

## MINI-PROJEKT 1

### 1. Was ist eine Zeitschleife:

Eine Zeitschleife ist ein Phänomen, welches einen Start- und Endpunkt enthält. Jedoch sind beide Punkte „Events“ oder Werte. Die Zeit zwischen beiden Events ist die „Schleife“.

Der Ablauf in dieser Schleife kann sich immer unterscheiden, was dazu führt, dass die Länge und die Werte innerhalb der Schleife variieren können.

Zusätzlich kann eine Schleife auch durch/ unterbrochen werden, indem sich der Start / End-Punkt nicht wiederholt.

In vielen Filmen findet man verschiedene Zeitschleifen:

Happy Deathday => Sie wacht immer am Morgen ihres Geburtstags auf, nachdem sie ermordet wird. Sie ist in einer Zeitschleife und der einzige Weg diese zu durchbrechen ist es ihren Geburtstag zu Überleben („Der Endpunkt tritt nicht ein = Zeitschleife wurde durchbrochen“)

### 2. 1000 Zahlen:

Wir haben mit ChatGPT und mit Java jeweils eine Liste mit 1000 Zahlenwerten mit den Zahlen [0,9] erstellen lassen/ erstellt.

CT: Man benötigt 1000 Zahlen zwischen 0 und 9, wir haben rumprobiert wie man diese am besten darstellen bzw. erstellen kann. Außerdem haben wir Computational Thinking auch beim Verwenden von ChatGPT und Java angewendet.

ST: Nicht explizit in diesem Bereich angewendet, aber bei 1.

### 3. Pseudocode:

„Eine Zeitschleife würde sich zeigen, wenn eine Serie von Zahlenwerten öfter vorkommt, die wäre der Anfang einer Wiederholung.“ Kombiniert mit unserer Definition für Zeitschleifen sollte das bedeuten, dass es sich um eine Wiederholung von Zahlenwerten am Anfang und am Ende handelt.

Da ein einzelner Wert von 0 bis 9 ca. 100-mal in 1000 Werten vorkommt wäre es vermutlich besser den Anfang und das Ende mit mehreren Werten darzustellen. 5 wären zu viele da die Chance zu niedrig ist, dass man sie in 200 verschiedenen Werten 2-mal erhält, somit gilt:

„Je weniger verschiedene Werte, desto höher die Chance, dass eine Zeitschleife auftritt“

Max Hammerer (12424788), Ben Katz (12419322)

Nimmt man an, dass der Anfangs/Endwert aus 3 Zahlen besteht: Wenn dieser Wert nochmal auftritt dann wird die Schleife gefunden, wobei die 3 Zahlen von dem Anfangswert („der Wiederholung der Schleife“) der Endwert sind. Also kann man sagen:

„Wenn die Startzeichen („Anfangswert“) mehrmals vorkommen gibt es eine Zeitschleife,  
wenn nicht dann gibt es keine“

If (derzeitige Zahlen = Anfangszahlen)

Then „Zeitschleife gefunden“

#### 4. Programmiersprachen

Pascal: 1971

Smalltalk: 1972

ELAN: 1974

Perl: 1987

Lua: 1993

Ruby: 1995

#### 5. Beschreibung

Gefunden auf: [https://de.wikipedia.org/wiki/Zeittafel\\_der\\_Programmiersprachen](https://de.wikipedia.org/wiki/Zeittafel_der_Programmiersprachen)

Pascal: [https://de.wikipedia.org/wiki/Pascal\\_\(Programmiersprache\)](https://de.wikipedia.org/wiki/Pascal_(Programmiersprache))  
<https://onecompiler.com/pascal>

ELAN: [https://de.wikipedia.org/wiki/ELAN\\_\(Programmiersprache\)](https://de.wikipedia.org/wiki/ELAN_(Programmiersprache))  
<https://elan-lang.org/beta/>

Smalltalk: [https://de.wikipedia.org/wiki/Smalltalk\\_\(Programmiersprache\)](https://de.wikipedia.org/wiki/Smalltalk_(Programmiersprache))  
[https://www.tutorialspoint.com/execute\\_smalltalk\\_online.php](https://www.tutorialspoint.com/execute_smalltalk_online.php)

Ruby: [https://de.wikipedia.org/wiki/Ruby\\_\(Programmiersprache\)](https://de.wikipedia.org/wiki/Ruby_(Programmiersprache))  
<https://onecompiler.com/ruby>

Perl: [https://de.wikipedia.org/wiki/Perl\\_\(Programmiersprache\)](https://de.wikipedia.org/wiki/Perl_(Programmiersprache))  
<https://onecompiler.com/perl>

Lua: <https://de.wikipedia.org/wiki/Lua>  
<https://onecompiler.com/lua>

## 6. Implementation

Perl war die am schnellsten verständliche Programmiersprache

Ruby und Lua haben auf Klammern und Semikolons verzichtet, was eine Umstellung von Java war

ELAN, Smalltalk und Pascal waren aufgrund ihres Alters schwerer logisch nachzuvollziehen, jedoch immer noch okay-isch zu verwenden

```
a = {}
for i=1, 1000 do
  a[i] = math.random(0,9)
  io.write(a[i])
end

io.write("\n")
io.write("\n Ausgabe")
io.write("\n")

for i = 1, 1000-2 do
  if a[i] == a[1] and a[i+1] == a[2] and a[i+2] == a[3] and i>1 then
    io.write("Zeitschleife gefunden an Stelle ", i, "\n")
    io.write(a[i], " | ", a[i+1], " | ", a[i+2], "\n")
  end
end

use warnings;
use strict;

my @array = (undef) x 1000;
my $schleife = 0;

for (my $i = 0; $i < @array; $i++) {
  $array[$i] = int(rand(10));
}

print "@array\n";

print("\n");

for (my $i = 0; $i < @array-2; $i++){
  if($array[$i] == $array[0] && $array[$i+1] == $array[1] && $array[$i+2] == $array[2] && $i != 0){
    print "Zeitschleife gefunden an Stelle $i\n";
    $schleife++;
  }
}

if($schleife==0){
  print "Keine Zeitschleife gefunden"
}

arra = Array.new(1000) { rand(9) }

puts arra.inspect

$i = 0
$num = 1000
$schleife = 0
while $i < $num do
  if arra[$i] == arra[0] && arra[$i+1] == arra[1] && arra[$i+2] == arra[2] && $i != 0
    print "Zeitschleife gefunden an der "
    print "Stelle #($i) mit dem Wert #($arra[$i]) "
    puts
    $schleife +=1
  end
  $i +=1
end
if $schleife == 0
  print "Keine Zeitschleife gefunden"
end
puts
```

## 7. Random Code & austricksen

Zahlengeneratoren befinden sich bereits in den Codesnippet.

Austricksen ist aufgrund unserer Definition und logischer Umsetzung nicht möglich!

Entweder die Zahlenfolge vom Anfang taucht wieder auf und es wird eine Zeitschleife erkannt oder nicht. Egal bei welchem Nummerninput (Ganzzahl von 0-9) kommt der Algorithmus auf ein richtiges Ergebnis.

## 8. Reflexion & Feedback auf eigener PDF

Max Hammerer (12424788), Ben Katz (12419322)

Ruby:

```
arra = Array.new(1000) { rand(9) }
```

```
puts arra.inspect
```

```
$i = 0
```

```
$num = 1000
```

```
$schleife = 0
```

```
while $i < $num do
```

```
  if arra[$i] == arra[0] && arra[$i+1] == arra[1] && arra[$i+2] == arra[2] && $i != 0
```

```
    print "Zeitschleife gefunden an der "
```

```
    print "Stelle #{ $i } mit dem Wert #{ arra[$i] } "
```

```
    puts
```

```
    $schleife +=1
```

```
  end
```

```
  $i +=1
```

```
end
```

```
if $schleife == 0
```

```
  print "Keine Zeitschleife gefunden"
```

```
end
```

```
puts
```

Max Hammerer (12424788), Ben Katz (12419322)

Perl:

```
use warnings;
```

```
use strict;
```

```
my @array = (undef) x 1000;
```

```
my $schleife = 0;
```

```
for (my $i = 0; $i < @array; $i++) {
```

```
    $array[$i] = int(rand(10));
```

```
}
```

```
print "@array\n";
```

```
print("\n");
```

```
for (my $i = 0; $i < @array-2; $i++){
```

```
    if($array[$i] == $array[0] && $array[$i+1] == $array[1] && $array[$i+2] == $array[2] && $i != 0){
```

```
        print "Zeitschleife gefunden an Stelle $i \n";
```

```
        $schleife ++;
```

```
    }
```

```
}
```

```
if($schleife==0){
```

```
    print "Keine Zeitschleife gefunden"
```

```
}
```

Max Hammerer (12424788), Ben Katz (12419322)

Lua:

```
a = {}
```

```
for i=1, 1000 do
```

```
    a[i] = math.random(0,9)
```

```
    io.write(a[i])
```

```
end
```

```
io.write("\n")
```

```
io.write("\n Ausgabe")
```

```
io.write("\n")
```

```
for i = 1, 1000-2 do
```

```
    if a[i] == a[1] and a[i+1] == a[2] and a[i+2] == a[3] and i>1 then
```

```
        io.write("Zeitschleife gefunden an Stelle ", i, "\n")
```

```
        io.write(a[i], " | ", a[i+1], " | ", a[i+2], "\n" )
```

```
    end
```

```
end
```

Max Hammerer (12424788), Ben Katz (12419322)

Smalltalk:

| arr Zeitschleifen|

arr := Array new: 1000.

randy := Random new.

1 to: 1000 do: [:i |

arr at: i put: ((randy next)\* 9) floor + 1.

].

1 to: 1000 do: [:i |

(arr at: i = arr at: 1 and: [ arr at: i+1 = arr at: 2]) and: [ arr at: i+2 = arr at: 3] and: [i>1]

ifTrue: [ Transcript show: ' gefunden'; cr ]

ifFalse: [ ].

].

Transcript show: 'Zeitschleife: ', arr printString; cr.

Max Hammerer (12424788), Ben Katz (12419322)

ELAN:

VAR

arr: ARRAY [1..1000] OF INTEGER;

FOR i := 1 TO 1000 DO

arr[i] := RANDOM(0, 10)

;

END;

FOR i := 1 TO 1000 DO

IF (arr[i] = arr[1]) AND (arr[i+1] = arr[2]) AND (arr[i+2] = arr[3]) AND (i>1) THEN

WRITE("Zeitschleife gefunden");

END;



Max Hammerer (12424788), Ben Katz (12419322)

Pascal:

```
program Zeischleife;
```

```
uses SysUtils;
```

```
var
```

```
    zufall: Integer;
```

```
var
```

```
    arr: array[1..1000] of Integer; { Ein Array mit 1000 Integer-Werten }
```

```
    i: Integer;
```

```
begin
```

```
    Randomize; {initialisieren des Randoms}
```

```
    writeln ('Zeischleife suchen gestartet');
```

```
    { Array initialisieren }
```

```
    for i := 1 to 1000 do
```

```
    begin
```

```
        arr[i] := Random(10);
```

```
        Write(arr[i]);
```

```
    end;
```

```
    writeln("");
```

```
    { Array ausgeben }
```

```
    for i := 1 to 1000 do
```

```
        if (arr[i] = arr[1]) and (arr[i+1] = arr[2]) and (arr[i+2] = arr[3]) and (i>1) then
```

```
        begin
```

```
            WriteLn( 'Zeischleife gefunden an Stelle ',i);
```

```
        end
```

```
end.
```