Arduino – Programmierpraxis mit Feldvariablen

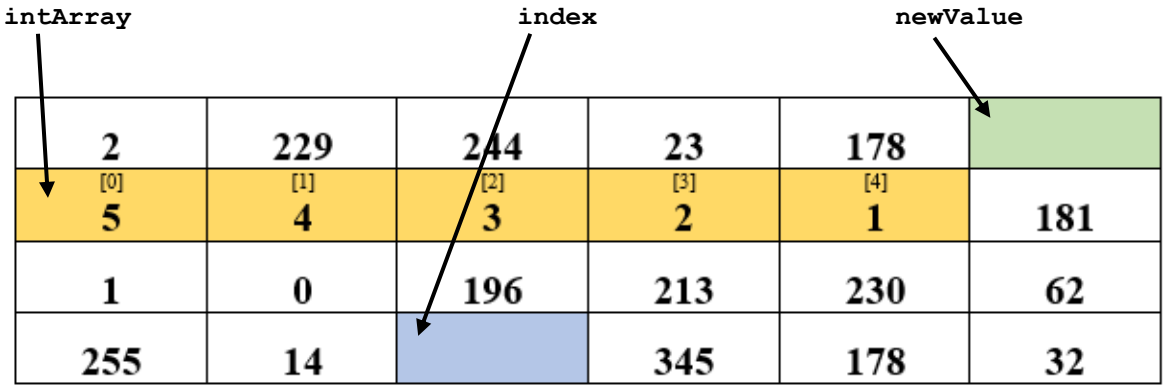
Arbeitsanregungen zu Grundlagen der Programmierung mit Feldvariablen

Aufgabe 1)

a) Analyse des nebenstehend abgebildeten Programmcodes (mit : **`int-Array`** als Array der Größe 5):

```
void loop() {
  int index;
  int newValue;

  index = 0;
  while(index < 4){
    newValue = intArray[index] + intArray[index + 1];
    Serial.print(" neuer Wert: ");
    Serial.println(newValue);
    index = index + 1;
  }
  Serial.println("");
  delay(5000);
}
```

Situation vor der Ausführung der Programmschleife (Vorgabe samt Eintrag für den Startwert der Variablen **`index`**):



2	229	244	23	178	
[0] 5	[1] 4	[2] 3	[3] 2	[4] 1	181
1	0	196	213	230	62
255	14		345	178	32

Eintragen des Startwertes der Variablen **`index`**:

2	229	244	23	178	
[0] 5	[1] 4	[2] 3	[3] 2	[4] 1	181
1	0	196	213	230	62
255	14	0	345	178	32

CC BY-NC-SA 4.0 Informatik-Werkstatt AAU 2022
Informatikwerkstatt.aau.at

Die Variable **index** hat zunächst den Wert __0__,

daher wird die Schleife

```
while(index < 4){
  newValue = intArray[index] + intArray[index + 1];
  Serial.print(" neuer Wert: ");
  Serial.println(newValue);
  index = index + 1;
}
```

_____ausgeführt_____ (ausgeführt/nicht ausgeführt)

…zuerst wird gerechnet (`intArray[index] + intArray[index + 1];`):

Berechnung:

5 + 4

2	229	244	23	178	
[0] 5	[1] 4	[2] 3	[3] 2	[4] 1	181
1	0	196	213	230	62
255	14	0	345	178	32

… dann (`newValue = intArray[index] + intArray[index + 1];`) wird das errechnete Ergebnis in der Variablen **newValue** gespeichert und (`Serial.println(newValue);`) ausgegeben):

Ergebnis:

9

2	229	244	23	178	9
[0] 5	[1] 4	[2] 3	[3] 2	[4] 1	181
1	0	196	213	230	62
255	14	0	345	178	32

Ausgabe:

9

…und (`index = index + 1;`) der Wert der Variablen **index** verändert:

2	229	244	23	178	9
[0] 5	[1] 4	[2] 3	[3] 2	[4] 1	181
1	0	196	213	230	62
255	14	~~0~~ 1	345	178	32

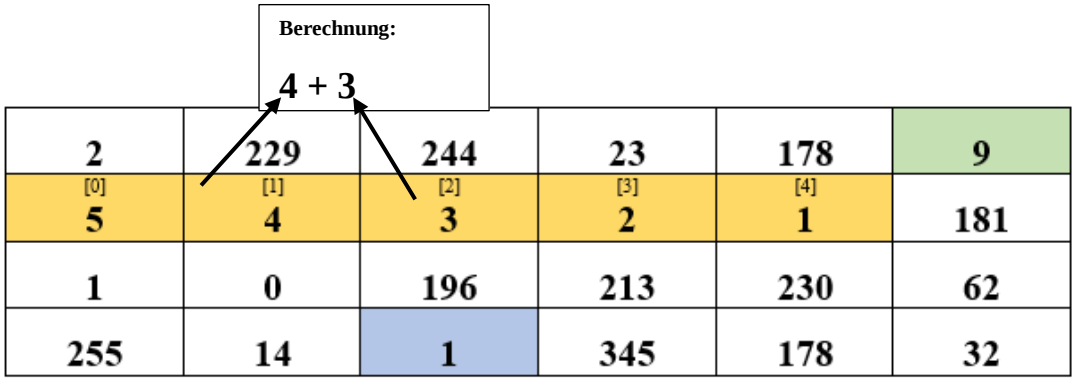
Die Variable **index** hat nun den Wert ___1___,

daher wird die Schleife

```
while(index < 4){
  newValue = intArray[index] + intArray[index + 1];
  Serial.print(" neuer Wert: ");
  Serial.println(newValue);
  index = index + 1;
}
```

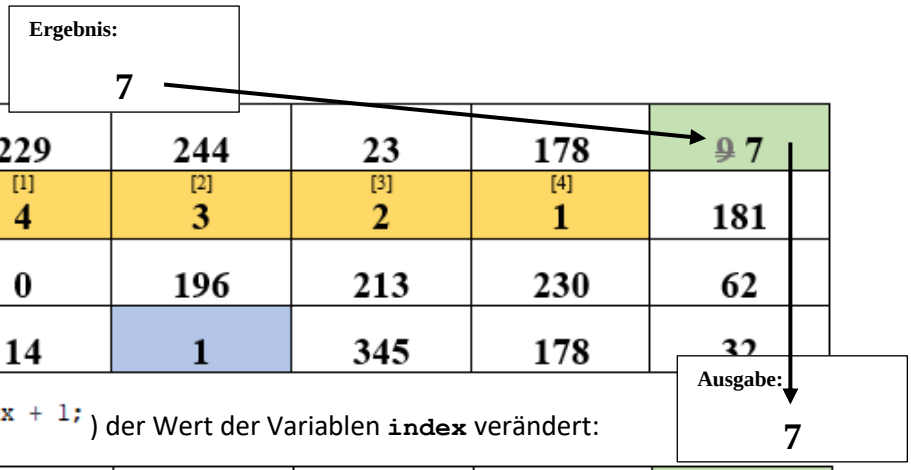
_____ausgeführt_____ (ausgeführt/nicht ausgeführt)

…zuerst wird gerechnet (`intArray[index] + intArray[index + 1];`):



2	229	244	23	178	9
[0] 5	[1] 4	[2] 3	[3] 2	[4] 1	181
1	0	196	213	230	62
255	14	1	345	178	32

… dann (`newValue = intArray[index] + intArray[index + 1];`) wird das errechnete Ergebnis in der Variablen **newValue** gespeichert und (`Serial.println(newValue);`) ausgegeben):



2	229	244	23	178	~~9~~ 7
[0] 5	[1] 4	[2] 3	[3] 2	[4] 1	181
1	0	196	213	230	62
255	14	1	345	178	32

…und (`index = index + 1;`) der Wert der Variablen **index** verändert:

2	229	244	23	178	7
[0] 5	[1] 4	[2] 3	[3] 2	[4] 1	181
1	0	196	213	230	62
255	14	~~1~~ 2	345	178	32

Die Variable **index** hat nun den Wert ___2__,

daher wird die Schleife

```
while(index < 4){
  newValue = intArray[index] + intArray[index + 1];
  Serial.print(" neuer Wert: ");
  Serial.println(newValue);
  index = index + 1;
}
```

_____ausgeführt_____ (ausgeführt/nicht ausgeführt)

…zuerst wird gerechnet (`intArray[index] + intArray[index + 1];`):

Berechnung:

3 + 2

2	229	244	23	178	7
[0] 5	[1] 4	[2] 3	[3] 2	[4] 1	181
1	0	196	213	230	62
255	14	2	345	178	32

… dann (`newValue = intArray[index] + intArray[index + 1];`) wird das errechnete Ergebnis in der Variablen **newValue** gespeichert und (`Serial.println(newValue);`) ausgegeben):

Ergebnis:

5

2	229	244	23	178	~~7~~ 5
[0] 5	[1] 4	[2] 3	[3] 2	[4] 1	181
1	0	196	213	230	62
255	14	2	345	178	32

Ausgabe:

5

…und (`index = index + 1;`) der Wert der Variablen **index** verändert:

2	229	244	23	178	5
[0] 5	[1] 4	[2] 3	[3] 2	[4] 1	181
1	0	196	213	230	62
255	14	~~2~~ 3	345	178	32

Die Variable **index** hat nun den Wert ___3___,

daher wird die Schleife

```
while(index < 4){
  newValue = intArray[index] + intArray[index + 1];
  Serial.print(" neuer Wert: ");
  Serial.println(newValue);
  index = index + 1;
}
```

_____ausgeführt_____ (ausgeführt/nicht ausgeführt)

…zuerst wird gerechnet (`intArray[index] + intArray[index + 1];`):

Berechnung: 2 + 1

2	229	244	23	178	5
[0] 5	[1] 4	[2] 3	[3] 2	[4] 1	181
1	0	196	213	230	62
255	14	3	345	178	32

… dann (`newValue = intArray[index] + intArray[index + 1];`) wird das errechnete Ergebnis in der Variablen **newValue** gespeichert und (`Serial.println(newValue);`) ausgegeben):

Ergebnis: 3

2	229	244	23	178	~~5~~ 3
[0] 5	[1] 4	[2] 3	[3] 2	[4] 1	181
1	0	196	213	230	62
255	14	3	345	178	32

Ausgabe: 3

…und (`index = index + 1;`) der Wert der Variablen **index** verändert:

2	229	244	23	178	3
[0] 5	[1] 4	[2] 3	[3] 2	[4] 1	181
1	0	196	213	230	62
255	14	~~3~~ 4	345	178	32

Die Variable **index** hat nun den Wert ___4___,

daher wird die Schleife

```
while(index < 4){
  newValue = intArray[index] + intArray[index + 1];
  Serial.print(" neuer Wert: ");
  Serial.println(newValue);
  index = index + 1;
}
```

____nicht ausgeführt_____ (ausgeführt/nicht ausgeführt)

b)

Analyse des abgebildeten Codeteils mit Hilfe einer Tabelle (Durchlauf Nr. 0 bezeichnet die Situation vor dem ersten Durchlauf):

```
void loop() {
  int index;
  int newValue;

  index = 0;
  while(index < 4){
    newValue = intArray[index] + intArray[index + 1];
    Serial.print(" neuer Wert: ");
    Serial.println(newValue);
    index = index + 1;
  }
  Serial.println("");
  delay(5000);
}
```

Durchlauf Nr.	index	intArray	newValue
0	0	{ 5, 4, 3, 2, 1 }	
1	0 → 1	{ **5**, **4**, 3, 2, 1 }	9
2	1 → 2	{ 5, **4**, **3,** 2, 1 }	7
3	2 → 3	{ 5, 4, **3**, **2**, 1 }	5
4	3 → 4	{ 5, 4, 3, **2**, **1** }	3
~~5~~	~~4~~	~~{ 5, 4, 3, 2, 1 }~~	~~3~~
…der Wert von **index** ist nicht mehr kleiner als __4__, ein fünfter Schleifendurchlauf erfolgt daher nicht mehr…			

Aufgabe 2)

a)

```
int intArray[4];

void setup() {
  Serial.begin(9600); //für die Ausgabe
  int index;

  index = 0;
  while(index < 4){
    intArray[index] = 2 * (index + 1);
    index = index + 1;
  }
}
```

Analyse des abgebildeten `setup`-Codeteils mit Hilfe einer Tabelle (Durchlauf Nr. 0 bezeichnet die Situation vor dem ersten Durchlauf):

Durchlauf Nr.	index	intArray	Bemerkung
0	0	{ 0, 0, 0, 0 }	**intArray** wird außerhalb aller Befehle oder Fragen vereinbart und ist daher eine globale Array-Variable vom Typ **int**. Eine solche wird standardmässig mit dem Wert 0 „befüllt“ (initialisiert)
1	0 → 1	{ **2**, 0, 0, 0 }	
2	1 → 2	{ 2, **4**, 0, 0 }	
3	2 → 3	{ 2, 4, **6**, 0 }	
4	3 → 4	{ 2, 4, 6, **8** }	
~~5~~	~~**4**~~	~~{ 2, 0, 0, 8 }~~	
…der Wert von **index** ist nicht mehr kleiner als __4__, ein fünfter Schleifendurchlauf erfolgt daher nicht mehr…			

```
void loop() {
  int index;
  int ergebnis;

  ergebnis = 1;
  index = 0;
  while(index < 4){
    ergebnis = ergebnis * intArray[index];
    index = index + 1;
  }
  Serial.print(" Ergebnis: ");
  Serial.println(ergebnis);
  Serial.println("");
  delay(5000);
}
```

Analyse der Berechnungen im abgebildeten `loop`-Programmblock (Fortsetzung von obigem `setup`-Programmblock) mit Hilfe einer Tabelle (Durchlauf Nr. 0 bezeichnet wieder die Situation vor dem ersten Durchlauf):

Durchlauf Nr.	index	intArray	ergebnis
0	0	{ 2, 4, 6, 8 }	1
1	0 → 1	{ **2**, 4, 6, 8 }	1 · 2 → 2
2	1 → 2	{ 2, **4**, 6, 8 }	2 · 4 → 8
3	2 → 3	{ 2, 4, **6**, 8 }	8 · 6 → 48
4	3 → 4	{ 2, 4, 6, **8** }	48 · 8 → 384
~~5~~	~~**4**~~	~~{ 2, 4, 6, 8 }~~	~~384~~
…der Wert von **index** ist nicht mehr kleiner als __4__, ein fünfter Schleifendurchlauf erfolgt daher nicht mehr…			

b)

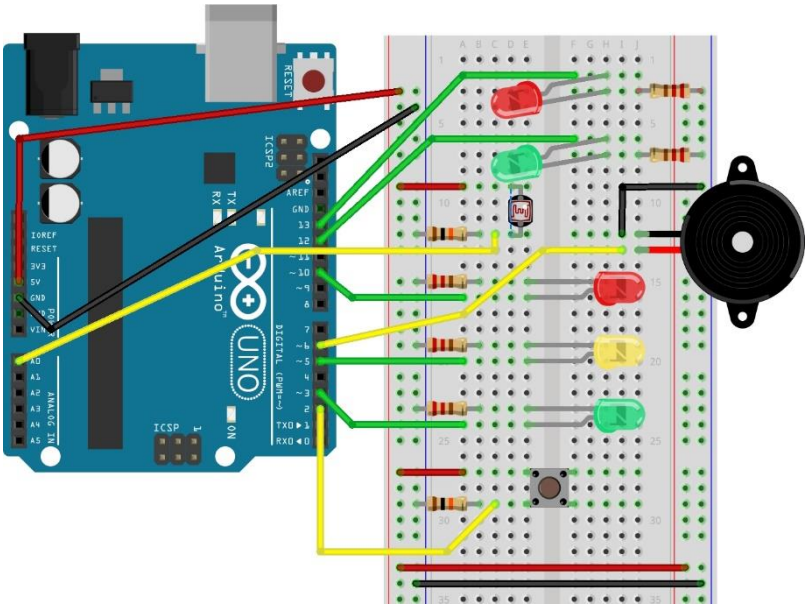
```
void loop() {
  int index;
  int zwErg;
  float ergebnis;

  zwErg = 0;
  index = 0;
  while(index < 4){
    zwErg = zwErg + intArray[index];
    index = index + 1;
  }
  ergebnis = zwErg/(float)4;
  Serial.print(" Ergebnis: ");
  Serial.println(ergebnis);
  Serial.println("");
  delay(5000);
}
```

Analyse der Berechnungen im abgebildeten `loop`-Programmblock (alternative Fortsetzung des `setup`-Programmblocks aus Aufgabenteil **a)**) mit Hilfe einer Tabelle (Durchlauf Nr. 0 bezeichnet wieder die Situation vor dem ersten Durchlauf):

Durchlauf Nr.	index	intArray	zwErg	ergebnis
0	0	{ 2, 4, 6, 8 }	0	
1	0 → 1	{ **2**, 4, 6, 8 }	0 + 2 → 2	
2	1 → 2	{ 2, **4**, 6, 8 }	2 + 4 → 6	
3	2 → 3	{ 2, 4, **6**, 8 }	4 + 6 → 10	
4	3 → 4	{ 2, 4, 6, **8** }	10 + 8 → 18	
…der Wert von **index** ist nicht mehr kleiner als 4, ….				
~~5~~	~~**4**~~	~~{ 2, 4, 6, 8 }~~		18 / 4 → 4.5

Aufgabe 3)

Die nachfolgenden Codeblöcke können auf einem Arduino-Board OHNE zusätzliche Steckplatine ausgeführt werden, es kann aber auch die nebenstehende (bekannte) Arduino-Ampelschaltung verwendet werden, die für die Bearbeitung der Arbeitsanregungen aus Abschnitt 11 wieder bedeutsam wird:



a) Codevorschlag:

```
int intArray[100];

void setup() {
  Serial.begin(9600);
  int index;
  index = 0;
  randomSeed(analogRead(A1));
  while(index < 100){
    intArray[index] = random(1,1001);
    index = index + 1;
  }
}
```

```
void printIntArray(int valArray[], int dim){
  int index;

  index = 0;
  while(index < dim){
    Serial.print("Wert an Indexposition ");
    Serial.print(index);
    Serial.print(": ");
    Serial.println(intArray[index]);
    index = index + 1;
    delay(1000);
  }
}
void loop() {
    //Ausgabe des int-Feldes
    printIntArray(intArray,100);
}
```

b) Codevorschlag zur Berechnung der Summe aller im Array gespeicherten `int`-Werte (in starker Anlehnung an das Codebeispiel aus der Informationsdatei **ST_I_10Arduino_Praxis_Feldvariable**):

```
long frageIntArrSum(int valArray[], int dim){
  int index;
  long antwort;

  index = 0;
  antwort = 0;
  while(index < dim){
    antwort = antwort + valArray[index];
    index = index + 1;
  }
  return antwort;
}
```

…und die Modifikation des `loop`-Programmblocks:

```
void loop() {
  long sum;

  //Berechnen der Summe der Feldwerte
    sum = frageIntArrSum(intArray,100);

  //Ausgabe des int-Feldes
    printIntArray(intArray,100);

    //...und der Summe
    Serial.println("");
    Serial.print("Summe: ");
    Serial.println(sum);
}
```

c) Codevorschlag zur Berechnung des Mittelwerts aller im Array gespeicherten `int`-Werte (unter Verwendung des Frage-Blocks zur Berechnung der Summe aus Aufgabenteil **b)**):

```
float frageIntArrMean(int valArray[], int dim){
  long sum;
  float antwort;

  antwort = 0;
  sum = frageIntArrSum(valArray,dim);
  antwort = sum/(float)dim;
  return antwort;
}
```

…und die Modifikation des `loop`-Programmblocks:

```
void loop() {
  long sum;
  float mean;

  //Berechnen der Summe der Feldwerte
    sum = frageIntArrSum(intArray,100);

  //Berechnen des Mittelwerts der Feldwerte
    mean = frageIntArrMean(intArray,100);

  //Ausgabe des int-Feldes
    printIntArray(intArray,100);

  //...und der Summe
    Serial.println("");
    Serial.print("Summe: ");
    Serial.println(sum);

  //...und des Mittelwerts
    Serial.println("");
    Serial.print("Mittelwert: ");
    Serial.println(mean);
}
```

d) Codevorschlag zur Veränderung des Befehlsblocks zur Ausgabe aller Werte einer `int`-Feldvariablen…

…als „codeintensive“ Variante:

```
void printIntArray(int valArray[], int dim, int dir){
  int index;

  if(dir == 0){
    index = 0;
    while(index < dim){
      Serial.print("Wert an Indexposition ");
      Serial.print(index);
      Serial.print(": ");
      Serial.println(intArray[index]);
      index = index + 1;
      delay(1000);
    }
  }
  else{
    index = dim-1;
    while(index >= 0){
      Serial.print("Wert an Indexposition ");
      Serial.print(index);
      Serial.print(": ");
      Serial.println(intArray[index]);
      index = index - 1;
      delay(1000);
    }
  }
}
```

…als „schlankere“ Variante:

```
void printIntArray(int valArray[], int dim, int dir){
  int index;
  int stopInd;
  int stepInd;

  if(dir == 0){
    index = 0;
    stopInd = dim;
    stepInd = 1;
  }
  else{
    index = dim - 1;
    stopInd = -1;
    stepInd = -1;
  }
  while(index != stopInd){
      Serial.print("Wert an Indexposition ");
      Serial.print(index);
      Serial.print(": ");
      Serial.println(intArray[index]);
      index = index + stepInd;
      delay(1000);
  }
}
```

…und die Modifikation des `loop`-Programmblocks:

```
void loop() {
  long sum;
  float mean;

  //Berechnen der Summe der Feldwerte
    sum = frageIntArrSum(intArray,100);

  //Berechnen des Mittelwerts der Feldwerte
    mean = frageIntArrMean(intArray,100);

  //Ausgabe des int-Feldes
    printIntArray(intArray,100,0);

  //bzw. alternativ:
    printIntArray(intArray,100,1);

  //...und der Summe
    Serial.println("");
    Serial.print("Summe: ");
    Serial.println(sum);

  //...und des Mittelwerts
    Serial.println("");
    Serial.print("Mittelwert: ");
    Serial.println(mean);
}
```