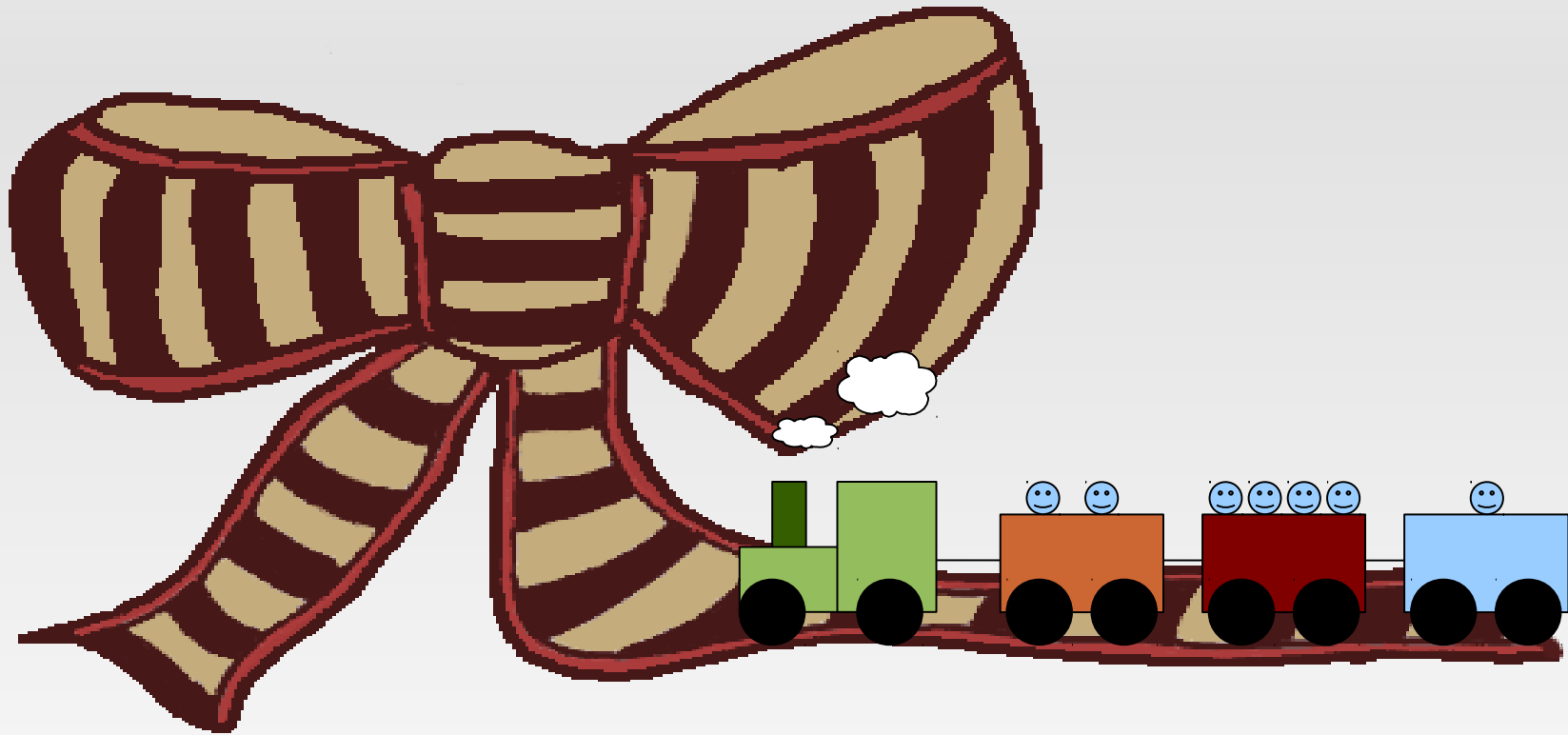


Arrays und Schleifen



Inhaltsverzeichnis

■ Arrays

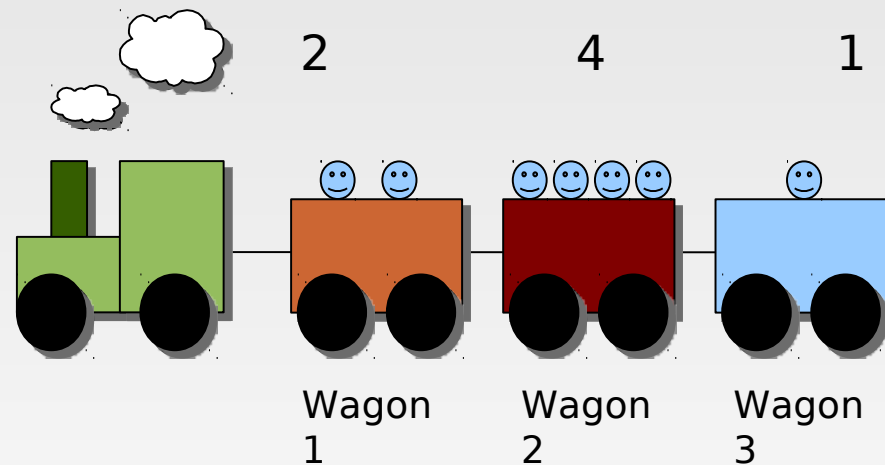
- Was ist ein Array?
- Bauanleitung
- Beispiel Matrix
- Fehler

■ Schleifen

- Wofür Schleifen?
- While-Schleife
 - Fehler
 - Der letzte Ausweg
 - Beispiel: Fakultät
- For-Schleife
 - String-Array
 - Beispiel Matrix

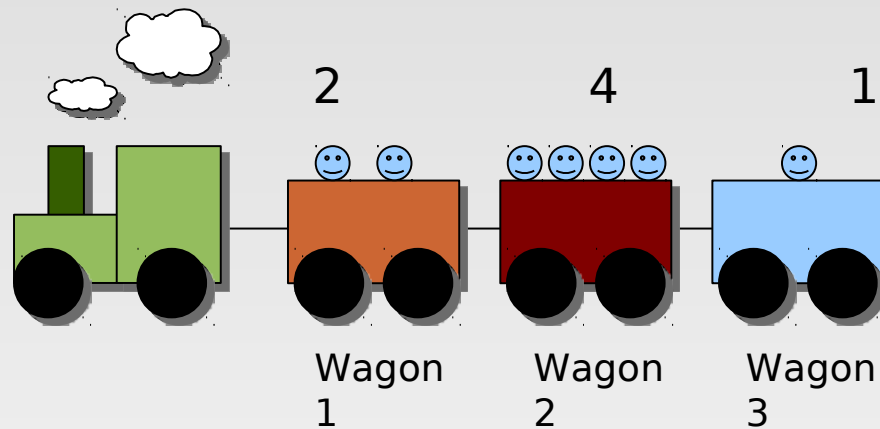
Was ist ein Array?

- Ein Array ist eine Gruppe von Variablen des gleichen Typs, die unter einem gemeinsamen Namen angesprochen werden...

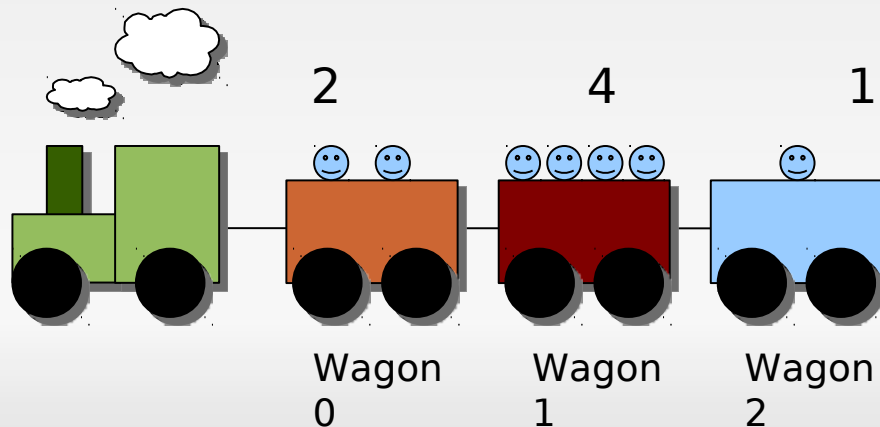


Was ist ein Array?

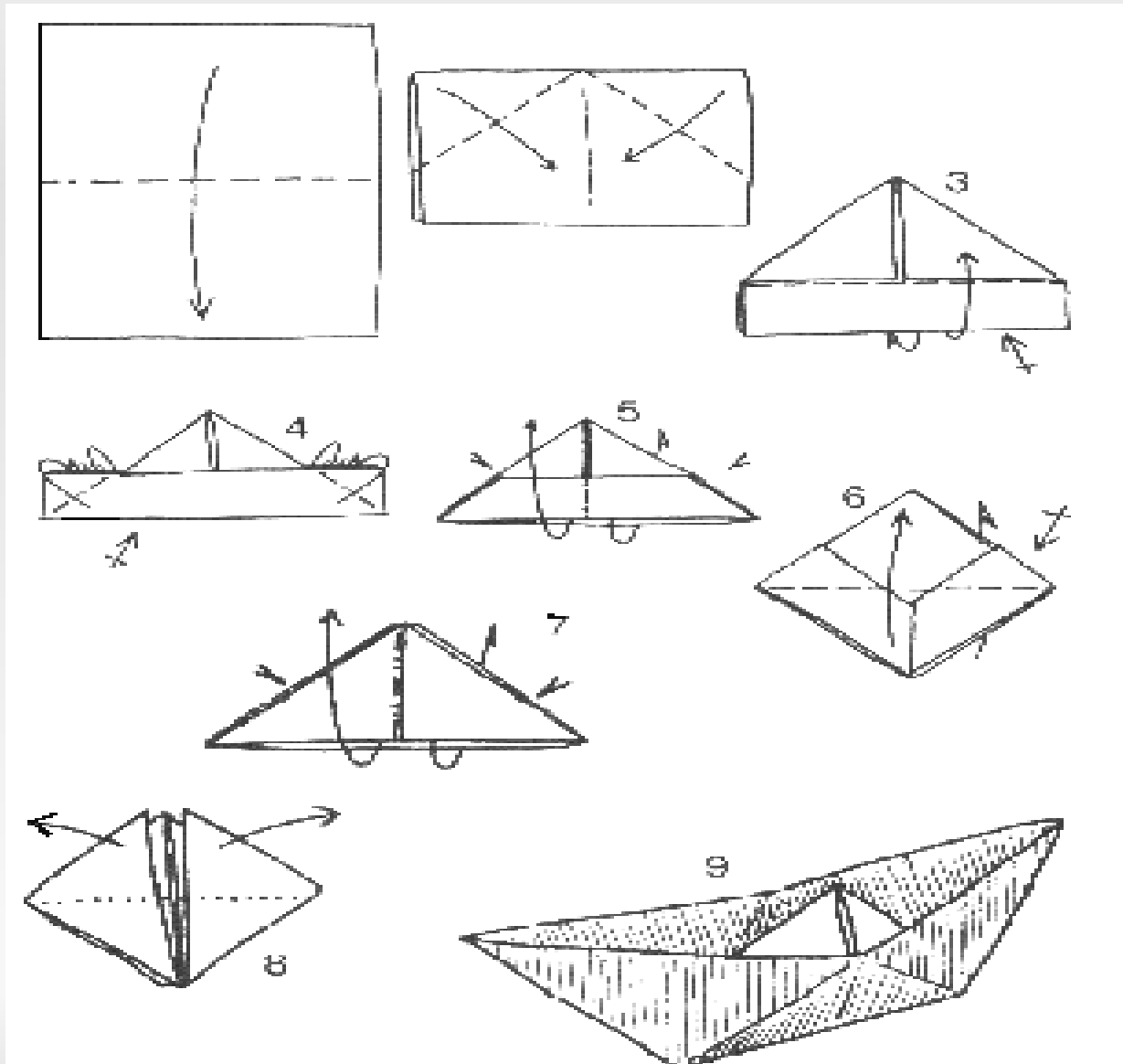
- Ein Array kann man sich wie einen Zug vorstellen



- Java beginnt bei der Nummerierung bei 0, also:



Array-Bauanleitung

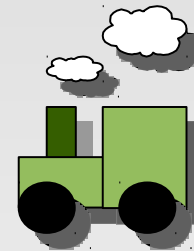


Array-Bauanleitung

- Wie sieht so etwas in Java aus?

Wir haben einen Zug mit Zahlen

```
int[ ] zug;
```



Array-Bauanleitung

- Wie sieht so etwas in Java aus?

Wir haben einen Zug mit Zahlen



```
int[ ] zug;
```

Datentyp

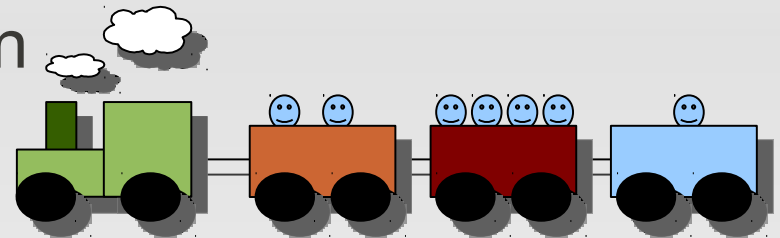
Name

[]Arraysymbol

Array-Bauanleitung

- Wie sieht so etwas in Java aus?

Wir haben einen Zug mit Zahlen



```
int[ ] zug;
```

Außerdem möchten wir einen Zug mit 3 Wagons haben

```
zug = new int[3];
```

Nun können wir den Wagons Werte (Passagiere) zuweisen

```
zug[0] = 2;  
zug[1] = 4;  
zug[2] = 1;
```


Array-Bauanleitung

- Bisherige Deklaration

```
String name;
```

- Bisherige Initialisierung

```
name = "Tim";
```

Array-Bauanleitung

- Deklaration von Arrays

```
String[ ] namen;
```

Arraysymbol[]

A thin black line with an arrowhead at the top points from the 'Arraysymbol[]' box to the '[' character in the 'String[]' part of the code above.

Array-Bauanleitung

- Deklaration von Arrays

```
String[ ] namen;
```

```
String [ ]namen;
```

Äquivalent

Array-Bauanleitung

- Deklaration von Arrays

```
String[ ] namen;
```

```
String [ ]namen;
```

```
String namen[ ];
```



Äquivalent

Array-Bauanleitung

- Deklaration von Arrays

```
String[ ] namen;
```

```
String [ ]namen;
```

```
String namen[ ];
```

Äquivalent

- Der Datentyp(String) kann durch jeden beliebigen ersetzt werden
 - Bsp. int, boolean, double, char...

Array-Bauanleitung

- Deklaration von Arrays

```
String[ ] namen;
```

```
String [ ]namen;
```

```
String namen[ ];
```

- Initialisierung von Arrays

Array-Bauanleitung

- Deklaration von Arrays

```
String[ ] namen;
```

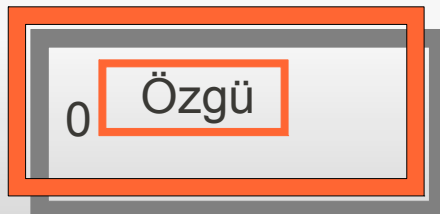
```
String [ ]namen;
```

```
String namen[ ];
```

- Initialisierung von Arrays

```
String[] namen={"Özgü","Tim","Thaddäus"};
```

namen



Array-Bauanleitung

- Deklaration von Arrays

```
String[ ] namen;
```

```
String [ ]namen;
```

```
String namen[ ];
```

- Initialisierung von Arrays

```
String[] namen={"Özgü","Tim","Thaddäus"};
```

namen



Array-Bauanleitung

- Deklaration von Arrays

```
String[ ] namen;
```

```
String [ ]namen;
```

```
String namen[ ];
```

- Initialisierung von Arrays

```
String[] namen={"Özgü","Tim","Thaddäus"};
```

namen



Array-Bauanleitung

- Initialisierung von Arrays
 - Oder so:

```
String[ ] namen= new String[3];
```

namen



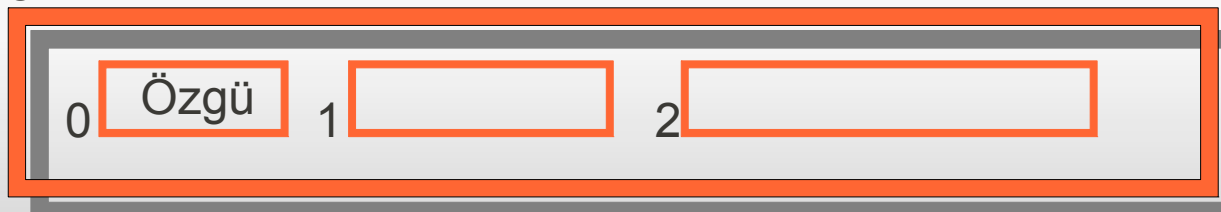
Array-Bauanleitung

- Initialisierung von Arrays
 - eine weitere Möglichkeit

```
String[ ] namen= new String[3];
```

```
namen[0] = "Özgü";
```

namen



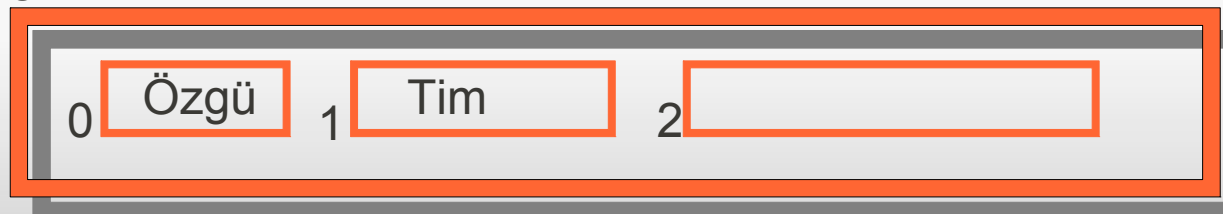
Array-Bauanleitung

- Initialisierung von Arrays
 - eine weitere Möglichkeit

```
String[ ] namen= new String[3];
```

```
namen[0] = "Özgü";  
namen[1] = "Tim";
```

namen



Array-Bauanleitung

- Initialisierung von Arrays
 - eine weitere Möglichkeit

```
String[ ] namen= new String[3];
```

```
namen[0] = "Özgü";  
namen[1] = "Tim";  
namen[2] = "Thaddäus";
```

namen

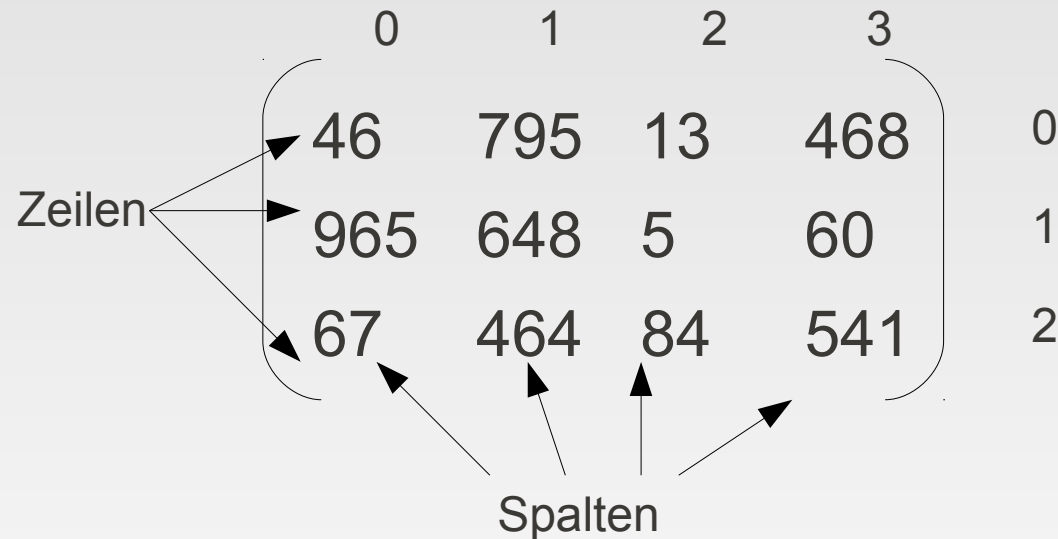


Array – Beispiel: Matrix

- Wie würde man eine Matrix in Java mittels Array realisieren?

Array – Beispiel: Matrix

- Wie würde man eine Matrix in Java mittels Array realisieren?



Array – Beispiel: Matrix

- Wie würde man eine Matrix in Java mittels Array realisieren?

```
int [][]matrix = new int[3][4];  
matrix[0][0] = 46;  
matrix[0][1] = 795;  
matrix[0][2] = 13;  
matrix[0][3] = 468;  
  
matrix[1][0] = 965;  
matrix[1][1] = 648;  
matrix[1][2] = 5;  
matrix[1][3] = 60;  
  
matrix[2][0] = 67;  
matrix[2][1] = 464;  
matrix[2][2] = 84;  
matrix[2][3] = 541;
```

	0	1	2	3	
	46	795	13	468	0
	965	648	5	60	1
	67	464	84	541	2

Array – Beispiel: Matrix

- Wie würde man eine Matrix in Java mittels Array realisieren?

```
int [][]matrix = new int[3][4];  
matrix[0][0] = 46;  
matrix[0][1] = 795;  
matrix[0][2] = 13;  
matrix[0][3] = 468;  
  
matrix[1][0] = 965;  
matrix[1][1] = 648;  
matrix[1][2] = 5;  
matrix[1][3] = 60;  
  
matrix[2][0] = 67;  
matrix[2][1] = 464;  
matrix[2][2] = 84;  
matrix[2][3] = 541;
```

0	1	2	3	
46	795	13	468	0
965	648	5	60	1
67	464	84	541	2

Alternativ

```
int [][]matrix = new int[3][4];  
int []zeile0 = {46,795,13,468};  
int []zeile1 = {965,648,5,60};  
int []zeile2 = {67,464,84,541};  
  
matrix[0] = zeile0;  
matrix[1] = zeile1;  
matrix[2] = zeile2;
```

Array – Fehler #1

```
int []zug;  
zug[0] = 2;  
zug[1] = 4;  
zug[2] = 1;
```

- Wo liegt der Fehler?

Array – Fehler #1

```
int []zug;  
zug[0] = 2;  
zug[1] = 4;  
zug[2] = 1;
```

- Wo liegt der Fehler?
 - Fehlermeldung von Java:

```
ZugFehler1.java:5: variable zug might not have been initialized  
    zug[0] = 2;  
    ^  
1 error
```

Fehler, der beim Kompilieren erkannt wird

Array – Fehler #1

```
int []zug = new int[3];  
zug[0] = 2;  
zug[1] = 4;  
zug[2] = 1;
```

- Wo liegt der Fehler?

Array – Fehler #1

```
int []zug = new int[3];  
zug[0] = 2;  
zug[1] = 4;  
zug[2] = 1;
```

- Wo liegt der Fehler?

Kein Fehler

Array – Fehler #2

```
int []zug = new int[3];  
zug[3] = 2;
```

- Wo liegt der Fehler?

Array – Fehler #2

```
int []zug = new int[3];  
zug[3] = 2;
```

- Wo liegt der Fehler?
 - Fehlermeldung von Java:

```
Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: 3  
    at ZugFehler2.main(ZugFehler2.java:5)
```

Fehler wird erst zur Laufzeit erkannt

Array – Fehler #2

```
int []zug = new int[4];  
zug[3] = 2;
```

- Wo liegt der Fehler?

Array – Fehler #2

```
int []zug = new int[4];  
zug[3] = 2;
```

- Wo liegt der Fehler?

Kein Fehler

Array – Fehler

- Vermeidung einer *ArrayIndexOutOfBoundsException*
 - Mit Hilfe einer if-Bedingung

```
int []zug = new int[3];  
int index = 3;  
if (index < zug.length) {  
    zug[i] = 5;  
} else {  
    //Fehlerbehandlung  
}
```

Array

- So erhält man die Länge (Größe) eines Arrays:

```
variablenname.length
```

```
int[] zug = {2,4,1};  
int size = zug.length;
```

- Welche Zahl steht in size?

Array

- So erhält man die Länge (Größe) eines Arrays:

```
variablenname.length
```

```
int[] zug = {2,4,1};  
int size = zug.length;
```

- Welche Zahl steht in size?



Zusammenfassung

- Was ist ein Array?
- Deklaration und Initialisierung von Arrays
 - auch von 2-dimensionalen Arrays (Matrix)
- Fehlererkennung
 - Initialisierungsfehler
 - `ArrayIndexOutOfBoundsException`

Inhaltsverzeichnis

- Arrays

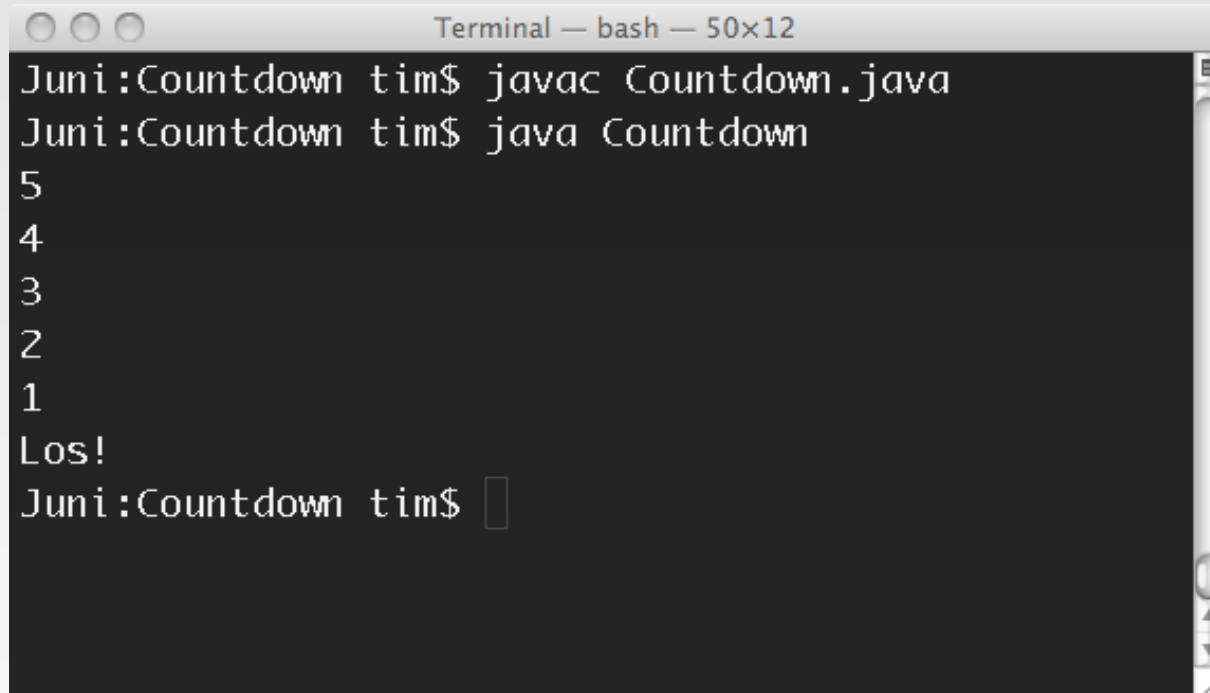
- Was ist ein Array?
- Bauanleitung
- Beispiel Matrix
- Fehler

- Schleifen

- Wofür Schleifen?
- While-Schleife
 - Fehler
 - Der letzte Ausweg
 - Beispiel: Fakultät
- For-Schleife
 - String-Array
 - Beispiel Matrix

Der Anfang einer Schleife

- Aufgabe: Schreibe ein Java Programm das einen Countdown von 5 abwärts auf der Konsole ausgibt:



```
Terminal — bash — 50x12
Juni:Countdown tim$ javac Countdown.java
Juni:Countdown tim$ java Countdown
5
4
3
2
1
Los!
Juni:Countdown tim$
```

Der Anfang einer Schleife

- Folgender Lösungsansatz:

```
1    System.out.println("5");  
2    System.out.println("4");  
3    System.out.println("3");  
4    System.out.println("2");  
5    System.out.println("1");  
6    System.out.println("Los!");
```


Der Anfang einer Schleife

- Folgender Lösungsansatz:

```
1    System.out.println("5");  
2    System.out.println("4");  
3    System.out.println("3");  
4    System.out.println("2");  
5    System.out.println("1");  
6    System.out.println("Los!");
```

Unsauber
Warum?

Der Anfang einer Schleife

- Aufgabe: Schreibe ein Java Programm das einen Countdown von 1000 abwärts auf der Konsole ausgibt:

```
System.out.println("1000");  
System.out.println("999");  
System.out.println("998");  
System.out.println("997");  
System.out.println("996");  
System.out.println("995");  
System.out.println("994");  
System.out.println("993");  
System.out.println("992");  
System.out.println("991");  
System.out.println("990");  
System.out.println("989");  
System.out.println("988");  
System.out.println("987");  
...
```

Ist das elegant?

Der Anfang einer Schleife

- Aufgabe: Schreibe ein Java Programm das einen Countdown von 1000 abwärts auf der Konsole ausgibt:

```
System.out.println("1000");  
System.out.println("999");  
System.out.println("998");  
System.out.println("997");  
System.out.println("996");  
System.out.println("995");  
System.out.println("994");  
System.out.println("993");  
System.out.println("992");  
System.out.println("991");  
System.out.println("990");  
System.out.println("989");  
System.out.println("988");  
System.out.println("987");  
...
```

Nein

Mehrfacher Code

Unnötige Schreibarbeit

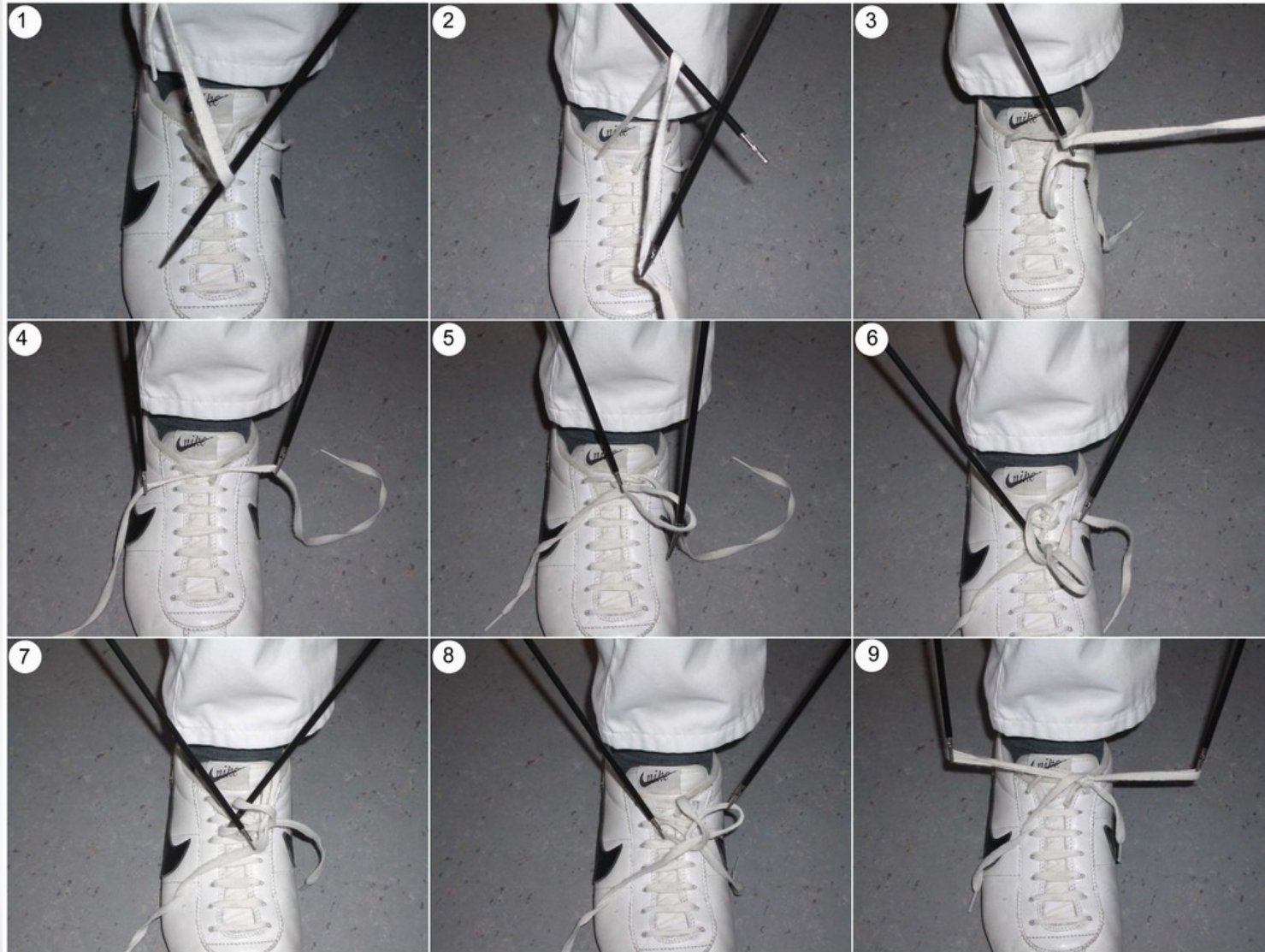
Copy & Paste Fehler

Schleifen

- Die Lösung des Problems



Wie bindet man eine Schleife?



Die while-Schleife

- Aufgabe: Countdown von 1000 abwärts:

```
1  int counter = 1000;  
2  
3  while(counter > 0){  
4      System.out.println(counter);  
5      counter = counter - 1;  
6  }  
7  System.out.println("Los!");
```

Die while-Schleife

- Aufgabe: Countdown von 1000 abwärts:

```
1  int counter = 1000;
2
3  while(counter > 0){
4      System.out.println(counter);
5      counter = counter - 1;
6  }
7  System.out.println("Los!");
```

The diagram illustrates the three main components of a while loop with orange callout boxes and arrows:

- Initialisierung** (Initialization) points to line 1: `int counter = 1000;`
- Bedingung** (Condition) points to line 3: `while(counter > 0){`
- Dekrement** (Decrement) points to line 5: `counter = counter - 1;`

Mögliche Fehlerquellen



Fehler #1

- Aufgabe: Gib alle ungeraden positiven Zahlen kleiner 10 aus

Fehler #1

- Aufgabe: Gib alle ungeraden positiven Zahlen kleiner 10 aus

```
1  int grenze = 10;  
2  int zahl = 1;
```

Fehler #1

- Aufgabe: Gib alle ungeraden positiven Zahlen kleiner 10 aus

```
1  int grenze = 10;  
2  int zahl = 1;  
3  
4  while(zahl < grenze){  
5  
6  
7  
8  
9  
10 }
```

Fehler #1

- Aufgabe: Gib alle ungeraden positiven Zahlen kleiner 10 aus

```
1  int grenze = 10;
2  int zahl = 1;
3
4  while(zahl < grenze){
5      // zahl ist ungerade
6      if(zahl % 2 == 1){
7
8
9      }
10 }
```

Fehler #1

- Aufgabe: Gib alle ungeraden positiven Zahlen kleiner 10 aus

```
1  int grenze = 10;
2  int zahl = 1;
3
4  while(zahl < grenze){
5      // zahl ist ungerade
6      if(zahl % 2 == 1){
7          System.out.println(zahl);
8
9      }
10 }
```

Fehler #1

- Aufgabe: Gib alle ungeraden positiven Zahlen kleiner 10 aus

```
1  int grenze = 10;
2  int zahl = 1;
3
4  while(zahl < grenze){
5      // zahl ist ungerade
6      if(zahl % 2 == 1){
7          System.out.println(zahl);
8          zahl = zahl + 1;
9      }
10 }
```

Ist das korrekt?

Fehler #1

- Aufgabe: Gib alle ungeraden positiven Zahlen kleiner 10 aus

```
1  int grenze = 10;  
2  int zahl = 1;  
3  
4  while(zahl < grenze){  
5      // zahl ist ungerade  
6      if(zahl % 2 == 1){  
7          System.out.println(zahl);  
8          zahl = zahl + 1;  
9      }  
10 }
```

Ist das korrekt?

Nein

Endlosschleife

Fehler #1

- Aufgabe: Gib alle ungeraden positiven Zahlen kleiner 10 aus

```
1  int grenze = 10;
2  int zahl = 1;
3
4  while(zahl < grenze){
5      // zahl ist ungerade
6      if(zahl % 2 == 1){
7          System.out.println(zahl);
8      }
9      zahl = zahl + 1;
10 }
```

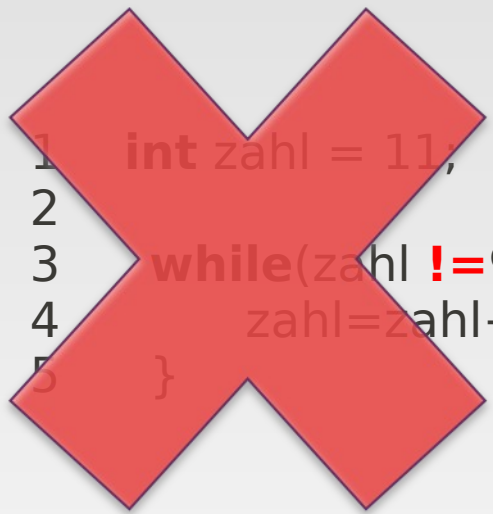
Ist das korrekt?

JA

Fehler #2

```
1  int zahl = 11;  
2  
3  while(zahl !=9){  
4      zahl=zahl+1;  
5  }
```

Fehler #2



```
1  int zahl = 11;  
2  
3  while(zahl != 9){  
4      zahl=zahl+1;  
5  }
```

```
1  int zahl = 11;  
2  
3  while(zahl < 9){  
4      zahl=zahl+1;  
5  }
```

Der letzte Ausweg

- Der Befehl break

```
while(zahl < 9){  
    if(zahl == -1){  
        break;  
    }  
    zahl = zahl + 1;  
}
```

Bitte nur im Ausnahmefall benutzen.

Beispiel: While-Schleife

- Aufgabe: Berechnen sie die Fakultät einer Zahl($4!$).
 - Zur Erinnerung: $4! = 1 * 2 * 3 * 4$

Beispiel: While-Schleife

- Aufgabe: Berechnen sie die Fakultät einer Zahl($4!$).
 - Zur Erinnerung: $4! = 1 * 2 * 3 * 4$
- Der Anfang:
 - Was brauchen wir?
 - Die zu berechnende Zahl
 - Das Ergebnis
 - Einen Zähler

Beispiel: While-Schleife

- Der Anfang:
 - Was brauchen wir?
 - Die zu berechnende Zahl
 - Das Ergebnis
 - Einen Zähler

```
1  int zahl = 4;  
2  int ergebnis = 1; //Denn schon  $0! = 1$   
3  int zaehler = 1;
```

Beispiel: While-Schleife

- Der Anfang:
 - Was brauchen wir?
 - Die zu berechnende Zahl
 - Das Ergebnis
 - Einen Zähler
 - While-Schleife

```
1  int zahl = 4;  
2  int ergebnis = 1; //Denn schon 0! = 1  
3  int zaehler = 1;  
  
4  while(zaehler <= zahl){  
5  
6  
7  }
```

Beispiel: While-Schleife

- Der Anfang:
 - Was brauchen wir?
 - Die zu berechnende Zahl
 - Das Ergebnis
 - Einen Zähler
 - While-Schleife
 - Die eigentliche Rechnung

```
1  int zahl = 4;
2  int ergebnis = 1; //Denn schon 0! = 1
3  int zaehler = 1;

4  while(zaehler <= zahl){
5      ergebnis = ergebnis * zaehler;
6
7  }
```


Beispiel: While-Schleife

- Der Anfang:
 - Was brauchen wir?
 - Die zu berechnende Zahl
 - Das Ergebnis
 - Einen Zähler
 - While-Schleife
 - Die eigentliche Rechnung
 - Schleifendurchlauf hochzählen

```
1  int zahl = 4;
2  int ergebnis = 1; //Denn schon 0! = 1
3  int zaehler = 1;

4  while(zaehler <= zahl){
5      ergebnis = ergebnis * zaehler;
6      zaehler = zaehler + 1;
7  }
8  System.out.println(ergebnis);
```



Fertig

Beispiel: String-Array

- Gib alle Elemente des Arrays `gedicht` auf der Konsole aus!

```
1 String[] gedicht = new String[4];  
2 gedicht[0] = "Ein Mops kam in die Küche,";  
3 gedicht[1] = "und stahl dem Koch ein Ei.";  
4 gedicht[2] = "Da nahm der Koch den Löffel";  
5 gedicht[3] = "und schlug den Mops zu Brei.";
```



Beispiel: String-Array

- Gib alle Elemente des Arrays `gedicht` auf der Konsole aus!

```
1  String[] gedicht = new String[4];
2  gedicht[0] = "Ein Mops kam in die Küche,";
3  gedicht[1] = "und stahl dem Koch ein Ei.";
4  gedicht[2] = "Da nahm der Koch den Löffel";
5  gedicht[3] = "und schlug den Mops zu Brei.";
6
7  for(
```

Beispiel: String-Array

- Gib alle Elemente des Arrays gedicht auf der Konsole aus!

```
1  String[] gedicht = new String[4];  
2  gedicht[0] = "Ein Mops kam in die Küche,";  
3  gedicht[1] = "und stahl dem Koch ein Ei.";  
4  gedicht[2] = "Da nahm der Koch den Löffel";  
5  gedicht[3] = "und schlug den Mops zu Brei.";  
6  
7  for(int i = 0;
```



Laufvariable

Beispiel: String-Array

- Gib alle Elemente des Arrays gedicht auf der Konsole aus!

```
1  String[] gedicht = new String[4];  
2  gedicht[0] = "Ein Mops kam in die Küche,";  
3  gedicht[1] = "und stahl dem Koch ein Ei.";  
4  gedicht[2] = "Da nahm der Koch den Löffel";  
5  gedicht[3] = "und schlug den Mops zu Brei.";  
6  
7  for(int i = 0; i < gedicht.length;
```



Laufvariable

Bedingung

Beispiel: String-Array

- Gib alle Elemente des Arrays gedicht auf der Konsole aus!

```
1 String[] gedicht = new String[4];  
2 gedicht[0] = "Ein Mops kam in die Küche,";  
3 gedicht[1] = "und stahl dem Koch ein Ei.";  
4 gedicht[2] = "Da nahm der Koch den Löffel";  
5 gedicht[3] = "und schlug den Mops zu Brei.";  
6  
7 for(int i = 0; i < gedicht.length; i = i + 1)
```



Laufvariable

Bedingung

Inkrement

Beispiel: String-Array

- Gib alle Elemente des Arrays `gedicht` auf der Konsole aus!

```
1  String[] gedicht = new String[4];
2  gedicht[0] = "Ein Mops kam in die Küche,";
3  gedicht[1] = "und stahl dem Koch ein Ei.";
4  gedicht[2] = "Da nahm der Koch den Löffel";
5  gedicht[3] = "und schlug den Mops zu Brei.";
6
7  for(int i = 0; i < gedicht.length; i = i + 1){
8      System.out.println(gedicht[i]);
9  }
```

Beispiel Matrix

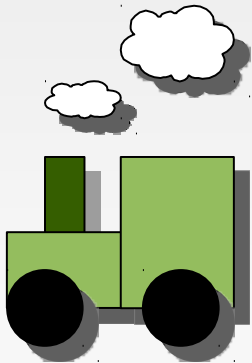
- Aufgabe: Finde das Maximum aus jeder Spalte und speichere das Ergebnis in einem neuen Array!

46	795	13	468
965	648	5	60
67	464	84	542

Beispiel Matrix

- Aufgabe: Finde das Maximum aus jeder Spalte und speichere das Ergebnis in einem neuen Array!

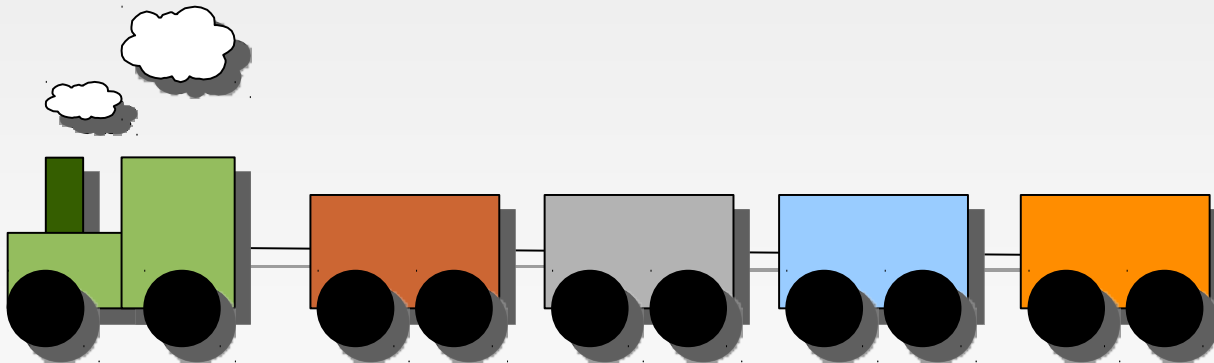
46	795	13	468
965	648	5	60
67	464	84	542



Beispiel Matrix

- Aufgabe: Finde das Maximum aus jeder Spalte und speichere das Ergebnis in einem neuen Array!

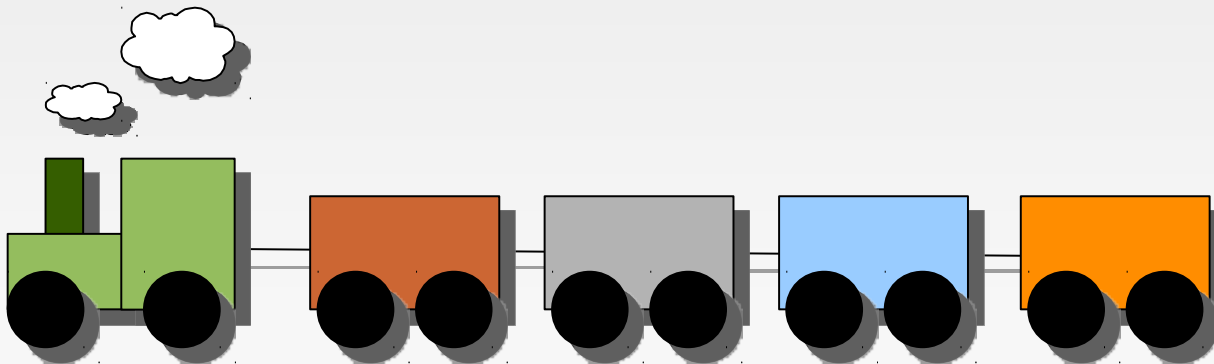
46	795	13	468
965	648	5	60
67	464	84	542



Beispiel Matrix

- Aufgabe: Finde das Maximum aus jeder Spalte und speichere das Ergebnis in einem neuen Array!

46	795	13	468
965	648	5	60
67	464	84	542



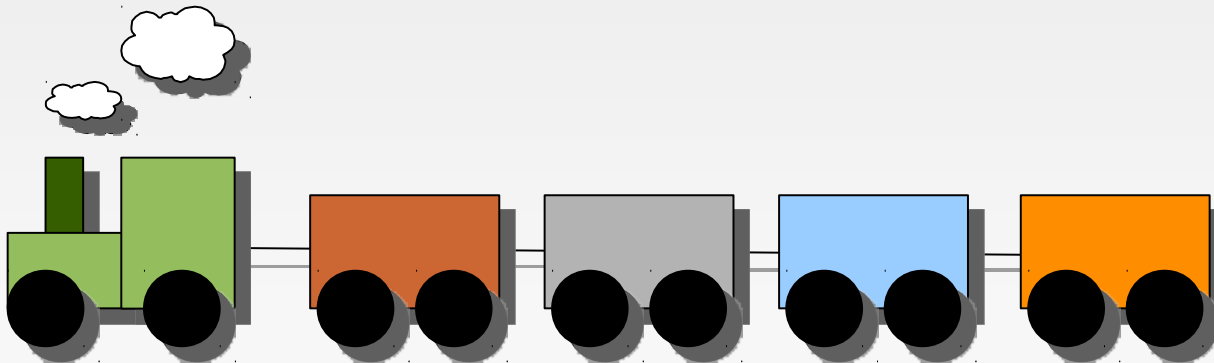
Aktuelles
Maximum

0

Beispiel Matrix

- Aufgabe: Finde das Maximum aus jeder Spalte und speichere das Ergebnis in einem neuen Array!

46	795	13	468
965	648	5	60
67	464	84	542



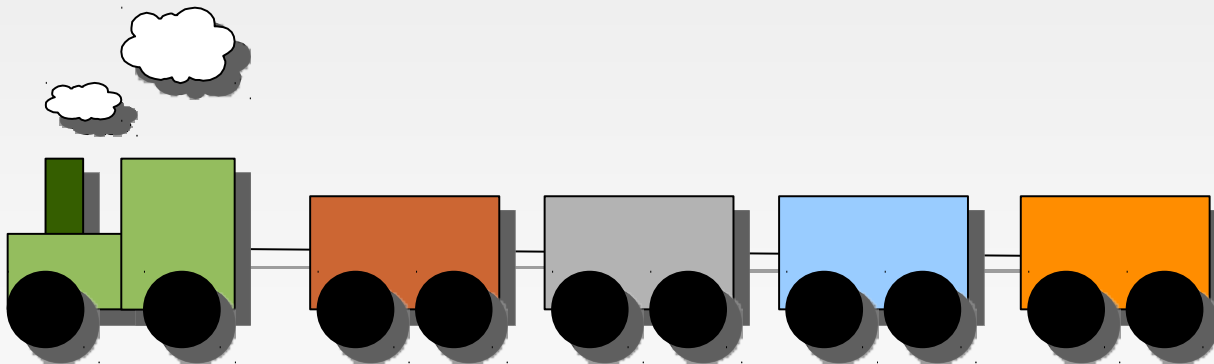
Aktuelles
Maximum

46

Beispiel Matrix

- Aufgabe: Finde das Maximum aus jeder Spalte und speichere das Ergebnis in einem neuen Array!

46	795	13	468
965	648	5	60
67	464	84	542



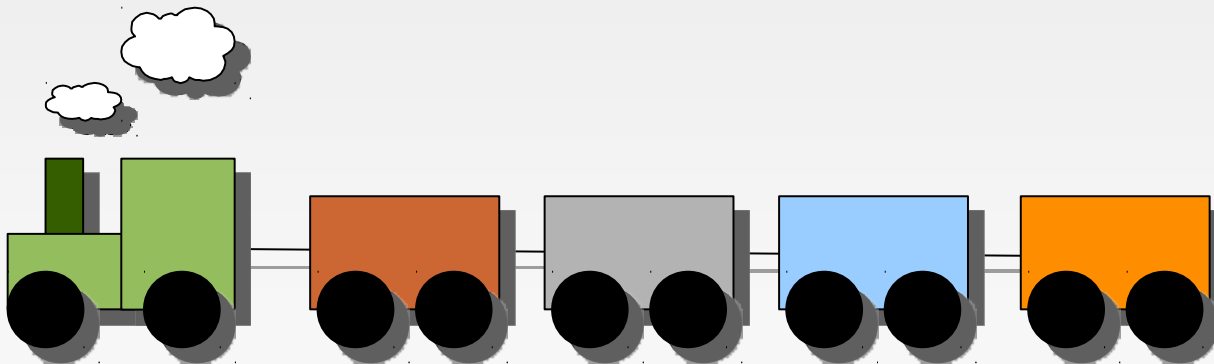
Aktuelles
Maximum

46

Beispiel Matrix

- Aufgabe: Finde das Maximum aus jeder Spalte und speichere das Ergebnis in einem neuen Array!

46	795	13	468
965	648	5	60
67	464	84	542



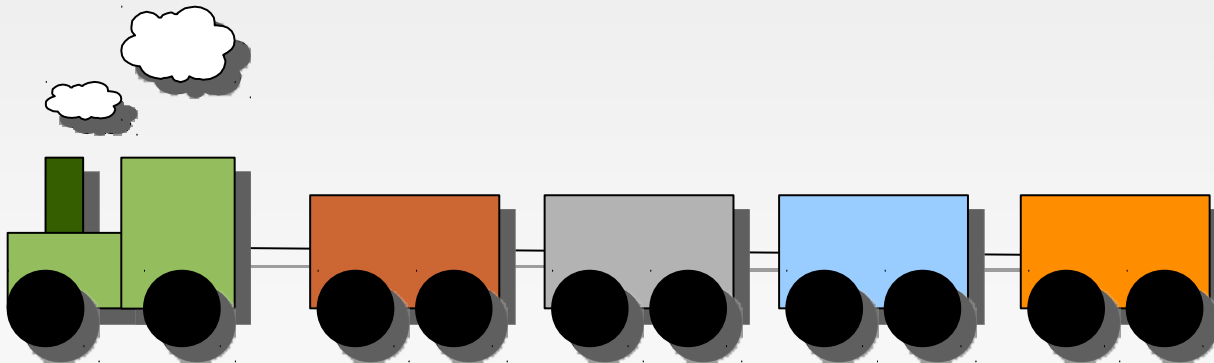
Aktuelles
Maximum

965

Beispiel Matrix

- Aufgabe: Finde das Maximum aus jeder Spalte und speichere das Ergebnis in einem neuen Array!

46	795	13	468
965	648	5	60
67	464	84	542



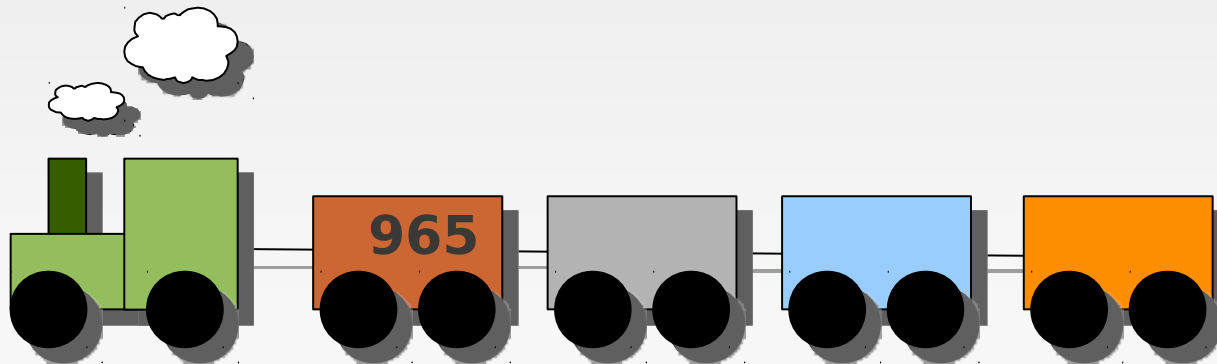
Aktuelles
Maximum

965

Beispiel Matrix

- Aufgabe: Finde das Maximum aus jeder Spalte und speichere das Ergebnis in einem neuen Array!

46	795	13	468
965	648	5	60
67	464	84	542



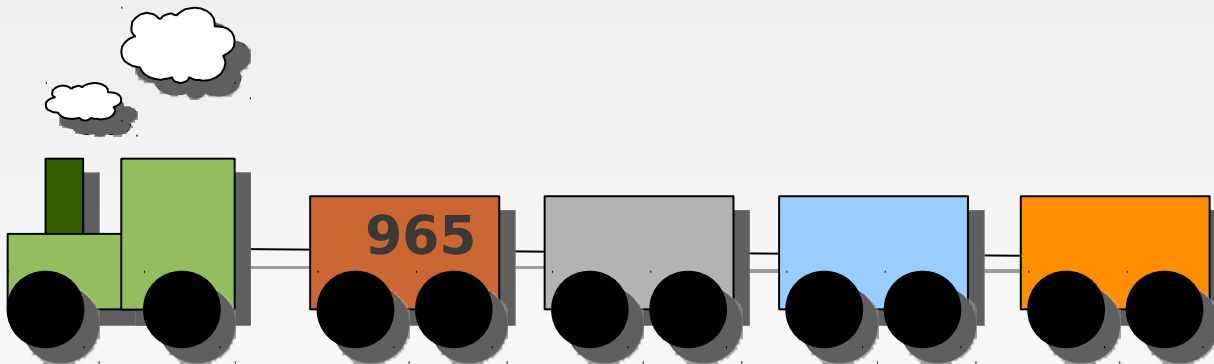
Aktuelles
Maximum

965

Beispiel Matrix

- Aufgabe: Finde das Maximum aus jeder Spalte und speichere das Ergebnis in einem neuen Array!

46	795	13	468
965	648	5	60
67	464	84	542

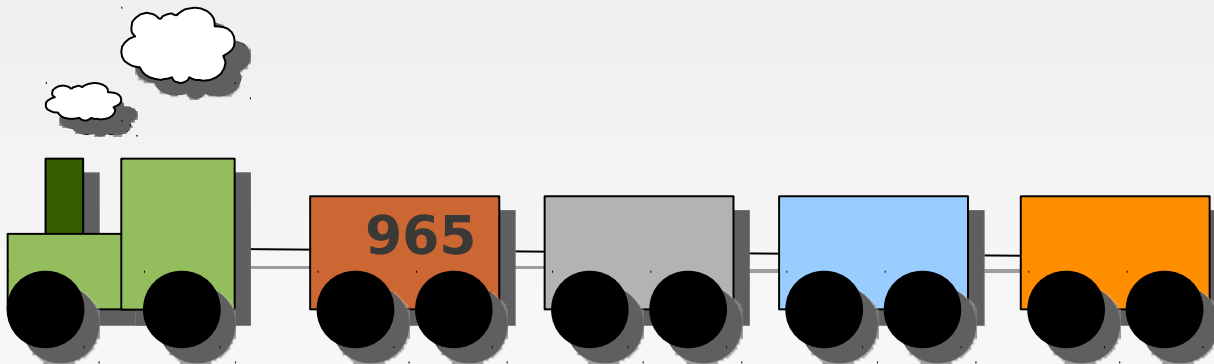


Aktuelles
Maximum
0

Beispiel Matrix

- Aufgabe: Finde das Maximum aus jeder Spalte und speichere das Ergebnis in einem neuen Array!

46	795	13	468
965	648	5	60
67	464	84	542



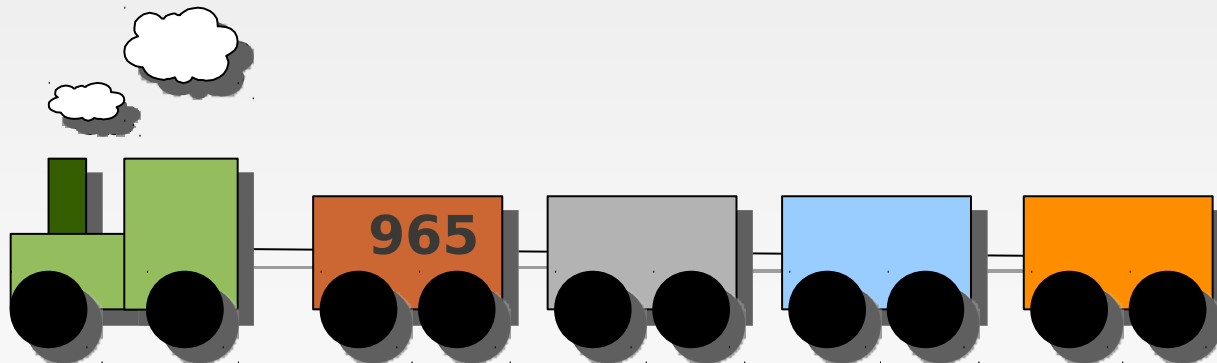
Aktuelles
Maximum

795

Beispiel Matrix

- Aufgabe: Finde das Maximum aus jeder Spalte und speichere das Ergebnis in einem neuen Array!

46	795	13	468
965	648	5	60
67	464	84	542



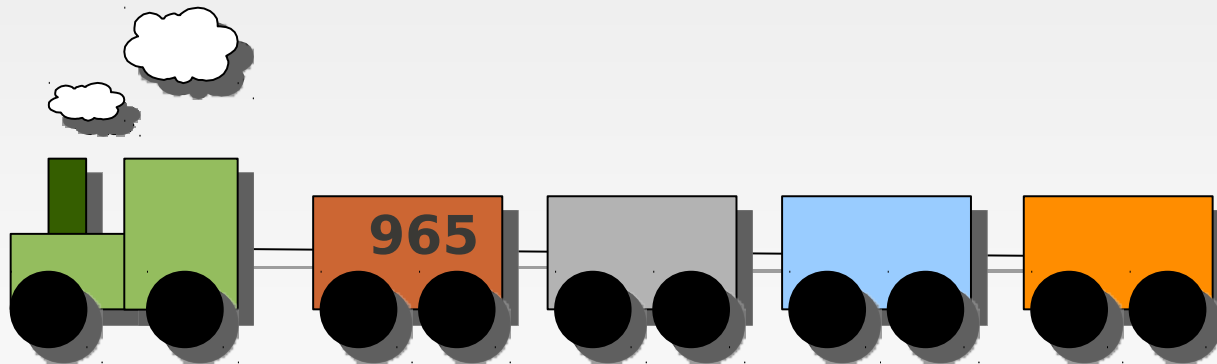
Aktuelles
Maximum

795

Beispiel Matrix

- Aufgabe: Finde das Maximum aus jeder Spalte und speichere das Ergebnis in einem neuen Array!

46	795	13	468
965	648	5	60
67	464	84	542



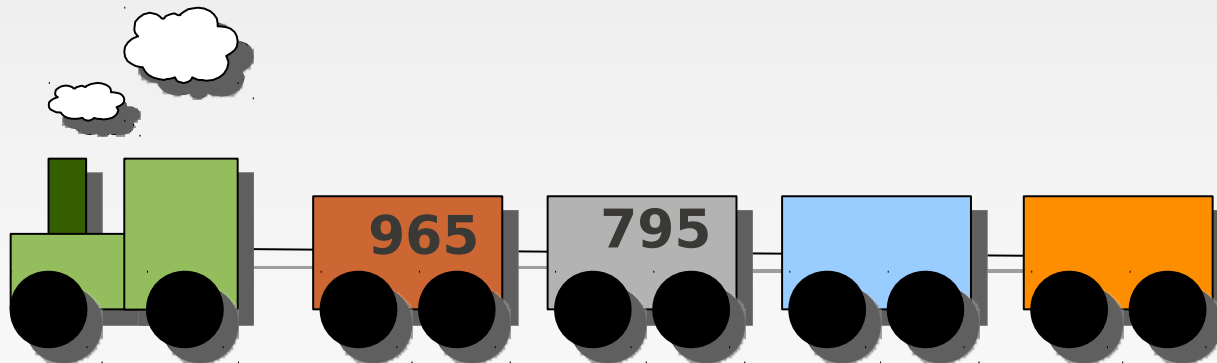
Aktuelles
Maximum

795

Beispiel Matrix

- Aufgabe: Finde das Maximum aus jeder Spalte und speichere das Ergebnis in einem neuen Array!

46	795	13	468
965	648	5	60
67	464	84	542



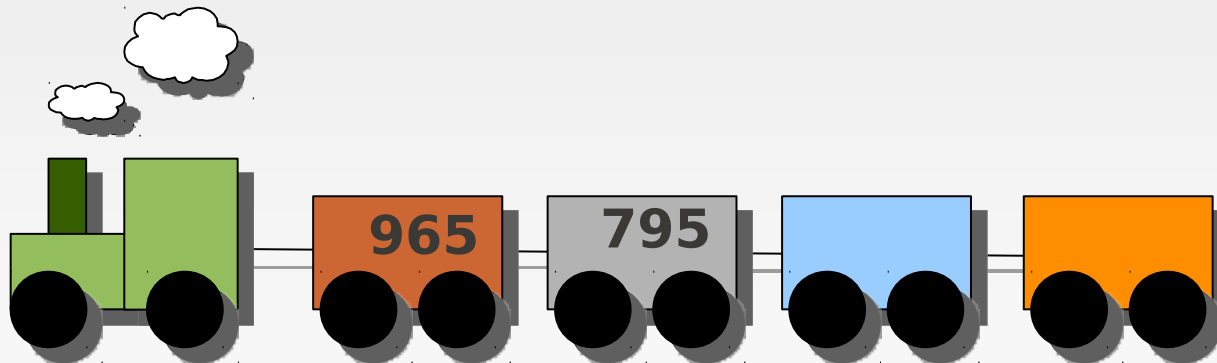
Aktuelles
Maximum

795

Beispiel Matrix

- Aufgabe: Finde das Maximum aus jeder Spalte und speichere das Ergebnis in einem neuen Array!

46	795	13	468
965	648	5	60
67	464	84	542



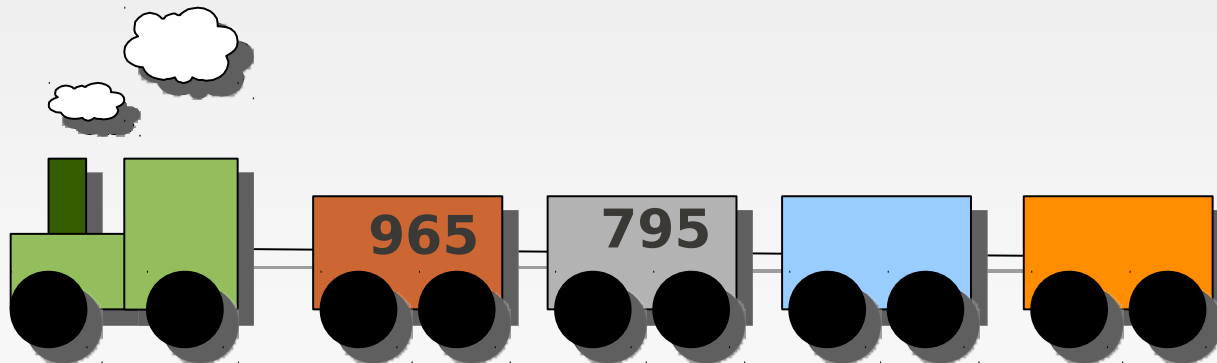
Aktuelles
Maximum

0

Beispiel Matrix

- Aufgabe: Finde das Maximum aus jeder Spalte und speichere das Ergebnis in einem neuen Array!

46	795	13	468
965	648	5	60
67	464	84	542



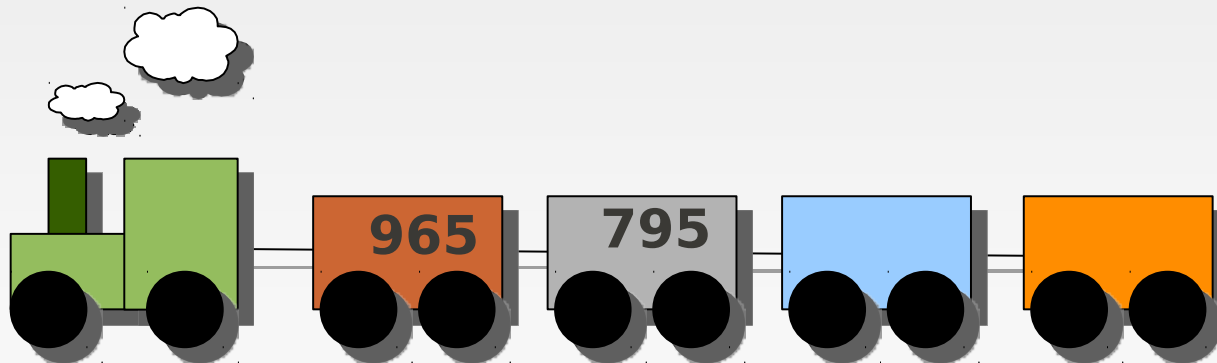
Aktuelles
Maximum

13

Beispiel Matrix

- Aufgabe: Finde das Maximum aus jeder Spalte und speichere das Ergebnis in einem neuen Array!

46	795	13	468
965	648	5	60
67	464	84	542



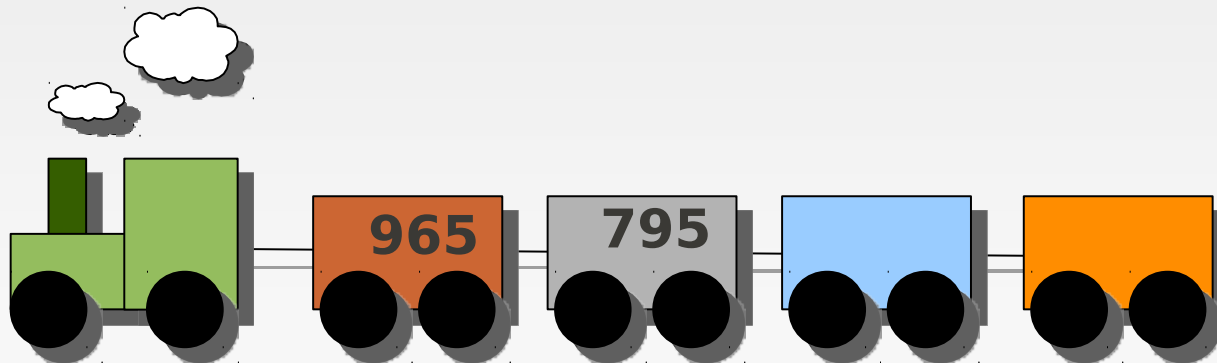
Aktuelles
Maximum

13

Beispiel Matrix

- Aufgabe: Finde das Maximum aus jeder Spalte und speichere das Ergebnis in einem neuen Array!

46	795	13	468
965	648	5	60
67	464	84	542



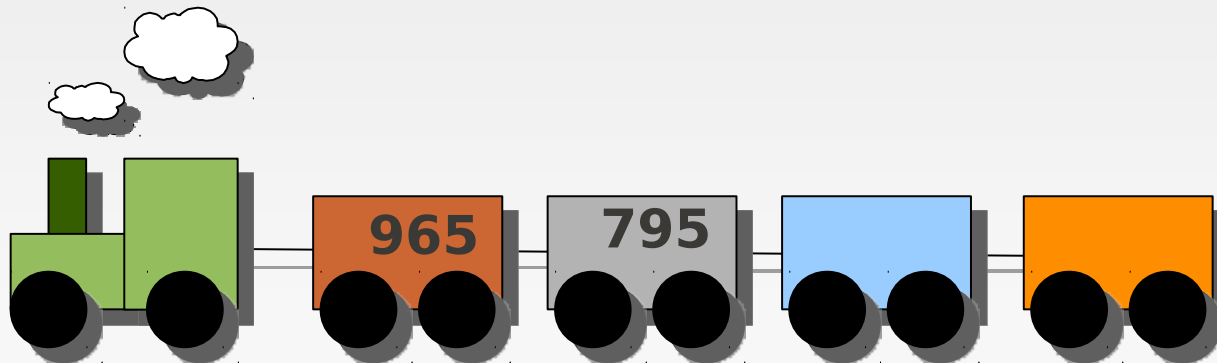
Aktuelles
Maximum

13

Beispiel Matrix

- Aufgabe: Finde das Maximum aus jeder Spalte und speichere das Ergebnis in einem neuen Array!

46	795	13	468
965	648	5	60
67	464	84	542



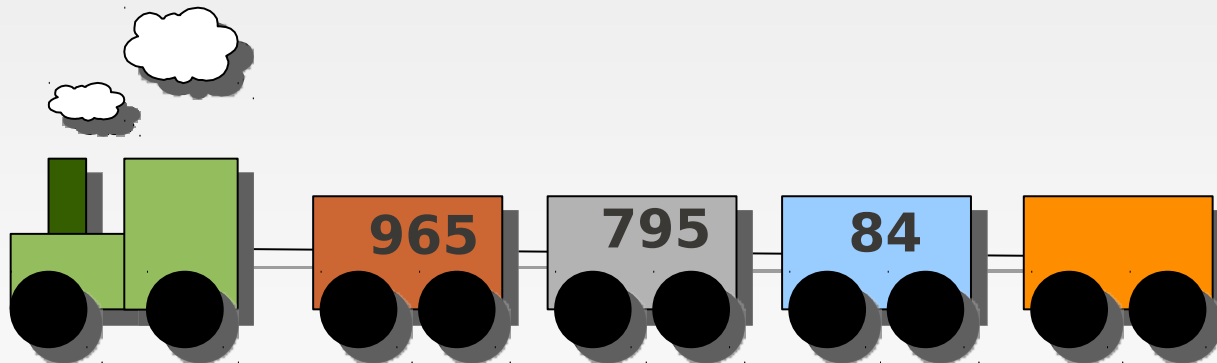
Aktuelles
Maximum

84

Beispiel Matrix

- Aufgabe: Finde das Maximum aus jeder Spalte und speichere das Ergebnis in einem neuen Array!

46	795	13	468
965	648	5	60
67	464	84	542



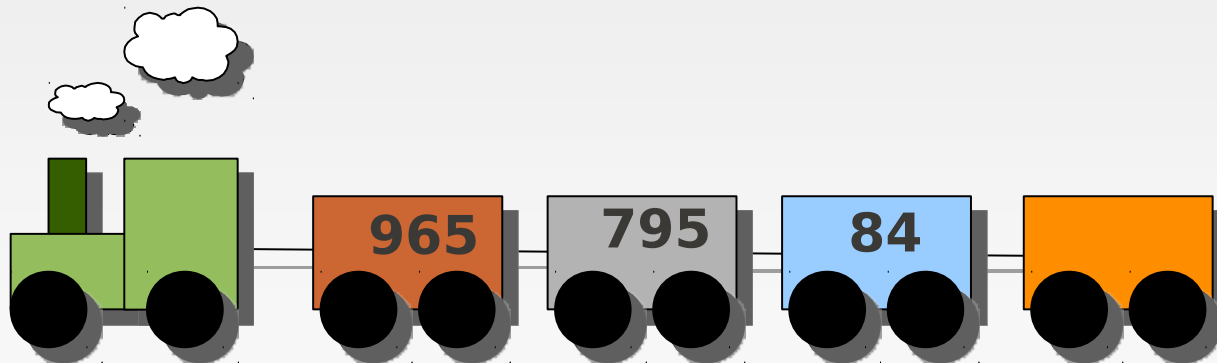
Aktuelles
Maximum

84

Beispiel Matrix

- Aufgabe: Finde das Maximum aus jeder Spalte und speichere das Ergebnis in einem neuen Array!

46	795	13	468
965	648	5	60
67	464	84	542

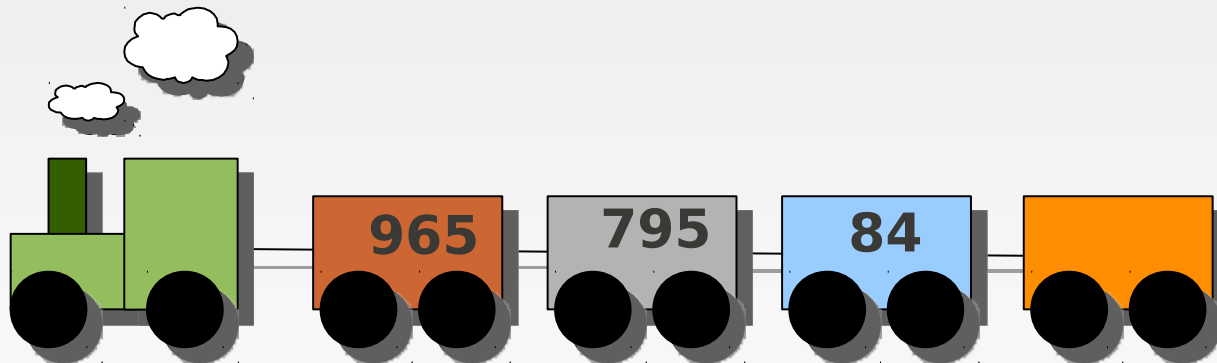


Aktuelles
Maximum
0

Beispiel Matrix

- Aufgabe: Finde das Maximum aus jeder Spalte und speichere das Ergebnis in einem neuen Array!

46	795	13	468
965	648	5	60
67	464	84	542



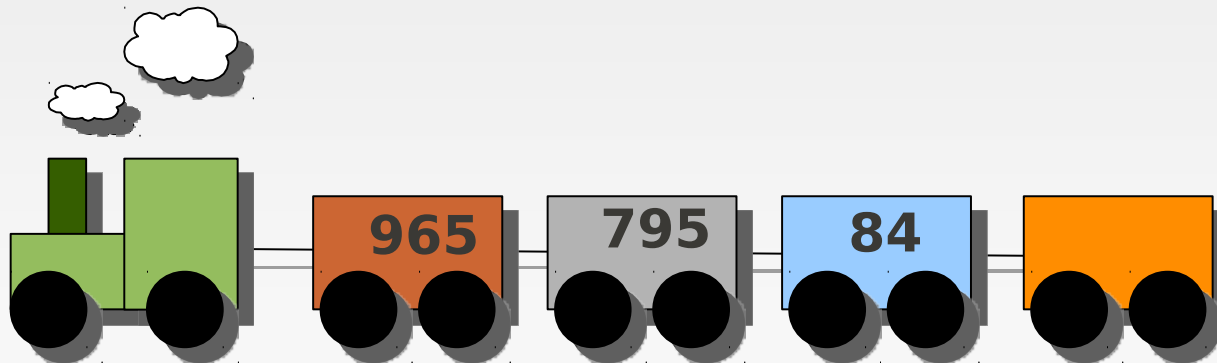
Aktuelles
Maximum

468

Beispiel Matrix

- Aufgabe: Finde das Maximum aus jeder Spalte und speichere das Ergebnis in einem neuen Array!

46	795	13	468
965	648	5	60
67	464	84	542



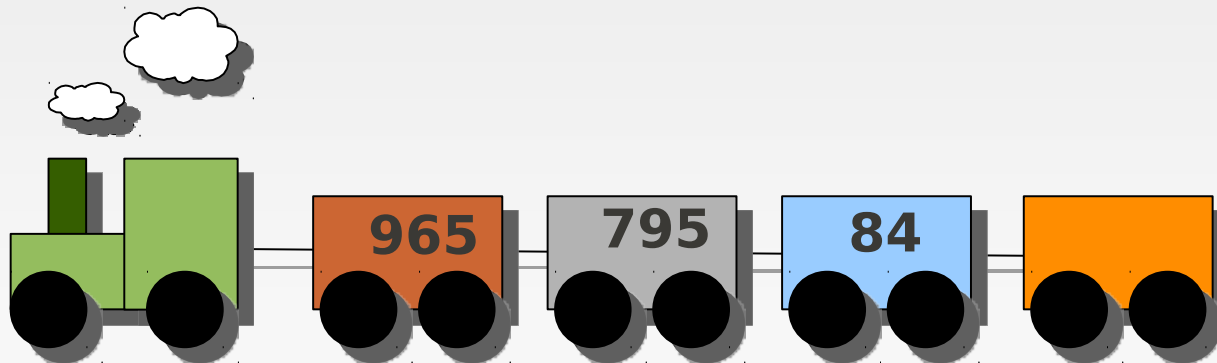
Aktuelles
Maximum

468

Beispiel Matrix

- Aufgabe: Finde das Maximum aus jeder Spalte und speichere das Ergebnis in einem neuen Array!

46	795	13	468
965	648	5	60
67	464	84	542



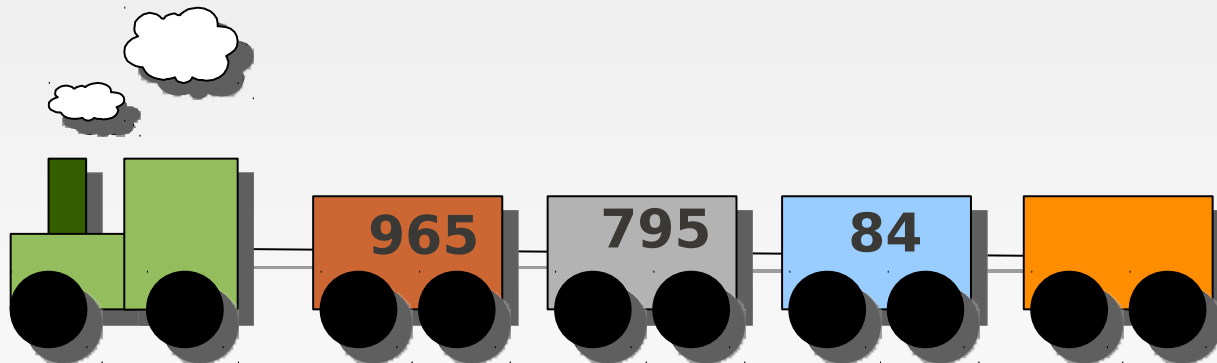
Aktuelles
Maximum

468

Beispiel Matrix

- Aufgabe: Finde das Maximum aus jeder Spalte und speichere das Ergebnis in einem neuen Array!

46	795	13	468
965	648	5	60
67	464	84	542



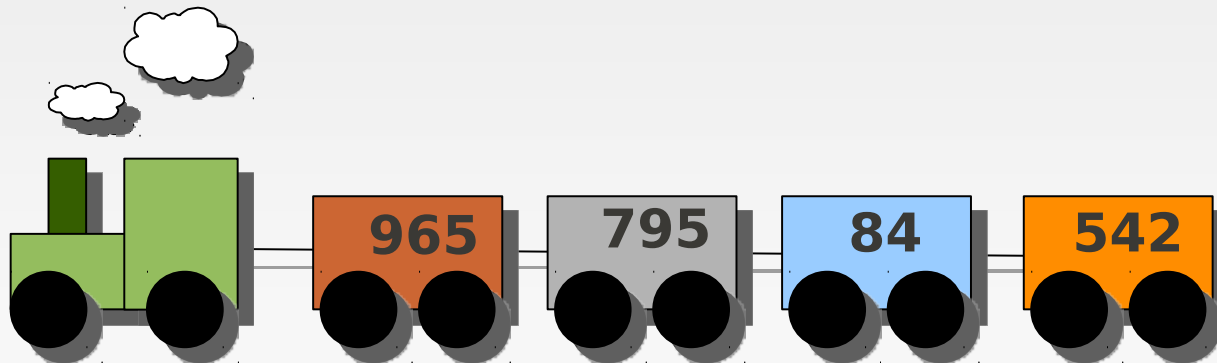
Aktuelles
Maximum

542

Beispiel Matrix

- Aufgabe: Finde das Maximum aus jeder Spalte und speichere das Ergebnis in einem neuen Array!

46	795	13	468
965	648	5	60
67	464	84	542



Aktuelles
Maximum

542

Beispiel Matrix

```
int[][] matrix =  $\begin{bmatrix} 46 & 795 & 13 & 468 \\ 965 & 648 & 5 & 60 \\ 67 & 464 & 84 & 542 \end{bmatrix}$ 
```

Beispiel Matrix

`int[][] matrix =` $\begin{bmatrix} 46 & 795 & 13 & 468 \\ 965 & 648 & 5 & 60 \\ 67 & 464 & 84 & 542 \end{bmatrix}$

1 `int[] ergebnis = new int[4];`

Beispiel Matrix

`int[][] matrix =` $\begin{bmatrix} 46 & 795 & 13 & 468 \\ 965 & 648 & 5 & 60 \\ 67 & 464 & 84 & 542 \end{bmatrix}$

```
1  int[] ergebnis = new int[4];
2
3  for(int j = 0; j < 4; j++){
4
5
6
7
8
9
10
11 }
```

Beispiel Matrix

`int[][] matrix =` $\begin{bmatrix} 46 & 795 & 13 & 468 \\ 965 & 648 & 5 & 60 \\ 67 & 464 & 84 & 542 \end{bmatrix}$

```
1  int[] ergebnis = new int[4];
2
3  for(int j = 0; j < 4; j++){
4
5      for(int i = 0; i < 3; i++){
6
7
8
9      }
10
11 }
```

Beispiel Matrix

`int[][] matrix =` $\begin{bmatrix} 46 & 795 & 13 & 468 \\ 965 & 648 & 5 & 60 \\ 67 & 464 & 84 & 542 \end{bmatrix}$

```
1  int[] ergebnis = new int[4];
2
3  for(int j = 0; j < 4; j++){
4      int maximum = 0;
5      for(int i = 0; i < 3; i++){
6
7
8
9      }
10
11 }
```

Beispiel Matrix

`int[][] matrix =` $\begin{bmatrix} 46 & 795 & 13 & 468 \\ 965 & 648 & 5 & 60 \\ 67 & 464 & 84 & 542 \end{bmatrix}$

```
1  int[] ergebnis = new int[4];
2
3  for(int j = 0; j < 4; j++){
4      int maximum = 0;
5      for(int i = 0; i < 3; i++){
6          if(maximum < matrix[i][j]){
7              maximum = matrix[i][j];
8          }
9      }
10
11 }
```

Beispiel Matrix

int[][] matrix = $\begin{bmatrix} 46 & 795 & 13 & 468 \\ 965 & 648 & 5 & 60 \\ 67 & 464 & 84 & 542 \end{bmatrix}$

```
1  int[] ergebnis = new int[4];
2
3  for(int j = 0; j < 4; j++){
4      int maximum = 0;
5      for(int i = 0; i < 3; i++){
6          if(maximum < matrix[j][i]){
7              maximum = matrix[j][i];
8          }
9      }
10     ergebnis[j] = maximum;
11 }
```


Beispiel Matrix

int[][] matrix = $\begin{bmatrix} 46 & 795 & 13 & 468 \\ 965 & 648 & 5 & 60 \\ 67 & 464 & 84 & 542 \end{bmatrix}$

```
1  int[] ergebnis = new int[4];
2
3  for(int j = 0; j < 4; j++){
4      int maximum = 0;
5      for(int i = 0; i < 3; i++){
6          if(maximum < matrix[i][j]){
7              maximum = matrix[i][j];
8          }
9      }
10     ergebnis[j] = maximum;
11 }
```

Fertig

while-Schleife / for-Schleife

- while-Schleifen und for-Schleifen können ineinander umgewandelt werden
- while-Schleife:

```
1  int counter = 1000;  
2  
3  while(counter > 0){  
4      System.out.println(counter);  
5      counter = counter - 1;  
6  }
```

while-Schleife / for-Schleife

- while-Schleifen und for-Schleifen können ineinander umgewandelt werden
- for-Schleife:

```
1  
2  
3  for(int counter = 1000; counter > 0; counter--){  
4      System.out.println(counter);  
5  
6  }
```

Zusammenfassung

- Wann/Warum eine Schleife?
- while-Schleife
 - Aufbau und Anwendung
 - mögliche Fehlerquellen
 - Beispiel Fakultätsrechnung
- for-Schleife
 - Aufbau und Anwendung
 - Beispiel String-Array
 - Beispiel Matrix

???

Fragen ???

Ende

Vielen Dank für die Aufmerksamkeit

Jetzt: Übungsaufgaben im Tel
Fragen jederzeit an uns

Quellen der Bilder chronologisch:

- [1] User: mightymikey73, URL: <http://www.flickr.com/photos/mightymikey73/2459063122/sizes/o/>
- [2] User: mirgo, URL: www.flickr.com/photos/migro/218662105/sizes/l/
- [3] User: oliklee, URL: http://farm4.static.flickr.com/3276/2945259290_df52829f09_o.jpg
- [4] User: tottix, URL: <http://www.flickr.com/photos/tottix/4279051374/sizes/l/>