

Programmierlogik und Programmiergrundlagen

Klausur 2024-08-08

Name nicht vergessen!

Syntax und Struktur von Programmschleifen

Schreiben Sie folgende *for*-Schleifen zu *while*-Schleifen um!

```
acc = 0;
for ( n = 0; n < a.length; n++ ) {
    acc += a[n];
}
```

```
acc = 0;
n = 0;
while ( n < a.length ) {
    acc += a[n];
    n++;
}
```

```
n = 0;
for ( acc = 0; acc < 100; acc += a[n] ) {
    n++;
}
```

```
n = 0;
acc = 0;
while ( acc < 100 ) {
    n++;
    acc += a[n];
}
```

```
for ( n = 0; tests[n].run() != false; n++ ) {
    System.out.println( "Test no. " + n + " OK" );
}
```

```
n = 0;
while ( tests[n].run() != false ) {
    System.out.println( "Test no. " + n + " OK" );
    n++;
}
```

Schreiben Sie folgende *while*-Schleifen zu *for*-Schleifen um

```
n = 999;
while ( n > 0 ) {
    System.out.println( n + " Gurkengläser" );
    n--;
}
```

```
for ( n = 999; n > 0; n-- ) {
    System.out.println( n + " Gurkengläser" );
}
```

```
n = last;
while ( n < a.length - 1 ) {
    n++;
    a[n] = null;
}
```

```
for ( n = last + 1; n < a.length; n++ ) {
    a[n] = null;
}
```

```
l = list;
while ( l.next != null ) {
    l = l.next;
}
```

```
for ( l = list; l.next != null; l = l.next ) {
}
```


Evaluation von Ausdrücken

Gegeben ist folgende Klasse:

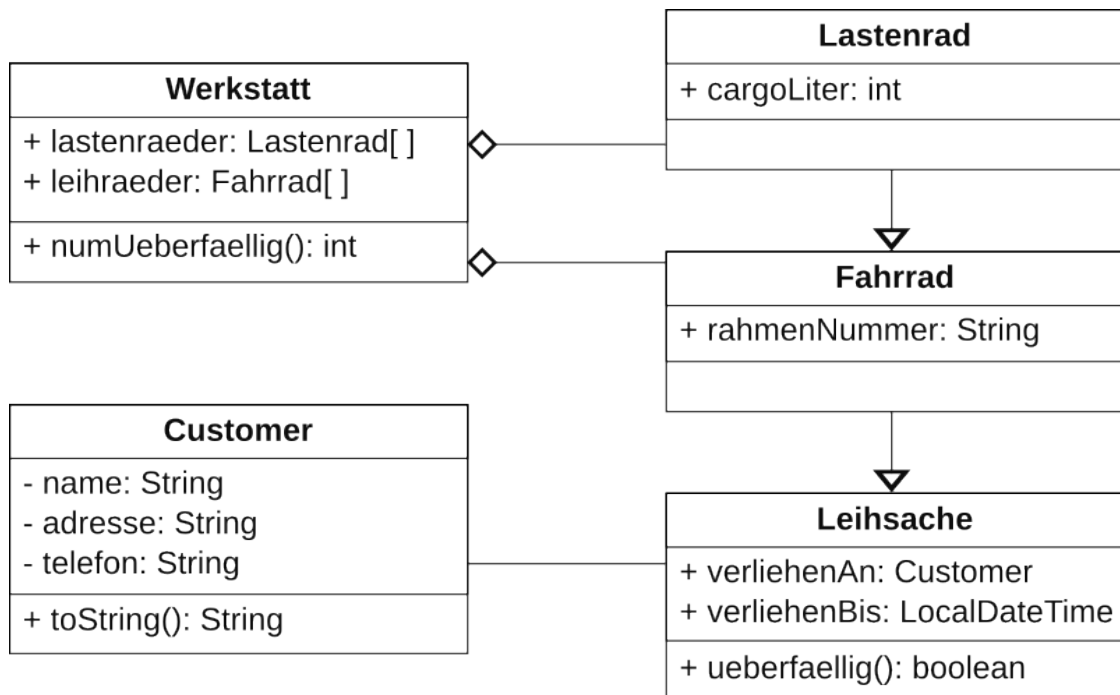
```
public class Evaluation {  
    private int[] arr = new int[]{ 9, 8, 7, 6, 5, 4, 3, 2, 1, 0 };  
  
    public int get(int position) {  
        return arr[position];  
    }  
  
    public void test() {  
        System.out.println( /* Ausdruck */ );  
    }  
}
```

Was gibt die Methode `test()` aus, wenn an Stelle von `/* Ausdruck */` der folgende Ausdruck eingesetzt wird:

- a) `arr[1]` 8
- b) `get(2)` 7
- c) `arr[get(3)]` 3
- d) `get(get(4))` 4
- e) `arr[get(8) + get(7)]` 6
- f) `get(arr[4] + 2)` 2
- g) `get(arr[4 + 2])` 6
- h) `arr[arr.length - arr.length / 2]` 4

Klassen und Vererbung

Gegeben ist ein Programm mit folgender Klassenstruktur:



Das Programm dient der Organisation eines Fahrradverleihs. Fahrräder werden direkt aus der Werkstatt verliehen.

Nehmen Sie für alle folgenden Aufgaben folgende Eigenschaften des Programms an:

- das Array *lastenraeder* enthält 10 Einträge
- Bei allen Einträgen im Array *lastenraeder* handelt es sich um ein Objekt der Klasse *Lastenrad*
- das Array *leihraeder* enthält 50 Einträge
- Bei allen Einträgen im Array *leihraeder* handelt es sich um ein Objekt der Klasse *Fahrrad*
- ist eine *Leihsache* gerade nicht verliehen, sind *verliehenAn* und *verliehenBis* auf den Wert *null* gesetzt
- Die Methode *ueberfaellig()* der Klasse *Leihsache* gibt an, ob das Datum *verliehenBis* in der Vergangenheit liegt. Das ist der Fall, wenn eine *Leihsache* nicht termingerecht zurück gegeben wurde.
- Die *rahmenNummer* eines Fahrrads dient der eindeutigen Identifikation im Falle eines Diebstahls

Aufgaben folgen auf der nächsten Seite.

1. Welche der folgenden Instruktionen sind ausführbar, welche können nicht ausgeführt werden? Begründen Sie ihre Entscheidung!

Extrapunkte: geben Sie an, ob ein fehlerhafter Ausdruck einen Fehler beim Compilieren, oder erst beim Ausführen verursacht.

- a) `leihraeder[0] = lastenraeder[0];`
geht, denn jedes Lastenrad ist auch ein Fahrrad
- b) `lastenraeder[0] = leihraeder[0];`
geht nicht, denn der Typ Lastenrad ist spezifisch und kann nicht ein einfaches Fahrrad aufnehmen, Compile-Time-Fehler
- c) `leihraeder[1] = null;`
geht, denn überall wo ein Objekt-Typ eingesetzt wird, kann auch ein null-Wert stehen denn Objektvariablen sind Pointer, und ein Pointer kann auf null zeigen
- d) `null = leihraeder[1];`
geht nicht, denn null ist ein literaler Wert, bzw. ein Schlüsselwort, und man kann es nicht neu zuweisen
- e) `Customer c = lastenraeder[0].verliehenAn;`
geht, denn eine Variable wird mit einem Wert des richtigen Typen zugewiesen
- f) `Customer c = lastenraeder[0].verliehenAn.toString();`
geht nicht, es wird der falsche Typ zugewiesen, ein String ist kein Customer
Compile-Time-Error

2. Die Methode `numUeberfaellig()` der Klasse `Werkstatt` soll zählen, wie viele Leihräder insgesamt überfällig sind. Vervollständigen Sie die fehlenden Ausdrücke und Schlüsselworte:

```
public int numUeberfaellig() {
    int num = 0; int cnt = 0;

    for ( cnt = 0; cnt < leihraeder.length; cnt++ ) {
        if ( leihraeder[ cnt ].ueberfaellig() ) {
            num++;
        }
    }

    return num;
}
```

3. Welcher Ausdruck ergibt die Rahmennummer vom 4. Lastenrad im Array `lastenraeder`?

`lastenraeder[3].rahmenNummer`

