WBS TRAINING

Programmierung(2)

UNSER ♥ SCHLÄGT FÜR BILDUNG

Agenda

- 2-dimensionale Arrays
  - **Definition, Motivation** und Beispiel
  - Darstellung im **PAP**
  - Darstellung im **Struktogramm**
  - Darstellung im **Pseudocode**
  - Syntax in **ANSI C**
- String-Arrays
  - **Definition, Motivation** und Beispiel
  - Darstellung im **PAP**
  - Darstellung im **Struktogramm**
  - Darstellung im **Pseudocode**
  - Syntax in **ANSI C**
- Fachpraktische Anwendungen

2-dimensionale Arrays – **Definition** und **Motivation**

- Wir haben uns bereits mit 1-dimensionalen Arrays beschäftigt. Für deren Motivation gab es die folgende Gedankenkette:

  Programme verlangen oft das **mehrfache hintereinander-Ausführen** von identischen (oder ähnlichen) Code-Abschnitten

  ⇒ durch den Einsatz von **Schleifen** kann das wiederholte Notieren dieser Abschnitte vermieden werden

  ⇒ falls pr**o Schleifendurchlauf jeweils neue Variablen** zu verwenden sind, so müssen diese jedoch „sortiert“ sein

  ⇒ dies führte zu der Einführung von **Arrays**, die eine **Ansammlung durchnummerierter Variablen** darstellen

- Wir können diesen Gedankengang nun auch für jene Fälle aufgreifen, bei denen innerhalb eines Programmes **mehrere Arrays nacheinander** abgearbeitet werden sollen:

  Erneut bietet sich der Einsatz von **Schleifen** an.

  ⇒ pro Durchlauf wird **jeweils ein neues Array** verwendet, auch diese müssen also „sortiert“ sein

  ⇒ entsprechend werden wir daher auch die Arrays **durchnummerieren**

  ⇒ es entsteht ein **2-dimensionales Array**:

  ⇒ die erste Dimension zählt die „**Array-Member“**

  ⇒ Die zweite Dimension zählt die **Elemente** pro Array-Member

  ⇒ int arr[**5**][**10**] definiert ein 2-dimensionales Integer-Array (**5** Arrays mit jeweils **10** Elementen)

  ⇒ arr[0][0] ist dann entsprechend das erste Element im ersten Array

  ⇒ arr[4][9] ist dann entsprechend das letzte Element im letzten Array

2-dimensionale Arrays – **Beispielaufgabe**

Vorbemerkung:
- Ein Ladenbesitzer besitzt **10 Filialen**.
- Pro Filiale sind die **12 Monatsumsätze** bekannt.
- Der Umsatz der **x-ten Filiale** des **y-ten Monats** soll in **arr[x][y]** abgespeichert werden (x ist ein Wert zwischen 0 und 9 / y zwischen 0 und 11)

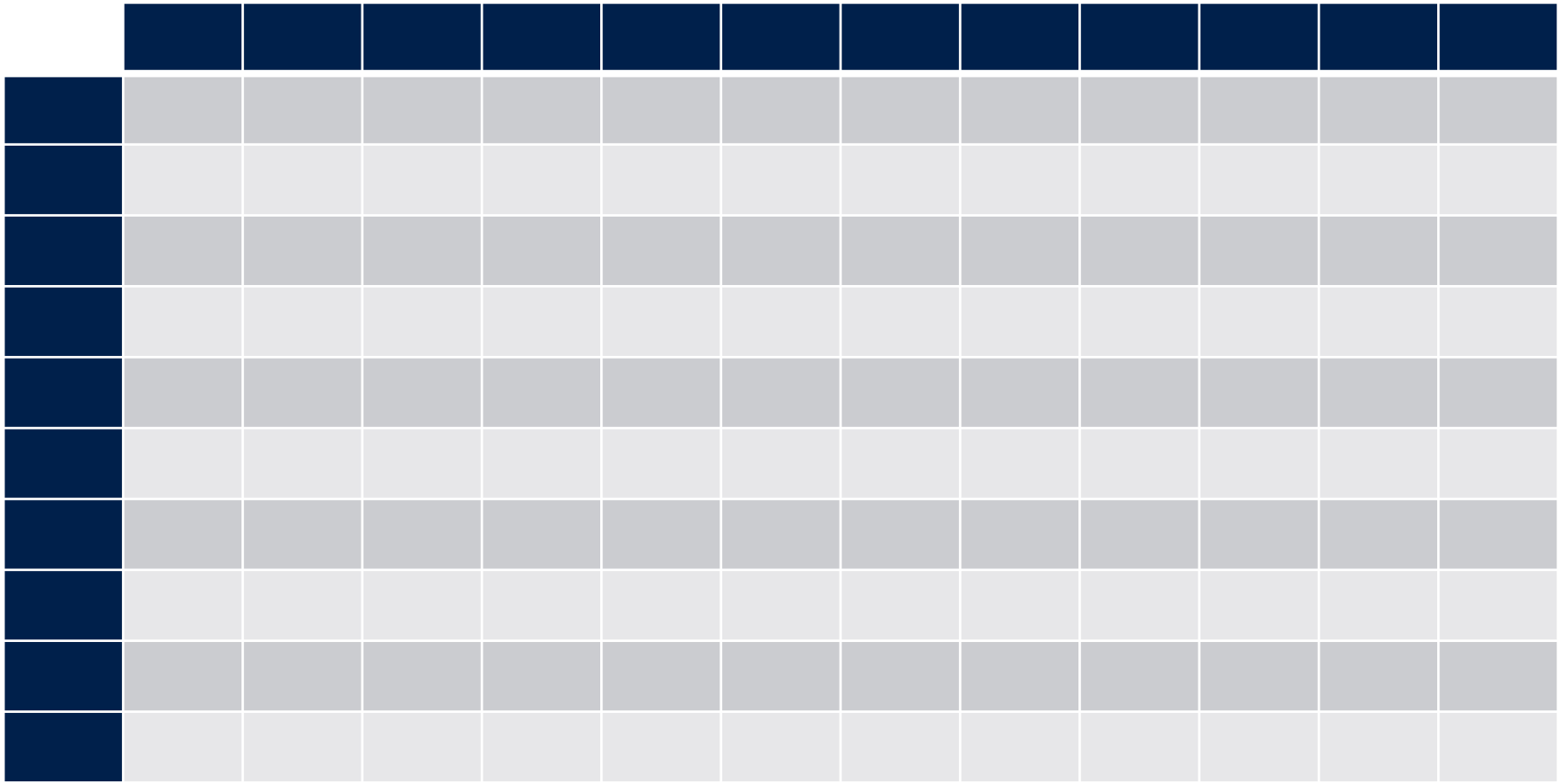
Aufgabenstellung

- Das Programm startet mit einer äußeren Schleife, die 10-mal durchlaufen wird. Pro Durchlauf …
  - wird eine Innere Schleife gestartet, die 12-mal durchlaufen wird. Pro Durchlauf …
    - wird ein Float-Wert abgefragt
    - Wird der Eingabewert im Array abgespeichert
  - Nach der Inneren Schleife endet der aktuelle Durchlauf der äußeren Schleife
- Nach der äußeren Schleife startet eine weitere verschachtelte Schleife. In dieser werden zur Kontrolle alle Element-Inhalte des 2-dimensionalen Arrays auf der Konsole wie folgt ausgegeben:

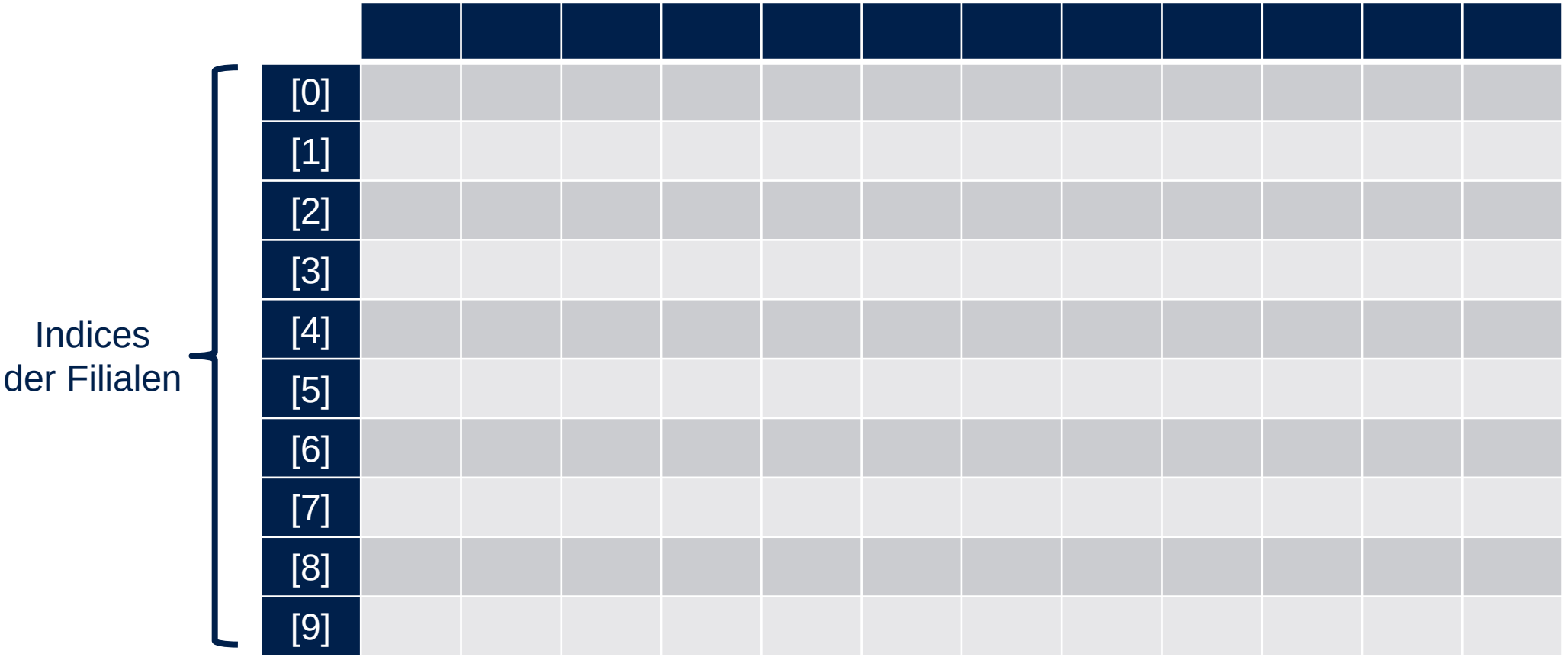
„In arr[0][0] ist der Umsatz des 1. Monats der 1. Filiale abgespeichert, er beträgt: …“
„In arr[0][1] ist der Umsatz des 2. Monats der 1. Filiale abgespeichert, er beträgt: …“
…
„In arr[1][0] ist der Umsatz des 1. Monats der 2. Filiale abgespeichert, er beträgt: …“
„In arr[1][1] ist der Umsatz des 2. Monats der 2. Filiale abgespeichert, er beträgt: …“
…

Auch für diese Aufgabe wollen wir zunächst **PAP**, **Struktogramm** und **Pseudocode** erstellen, um erst daraufhin den entsprechenden **Quellcode** in ANSI C zu codieren.

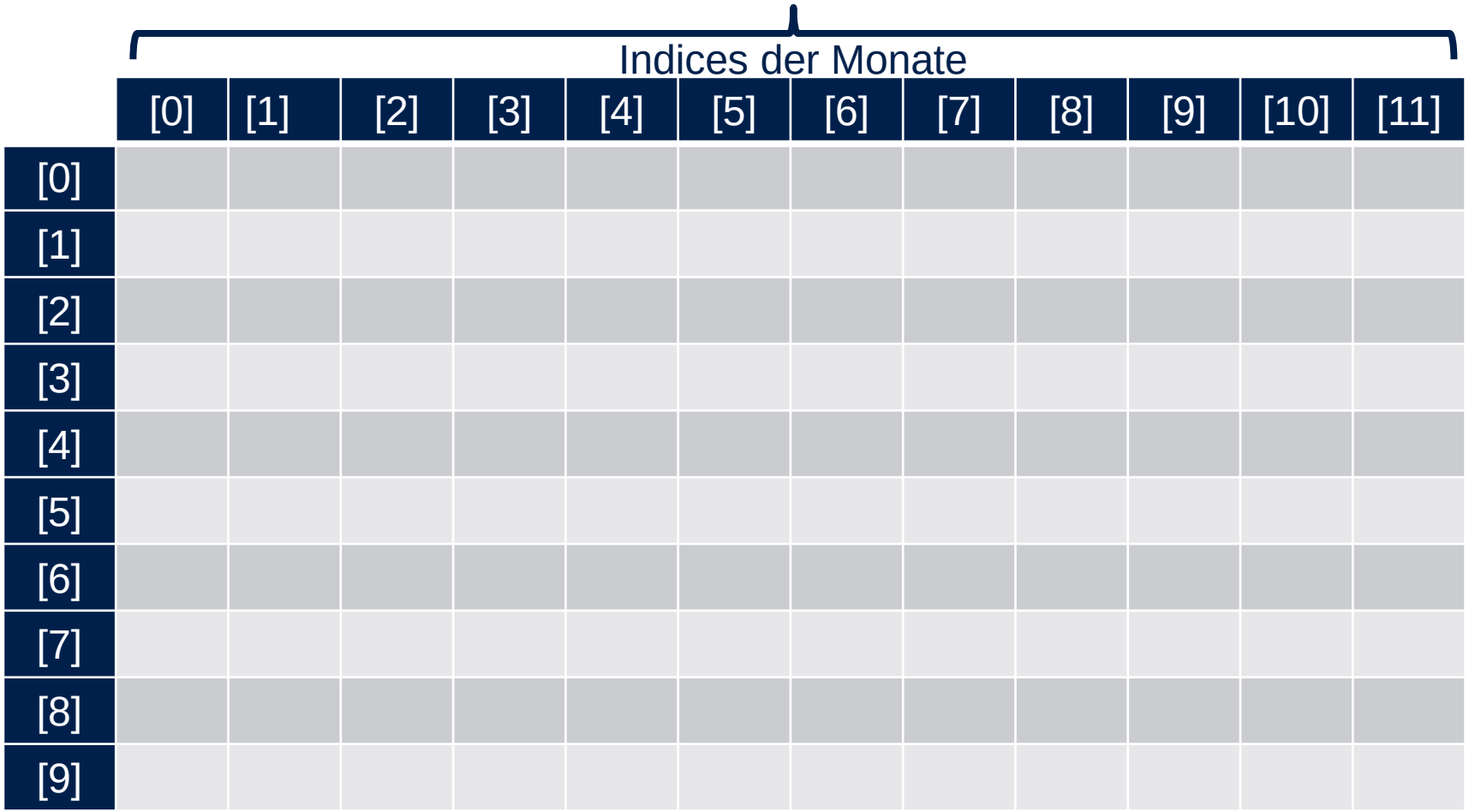
2-dimensionale Arrays – 2-dimensionale Veranschaulichung



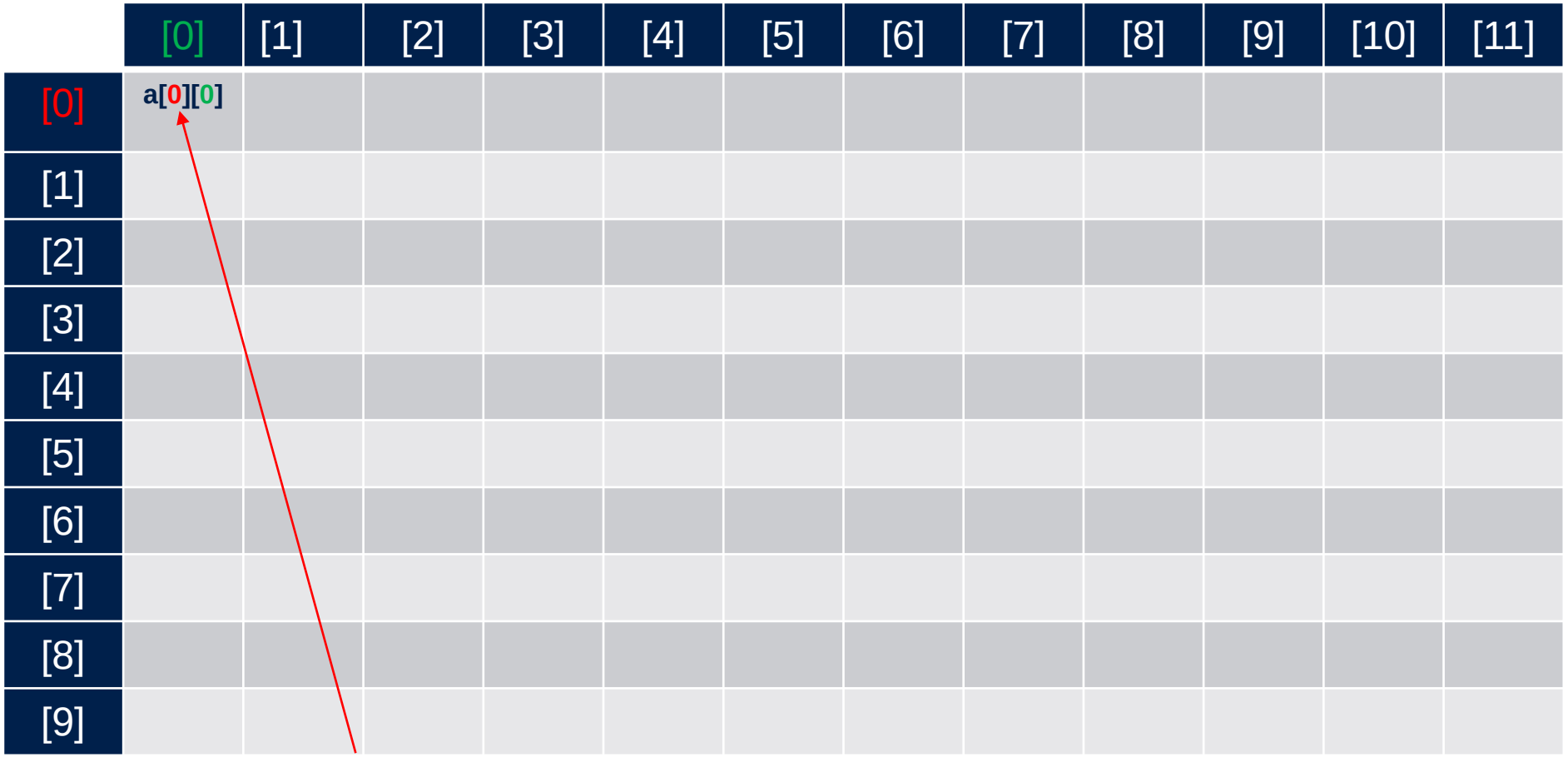
2-dimensionale Arrays – 2-dimensionale Veranschaulichung



2-dimensionale Arrays – **2-dimensionale Veranschaulichung**

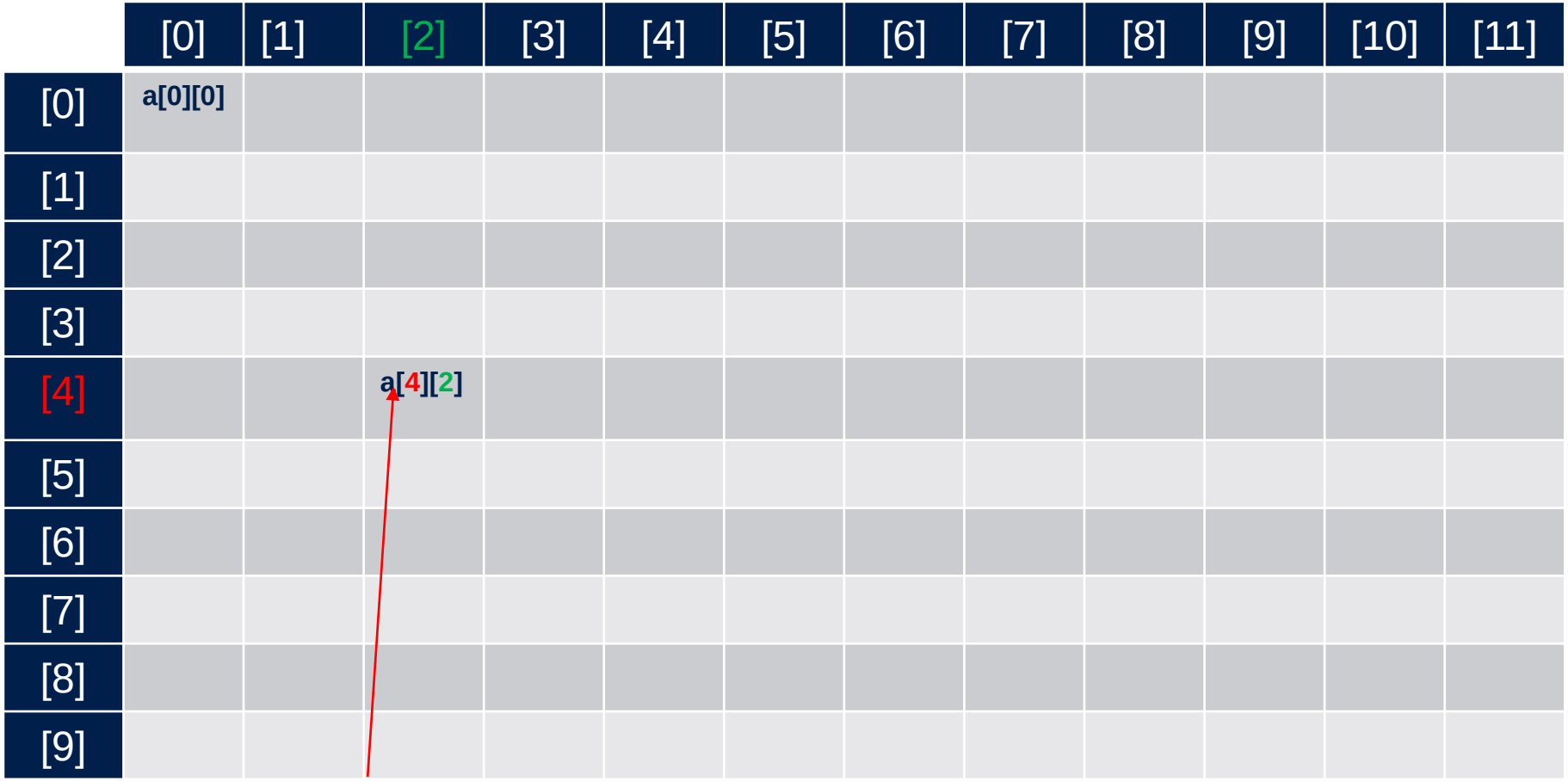


2-dimensionale Arrays – 2-dimensionale Veranschaulichung



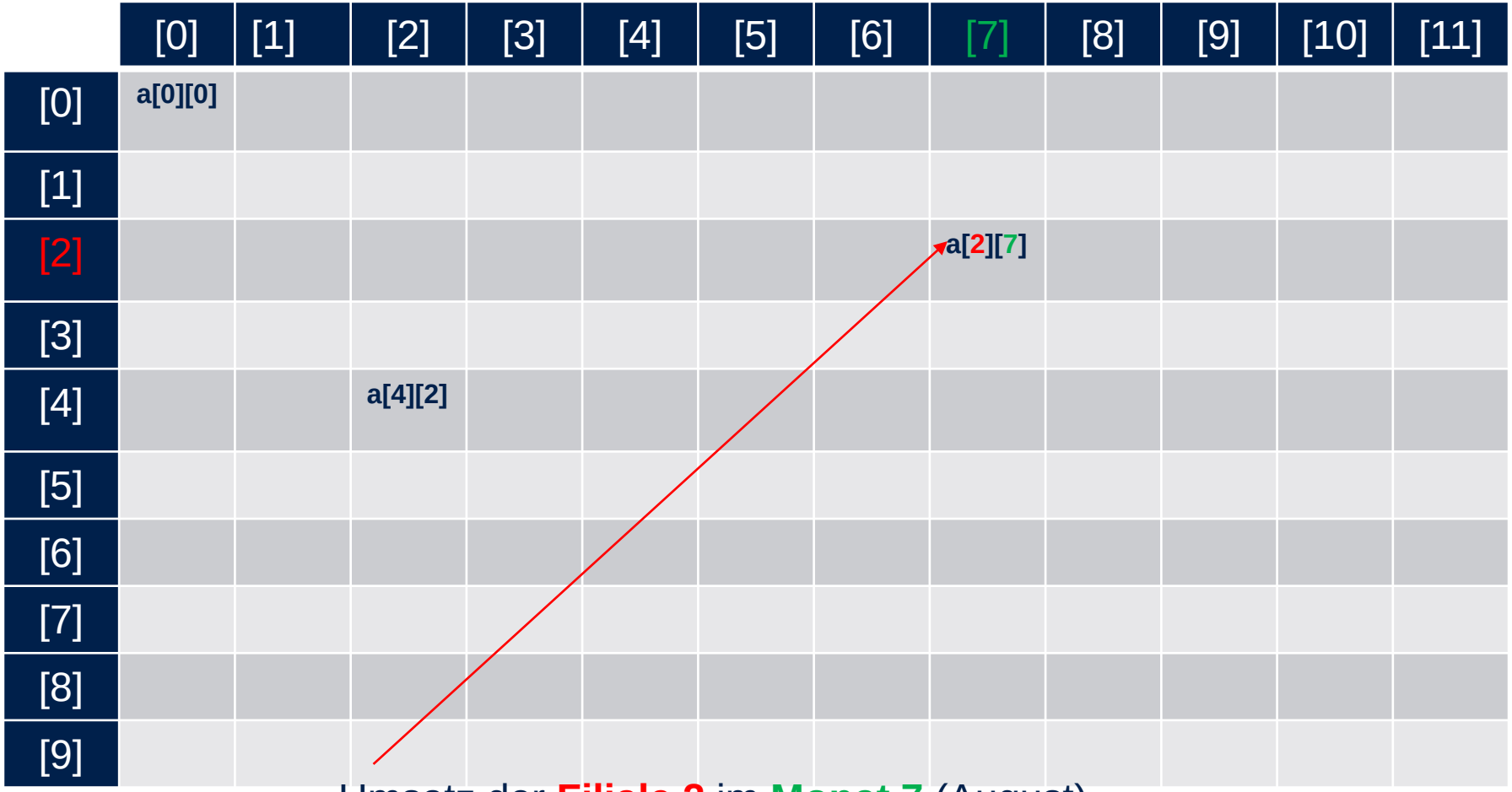
Umsatz der **Filiale 0** im **Monat 0** (Januar)

2-dimensionale Arrays – 2-dimensionale Veranschaulichung



Umsatz der **Filiale 4** im **Monat 2** (März)

2-dimensionale Arrays – 2-dimensionale Veranschaulichung

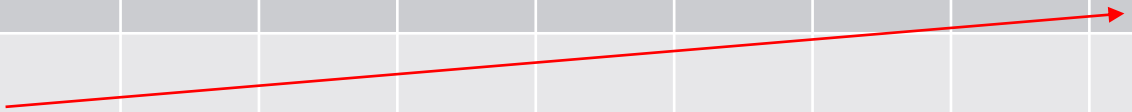


Umsatz der **Filiale 2** im **Monat 7** (August)

WBS TRAINING

2-dimensionale Arrays – 2-dimensionale Veranschaulichung

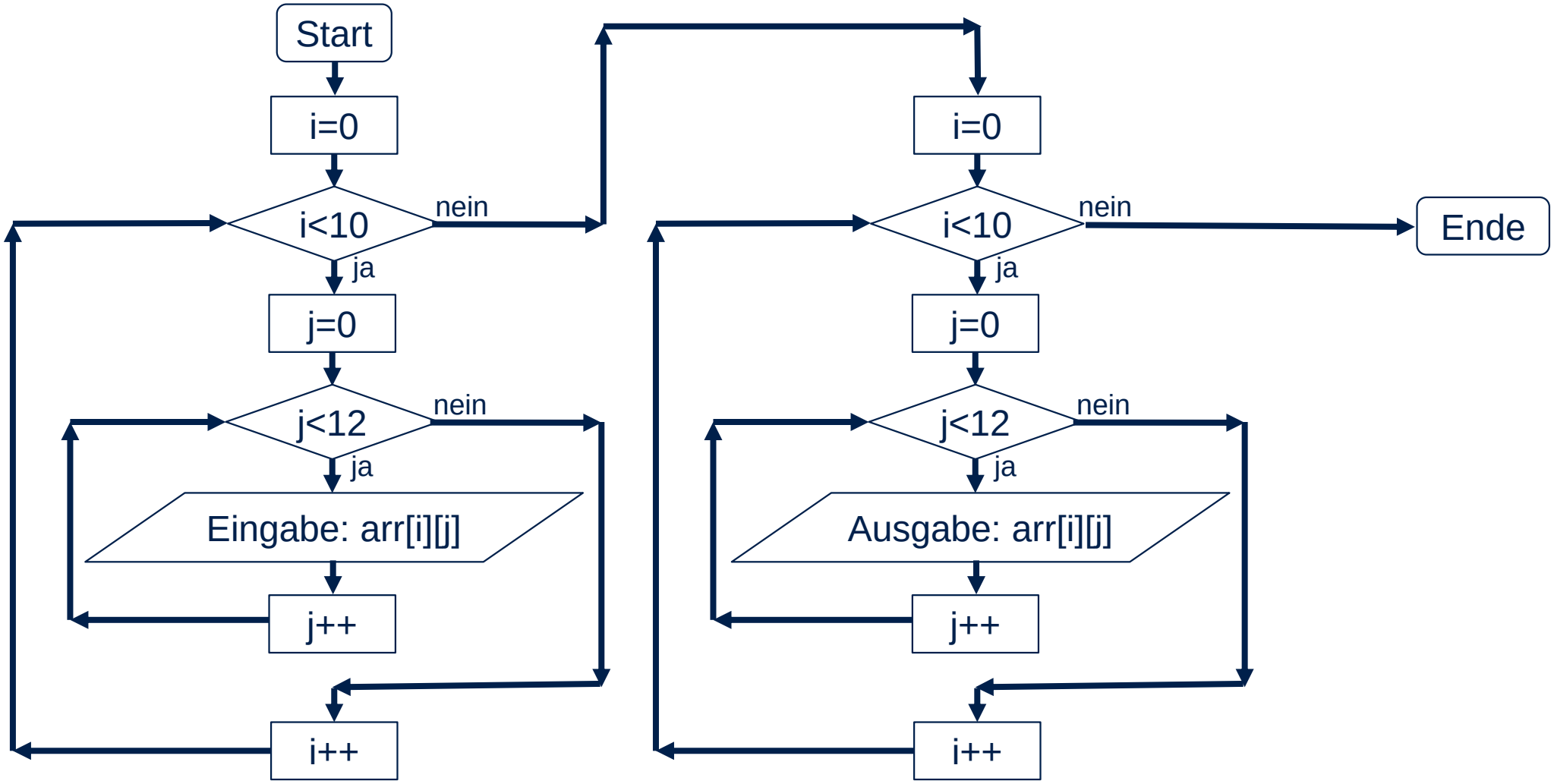
	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]	[11]
[0]	a[0][0]											
[1]												
[2]								a[2][7]				
[3]												
[4]			a[4][2]									
[5]												
[6]												
[7]												
[8]												
[9]												a[9][11]



Umsatz der **Filiale 9** im **Monat 11** (Dezember)

WBS TRAINING

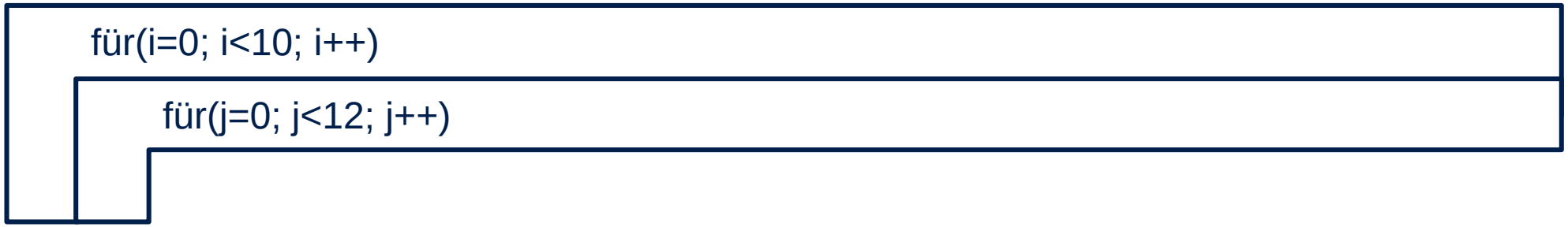
2-dimensionale Arrays – **Beispielaufgabe** – PAP



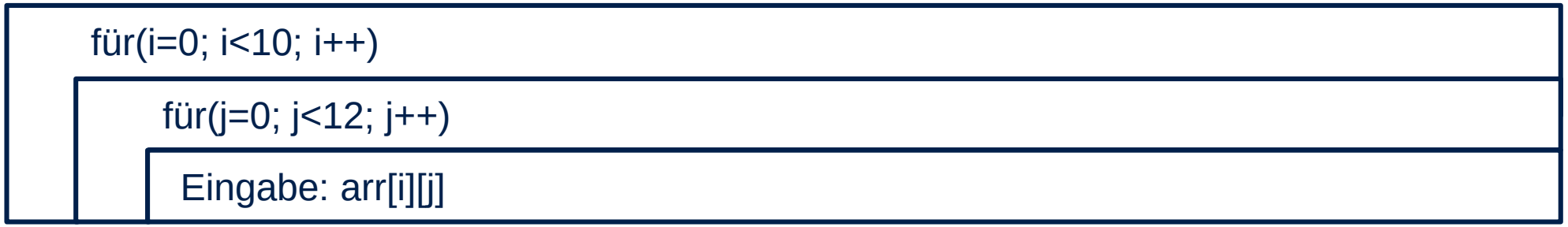
2-dimensionale Arrays – **Beispielaufgabe** – **Struktogramm**

2-dimensionale Arrays – **Beispielaufgabe** – **Struktogramm**

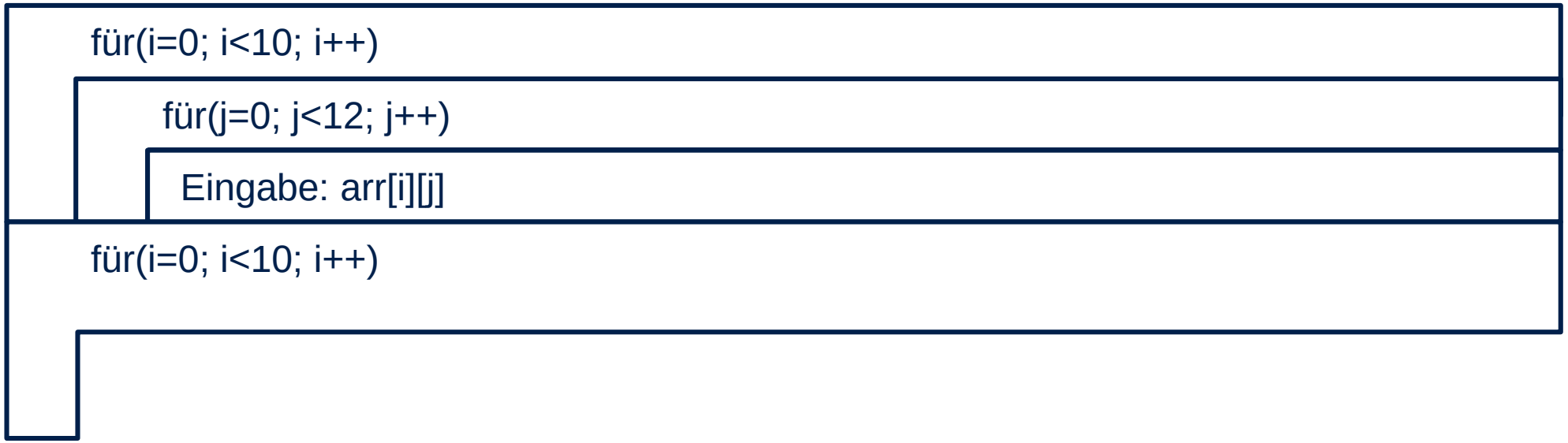
2-dimensionale Arrays – **Beispielaufgabe** – **Struktogramm**



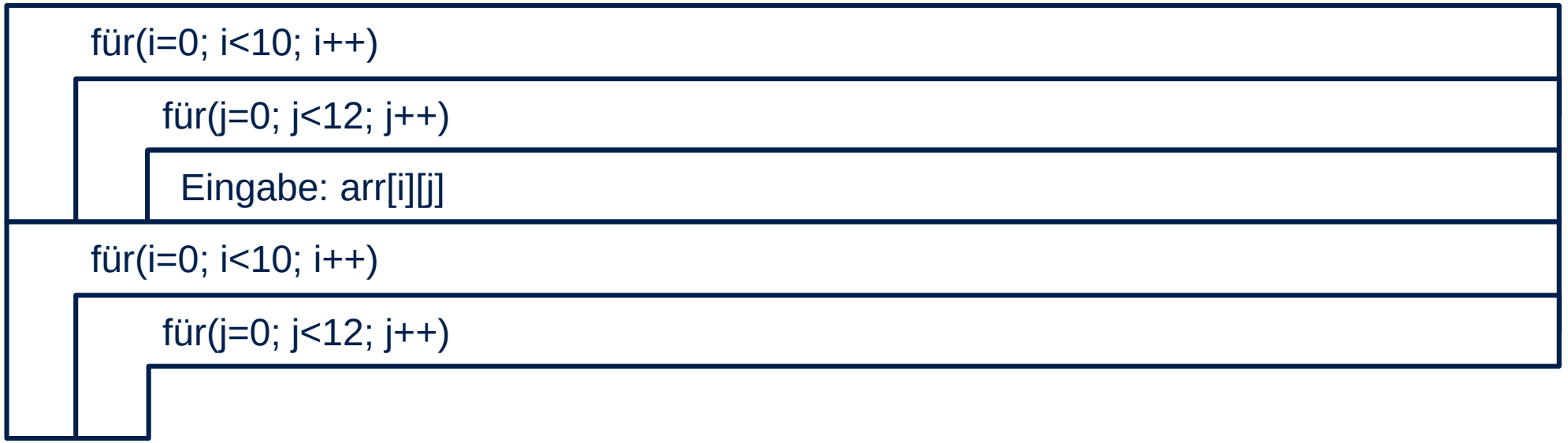
2-dimensionale Arrays – **Beispielaufgabe** – **Struktogramm**



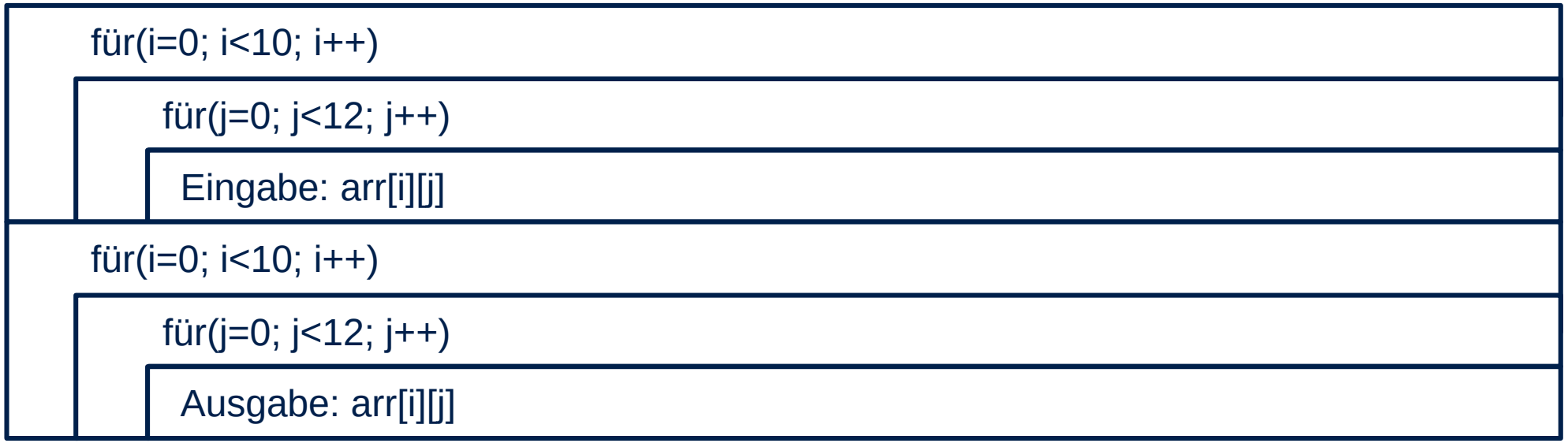
2-dimensionale Arrays – **Beispielaufgabe** – **Struktogramm**



2-dimensionale Arrays – **Beispielaufgabe** – **Struktogramm**



2-dimensionale Arrays – **Beispielaufgabe** – **Struktogramm**



2-dimensionale Arrays – **Beispielaufgabe** – **Pseudocode**

```
Programm „2-dimensionales-Array-Beispiel“
{
        für(i=0;i<10;i++)
        {
                für(j=0;j<12;j++)
                {
                        Eingabe: arr[i][j]
                }
        }

        für(i=0;i<10;i++)
        {
                für(j=0;j<12;j++)
                {
                        Ausgabe: arr[i][j]
                }
        }
}
```

2-dimensionale Arrays – **Beispielaufgabe** – **Quellcode**

```
#include <stdio.h>
#include <windows.h>

int main(void)
{
   system("chcp.com 1252");
   system("cls");

   int i,j;
   float arr[10][12];

   for(i=0;i<10;i++)
   {
      for(j=0;j<12;j++)
      {
         printf("Geben Sie bitte den Umsatz des %d. Monats der %d. Filiale: ",j+1,i+1);
         fflush(stdin);
         scanf("%f",&arr[i][j]);
      }
   }

   for(i=0;i<10;i++)
   {
      for(j=0;j<12;j++)
      {
         printf("In arr[%d][%d] ist der Umsatz des %d. Monats der %d. Filiale abgespeichert, er beträgt: %.2f €\n",i,j,j+1,i+1,arr[i][j]);
      }
   }

   printf("\n\n\n\n");
   system("pause");
   return 0;
}
```

WBS TRAINING

String-Arrays – Definition und Motivation

- Auch bei 2-dimensionalen Arrays kann der Typ natürlich frei gewählt werden. Entsprechend gibt es also auch 2-dimensionale Arrays von **Typ Character**.
- Es sei nun **arr[x][y]** ein 2-dimensionales Character-Array, dann besteht es aus x Array-Member vom Typ Character, die ihrerseits jeweils **y** Elemente besitzen.
- Falls nun in jedem der **x** Array-Member ein String abgespeichert wurde (mit maximal **y-1** Zeichen und **1** Terminator), so kann man **arr[x][y]** auch als ein (1-dimensionales) **String-Array** betrachten.
- Wie schon bei Arrays im Allgemeinen, so besteht die **Motivation** für den Einsatz von String-Arrays in der Möglichkeit, alle Strings des Arrays mittels einer Schleife der Reihe nach abzuarbeiten.
- Auch bei String-Arrays existiert eine abkürzende Schreibweise für die **Start-Adresse** jedes Strings:

  arr[0] ist die Abkürzung für &arr[0][0], also die **Adresse des 1. Elements des 1. Strings**
  arr[1] ist die Abkürzung für &arr[1][0], also die **Adresse des 1. Elements des 2. Strings**
  arr[2] ist die Abkürzung für &arr[2][0], also die **Adresse des 1. Elements des 3. Strings**
  ...
  arr[x-1] ist die Abkürzung für &arr[x-1][0], also die **Adresse des 1. Elements des x-ten Strings**

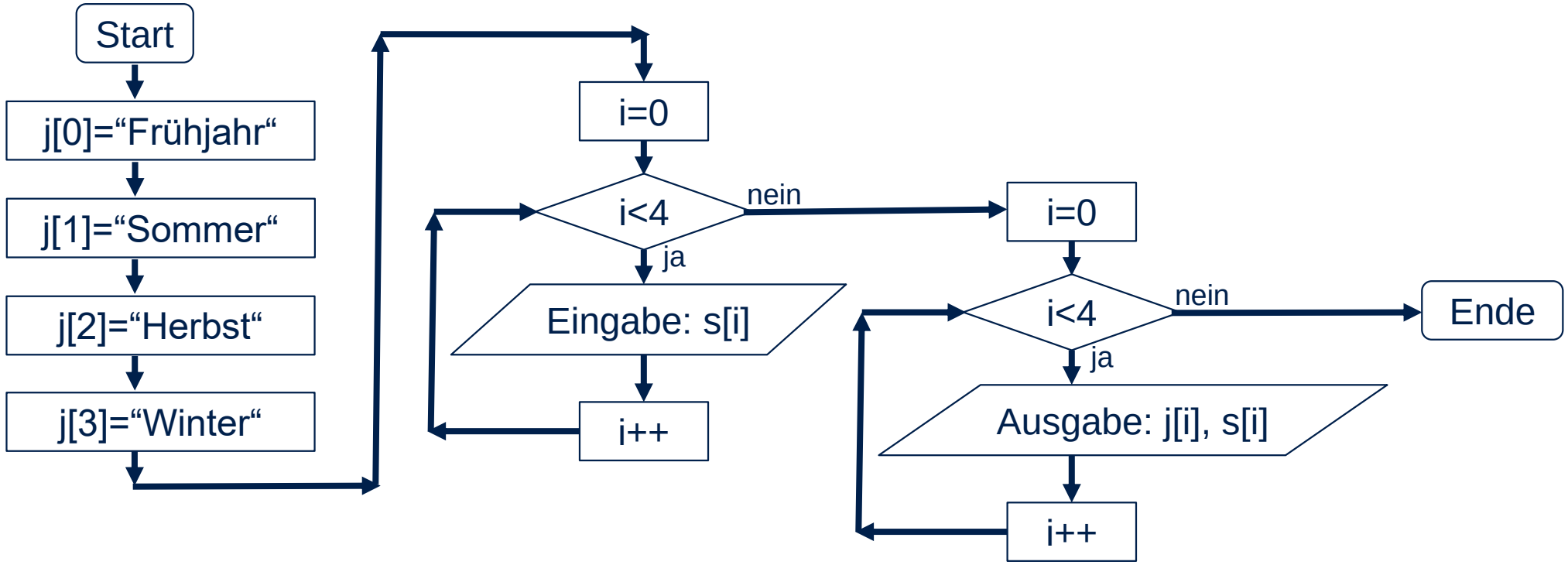
String-Arrays – **Beispielaufgabe**

Aufgabenstellung

- Zu Beginn des Programmes wird das Character-Array **jahreszeiten[4][10]** mit den Strings „Frühjahr“, „Sommer“, „Herbst“ und „Winter“ initialisiert.
- Es startet eine Eingabeschleife, die 4-mal durchlaufen wird, vom User die englischen Jahreszeit-Bezeichnungen abfragt und diese in **season[4][10]** abspeichert.
- Anschließend werden in einer Schleife alle Jahreszeiten durchlaufen und pro deutscher Bezeichnung die jeweilige englische ausgegeben. Danach endet das Programm.

String-Arrays – **Beispielaufgabe** – PAP

(aus Platzgründen kürzen wir jahrezeit[][] mit j[][] und season[][] mit s[][] ab)



String-Arrays – **Beispielaufgabe** – **Struktogramm**

String-Arrays – **Beispielaufgabe** – **Struktogramm**

j[0]=“Frühjahr“

String-Arrays – **Beispielaufgabe** – **Struktogramm**

j[0]=“Frühjahr“
j[1]=“Sommer“

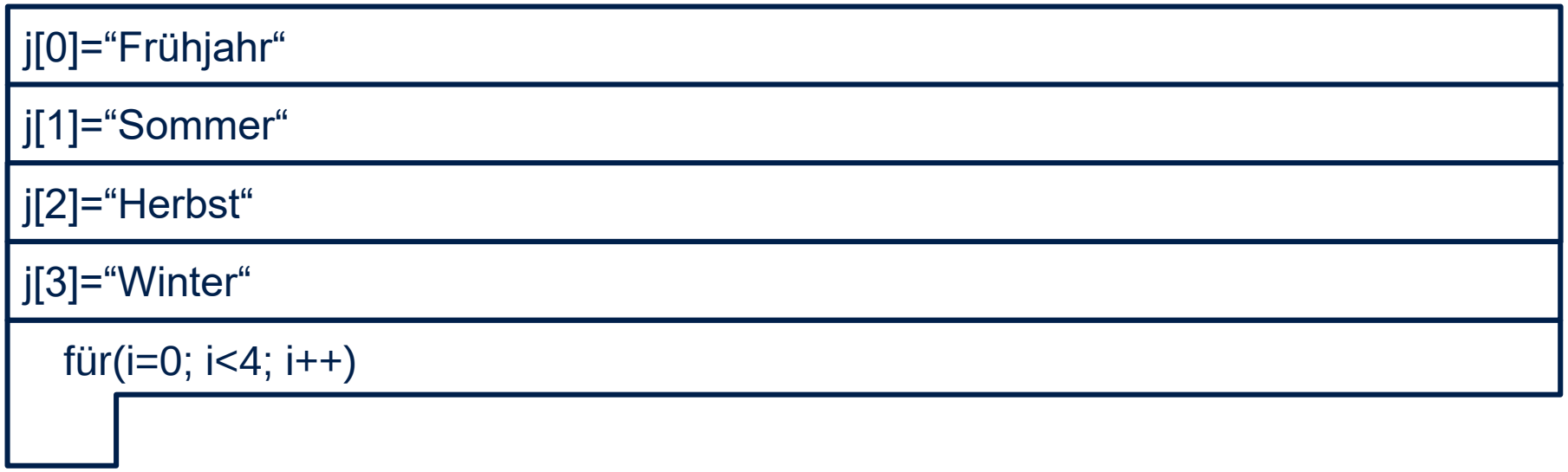
String-Arrays – **Beispielaufgabe** – **Struktogramm**

j[0]=“Frühjahr“
j[1]=“Sommer“
j[2]=“Herbst“

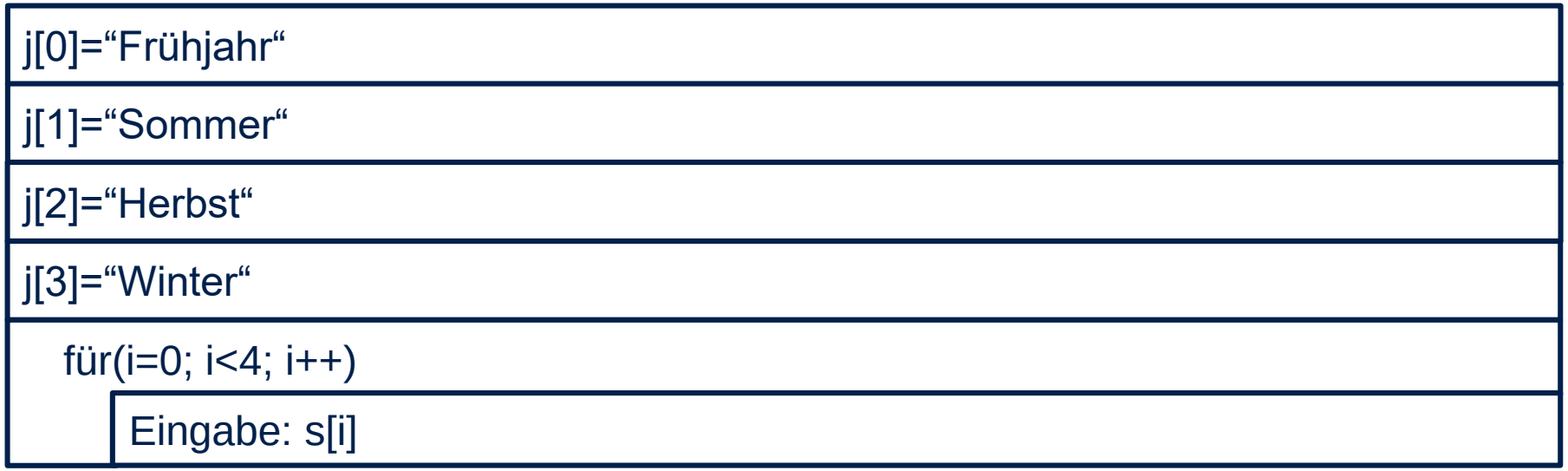
String-Arrays – **Beispielaufgabe** – **Struktogramm**

j[0]=“Frühjahr“
j[1]=“Sommer“
j[2]=“Herbst“
j[3]=“Winter“

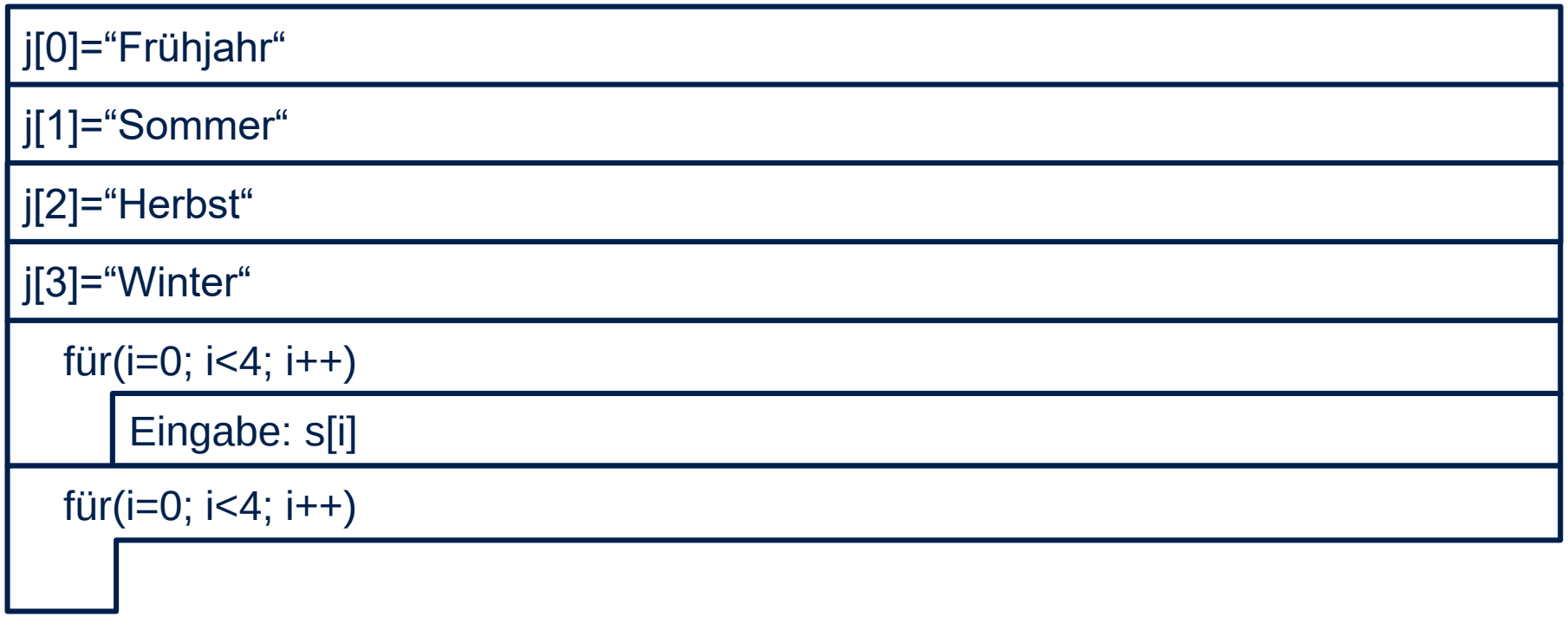
String-Arrays – **Beispielaufgabe** – **Struktogramm**



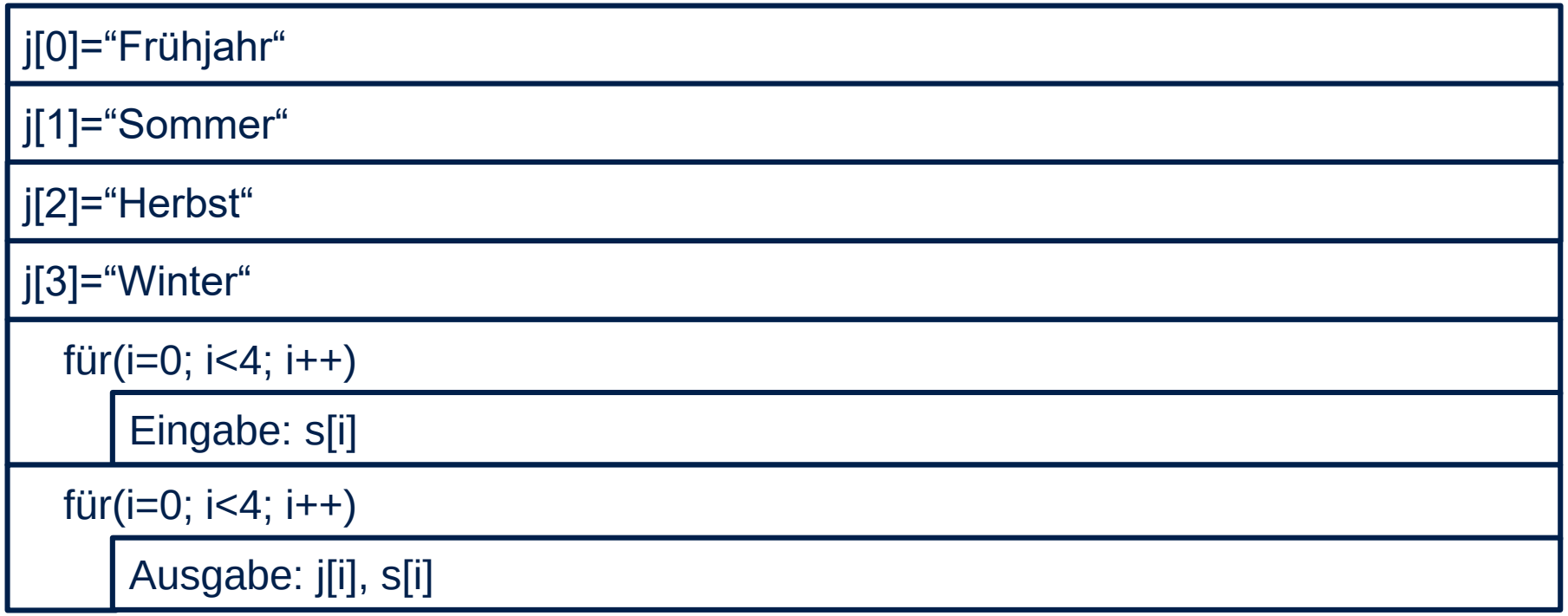
String-Arrays – **Beispielaufgabe** – **Struktogramm**



String-Arrays – **Beispielaufgabe** – **Struktogramm**



String-Arrays – **Beispielaufgabe** – **Struktogramm**



String-Arrays – **Beispielaufgabe** – **Pseudocode**

```
Programm „String-Array-Beispiel“
{
        j[0]=“Frühjahr“
        j[1]=“Sommer“
        j[2]=“Herbst“
        j[3]=“Winter“

        für(i=0;i<4;i++)
        {
                Eingabe: s[i]
        }

        für(i=0;i<4;i++)
        {
                Ausgabe: j[i], s[i]
        }
}
```

String-Arrays – **Beispielaufgabe** – **Quellcode**

```
#include <stdio.h>
#include <windows.h>
#include <string.h>

int main(void)
{
   system("chcp.com 1252");
   system("cls");

   int i;
   char jahreszeit[4][10],season[4][10];

   strcpy(jahreszeit[0],"Frühjahr");
   strcpy(jahreszeit[1],"Sommer");
   strcpy(jahreszeit[2],"Herbst");
   strcpy(jahreszeit[3],"Winter");

   for(i=0;i<4;i++)
   {
      printf("Geben Sie bitte die englische Bezeichnung der %d. Jahreszeit ein: ",i+1);
      fflush(stdin);
      scanf("%[^\n]",season[i]);
   }

   for(i=0;i<4;i++)
   {
      printf("Die Jahreszeit \"%s\" wird im englischen \"%s\" genannt\n",jahreszeit[i],season[i]);
   }

   printf("\n\n\n\n");
   system("pause");
   return 0;
}
```

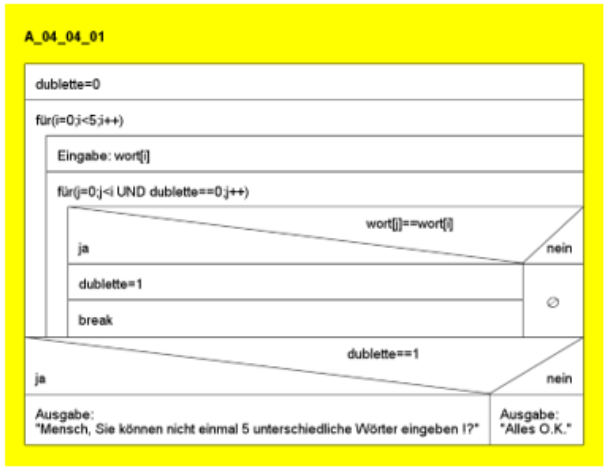
WBS TRAINING

2-dimensionale Arrays – Gemeinsame Übung A_04_04_01

WBS TRAINING

Aufgabe_04_04_01

Gegeben sei das folgende Struktogramm:



Erläuterung:
Bei der obigen Aufgabe sollen (auf jeden Fall) 5 Wörter abgefragt (und abgespeichert) werden. Ferner soll untersucht werden, ob es **mindestens** eine Dublette gibt. Entsprechend müssen keine weiteren Dubletten gesucht werden, falls zuvor bereits eine Dublette gefunden worden war.

Aufgabenstellung:
Bitte erstellen Sie dazu einen geeigneten **Quellcode** in ANSI C.

WBS TRAINING

VIELEN DANK FÜR IHRE AUFMERKSAMKEIT!