheit folgend;
    Einheit(String t, Einheit f){
        typ = t; folgend = f;
    }
}
```



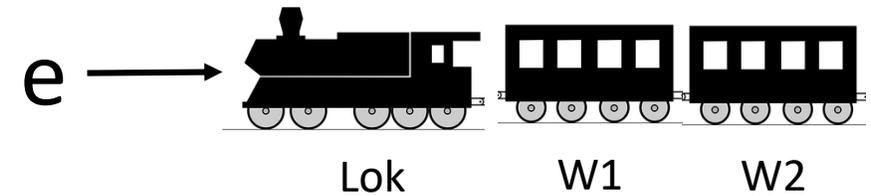
## Was gibt folgender Code aus?

```
Einheit e = new Einheit ("Lok",
                        new Einheit ("W1",
                                    new Einheit("W2", null)));
e = new Einheit("W3", e);
e.folgend.typ = "Speisewagen";
while (e != null){
    Out.print(e.typ + " ");
    e = e.folgend;
}
```

# SBB



```
class Einheit {  
    String typ;  
    Einheit folgend;  
    Einheit(String t, Einheit f){  
        typ = t; folgend = f;  
    }  
}
```



## Was gibt folgender Code aus?

```
Einheit e = new Einheit ("Lok",  
                        new Einheit ("W1",  
                                    new Einheit("W2", null)));  
e = new Einheit("W3", e);  
e.folgend.typ = "Speisewagen";  
while (e != null){  
    Out.print(e.typ + " ");  
    e = e.folgend;  
}
```

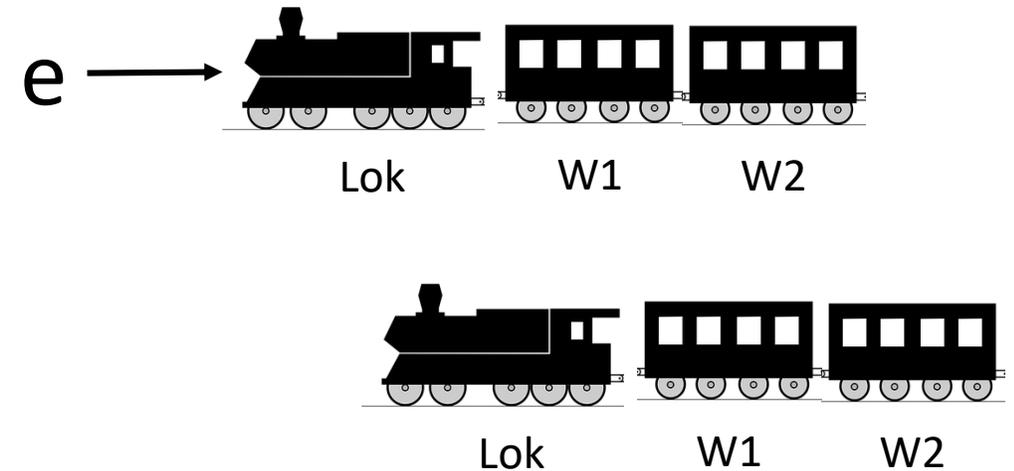
# SBB



```
class Einheit {  
    String typ;  
    Einheit folgend;  
    Einheit(String t, Einheit f){  
        typ = t; folgend = f;  
    }  
}
```

## Was gibt folgender Code aus?

```
Einheit e = new Einheit ("Lok",  
                        new Einheit ("W1",  
                                    new Einheit("W2", null)));  
e = new Einheit("W3", e);  
e.folgend.typ = "Speisewagen";  
while (e != null){  
    Out.print(e.typ + " ");  
    e = e.folgend;  
}
```



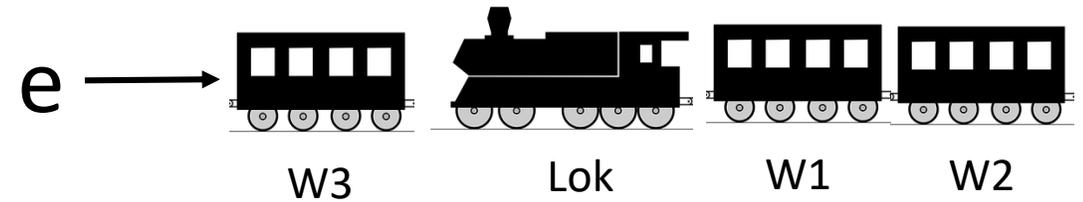
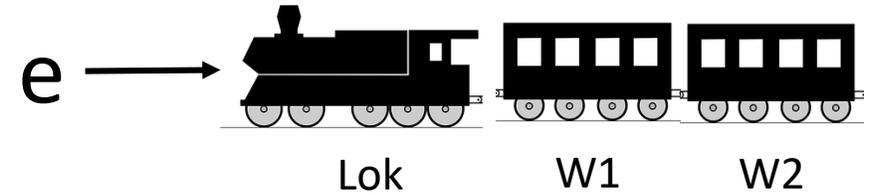
# SBB



```
class Einheit {
    String typ;
    Einheit folgend;
    Einheit(String t, Einheit f){
        typ = t; folgend = f;
    }
}
```

## Was gibt folgender Code aus?

```
Einheit e = new Einheit ("Lok",
                        new Einheit ("W1",
                                    new Einheit("W2", null)));
e = new Einheit("W3", e);
e.folgend.typ = "Speisewagen";
while (e != null){
    Out.print(e.typ + " ");
    e = e.folgend;
}
```



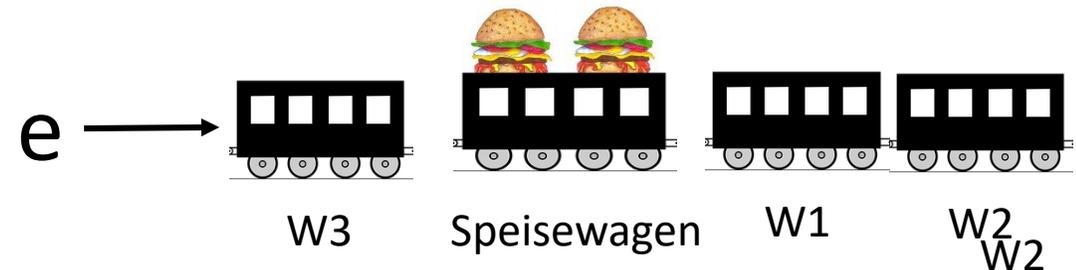
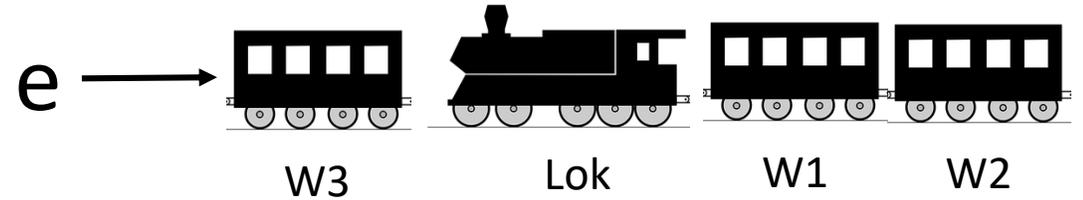
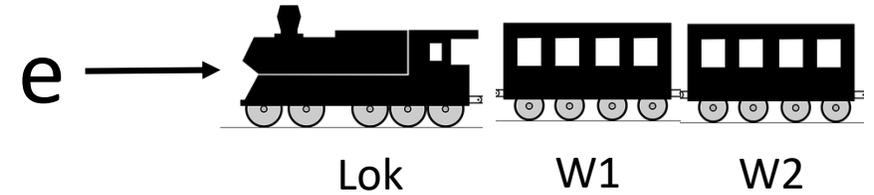
# SBB



```
class Einheit {  
    String typ;  
    Einheit folgend;  
    Einheit(String t, Einheit f){  
        typ = t; folgend = f;  
    }  
}
```

## Was gibt folgender Code aus?

```
Einheit e = new Einheit ("Lok",  
                        new Einheit ("W1",  
                                    new Einheit("W2", null)));  
e = new Einheit("W3", e);  
e.folgend.typ = "Speisewagen";  
while (e != null){  
    Out.print(e.typ + " ");  
    e = e.folgend;  
}
```



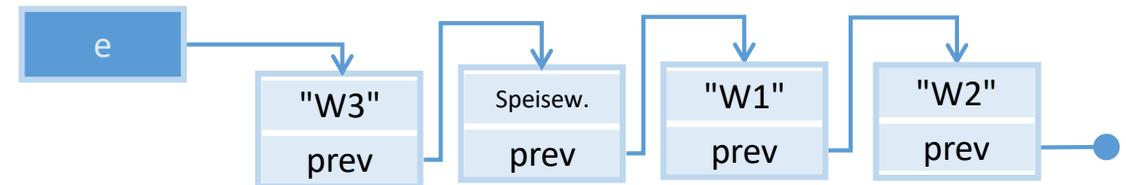
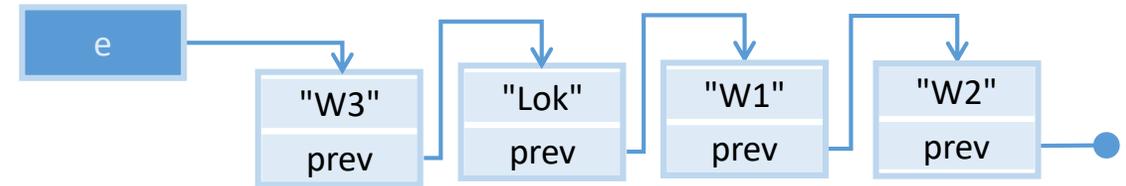
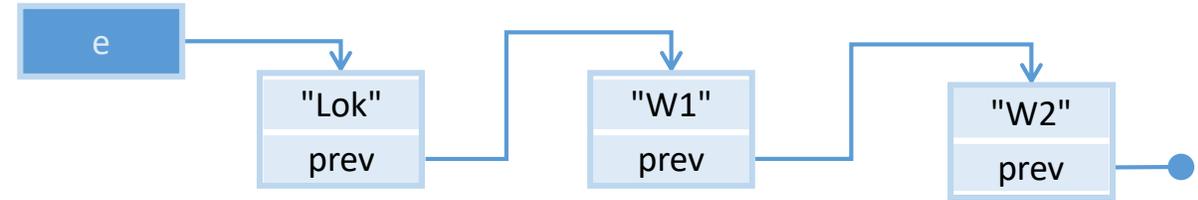
# SBB



```
class Einheit {  
    String typ;  
    Einheit prev;  
    Einheit(String t, Einheit p){  
        typ = t; prev = p;  
    }  
}
```

## Was gibt folgender Code aus?

```
Einheit e = new Einheit ("Lok",  
                        new Einheit ("W1",  
                                    new Einheit("W2", null)));  
e = new Einheit("W3", e);  
e.prev.typ = "Speisewagen";  
while (e != null){  
    Out.print(e.typ + " ");  
    e = e.prev;  
}
```



# SBB



```
class Einheit {
    String typ;
    Einheit folgend;
    Einheit(String t, Einheit f){
        typ = t; folgend = f;
    }
}
```

## Was gibt folgender Code aus?

```
Einheit e = new Einheit ("Lok",
                        new Einheit ("W1",
                                    new Einheit("W2", null)));
e = new Einheit("W3", e);
e.folgend.typ = "Speisewagen";
while (e != null){
    Out.print(e.typ + " ");
    e = e.folgend;
}
```

```
(1)Lok Speisewagen W1 W2
(2)W3 Speisewagen W2 W1
(3)W3 Speisewagen W1 W2
(4)Lok W1 Speisewagen W2
(5)Lok W2 Speisewagen W1
(6)W1 W2 W3 Speisewagen Lok
(7)Gar nichts
(8)Terminiert nicht
```