

# Programmierung(2)

# Agenda

- Verschachtelte Schleifen
  - **Definition, Motivation** und Beispiel
  - Darstellung im **PAP**
  - Darstellung im **Struktogramm**
  - Darstellung im **Pseudocode**
  - Syntax in **ANSI C**
- Sprungbefehl **break**
  - **Definition, Motivation** und Beispiel
  - Darstellung im **Struktogramm**
  - Darstellung im **Pseudocode**
  - Syntax in **ANSI C**
- Sprungbefehl **continue**
  - **Definition, Motivation** und Beispiel
  - Darstellung im **Struktogramm**
  - Darstellung im **Pseudocode**
  - Syntax in **ANSI C**
- Fachpraktische Anwendungen

# Verschachtelte Schleifen – Definition und Motivation

- Wenn innerhalb einer Schleife pro Durchlauf eine weitere Schleife gestartet wird, so spricht man von einer „**Verschachtelten Schleife**“.
- Die Schleife, die pro Durchlauf gestartet wird, wird entsprechend als „**Innere Schleife**“ bezeichnet.
- Die sie umgebende Schleife heißt dann naheliegender Weise „**Äußere Schleife**“.
- Die Motivation für (einfache) Schleifen haben wir bereits in P1 kennen gelernt:
  - Oft geht es nur darum, einen mehrfach auszuführenden Codeabschnitt nicht mehrfach codieren zu müssen.
  - Ferner kann zum Zeitpunkt der Programmerstellung unbekannt sein, wie oft ein Codeabschnitt auszuführen sein wird, weil dies erst während der Laufzeit des Programmes entschieden werden kann. In diesen Fällen sind Schleifen dann nicht nur eine Arbeitserleichterung, sondern in der Regel sogar unverzichtbar.
- Mit genau diesen Motivationen werden auch Innere und Äußere Schleifen verwendet, denn natürlich kann es innerhalb des Schleifenrumpfes einer (Äußeren) Schleife selbst wieder einen Codeabschnitt geben, der pro Durchlauf der (Äußeren) Schleife mehrfach hintereinander ausgeführt werden muss, so dass dann entsprechend innerhalb der (Äußeren) Schleife eine weitere (Innere) Schleife eingesetzt wird.

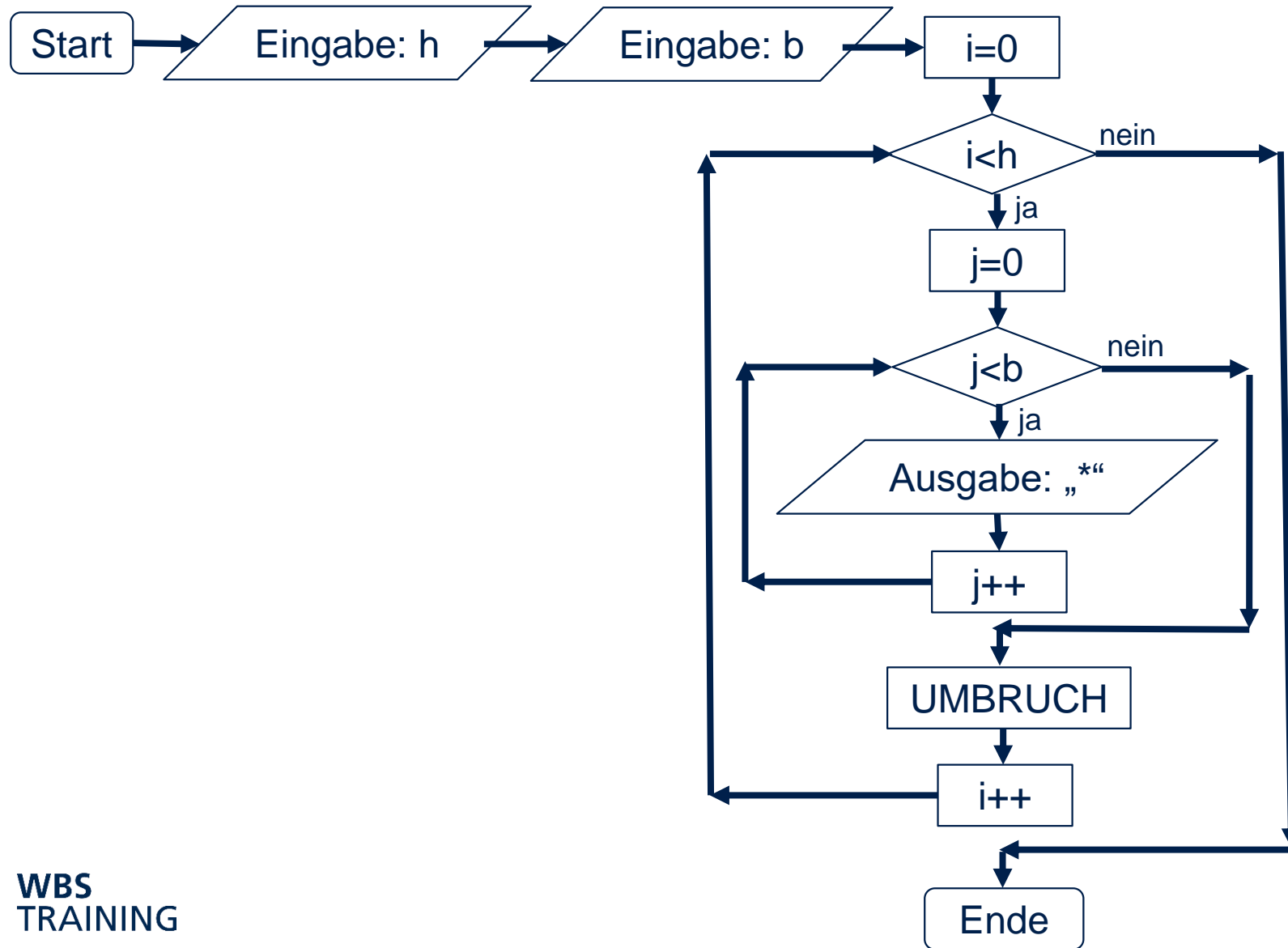
# Verschachtelte Schleifen – **Beispielaufgabe**

## Aufgabenstellung:

- Das Programm startet mit der Abfrage von **Höhe** und **Breite** eines zu zeichnenden Rechteckes.
- Anschließend startet eine Äußere Schleife, die so oft durchlaufen wird, wie das Rechteck hoch ist, pro Durchlauf:
  - startet eine Innere Schleife, die so oft durchlaufen wird, wie das Rechteck breit ist, pro Durchlauf:  
Wird ein Stern (‘\*’) auf der Konsole gesetzt (nebeneinander)
  - Nach der Inneren Schleife wird ein Umbruch gesetzt.
- Nach der Äußeren Schleife endet das Programm.

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

# Verschachtelte Schleifen – Beispielaufgabe – PAP



# Verschachtelte Schleifen – **Beispielaufgabe** – **Struktogramm**

# Verschachtelte Schleifen – **Beispielaufgabe** – **Struktogramm**

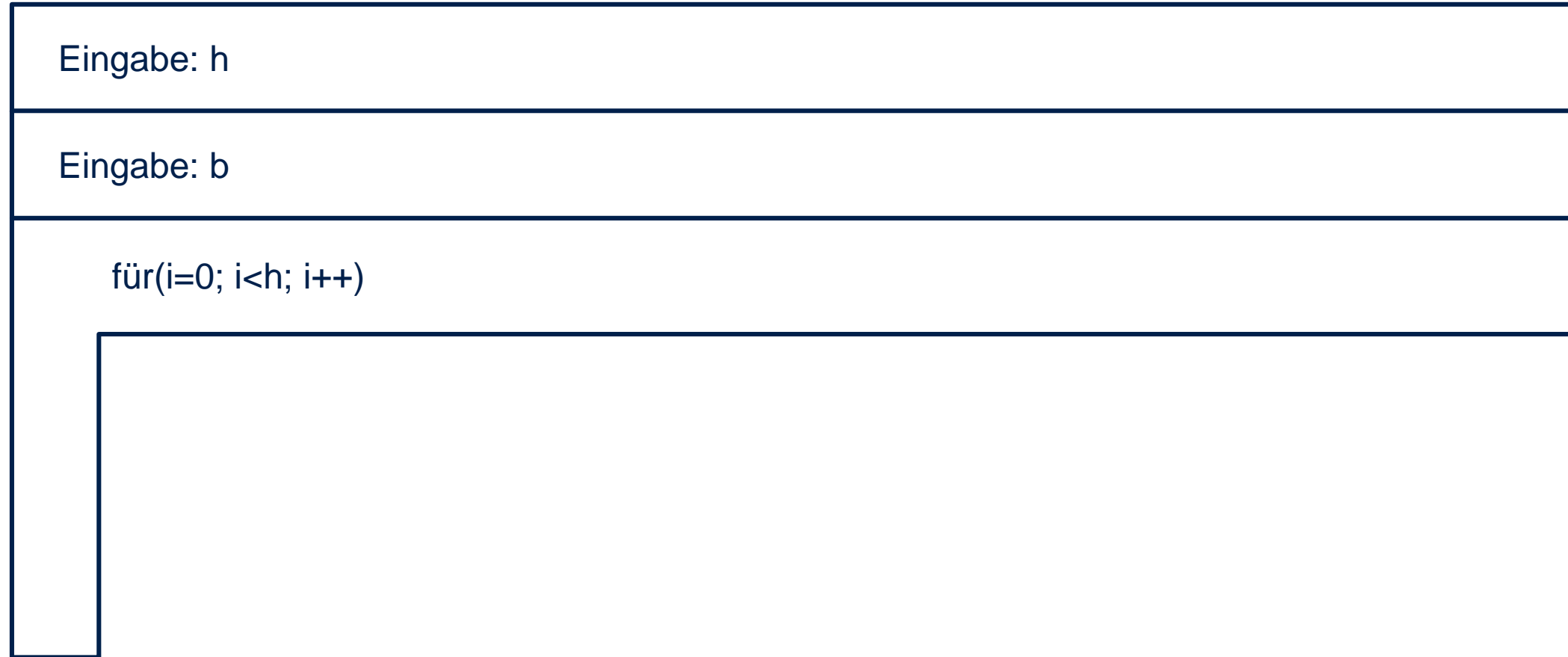
Eingabe: h

# Verschachtelte Schleifen – **Beispielaufgabe** – **Struktogramm**

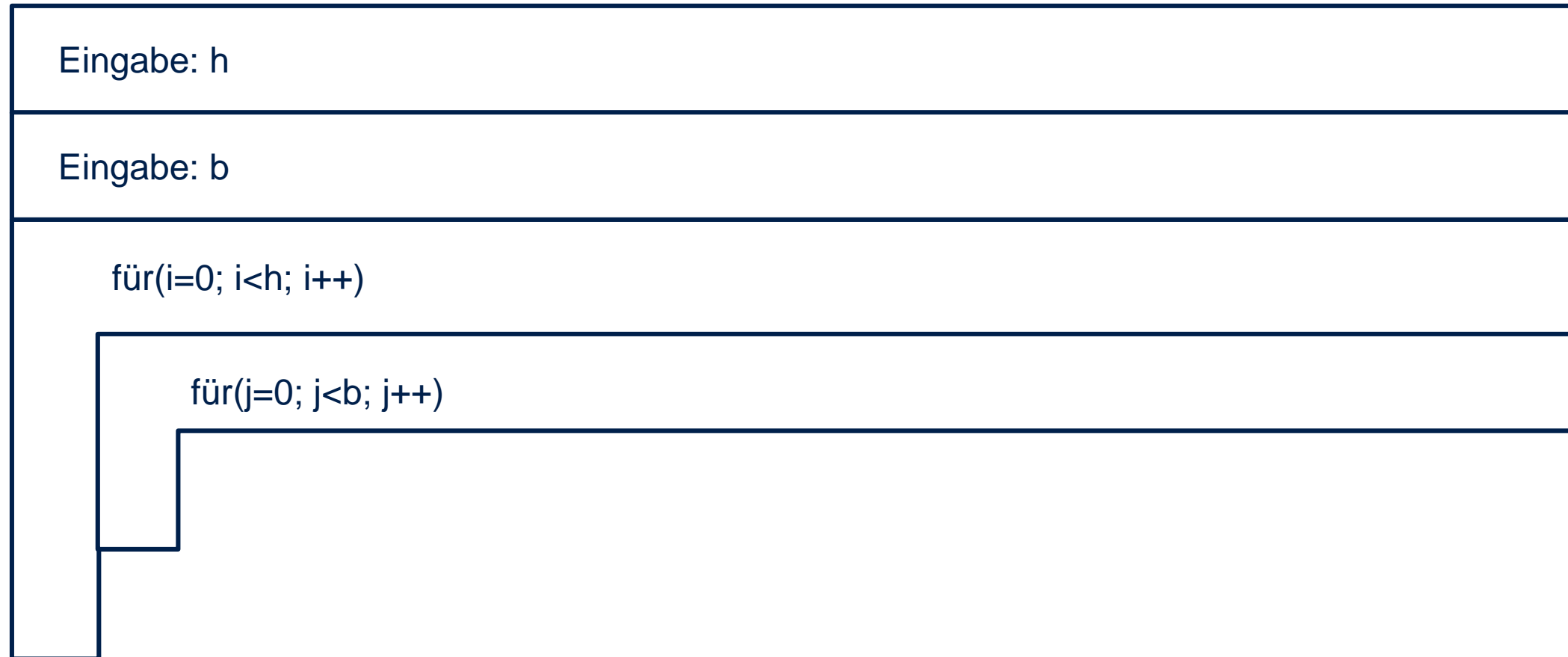
Eingabe: h
Eingabe: b



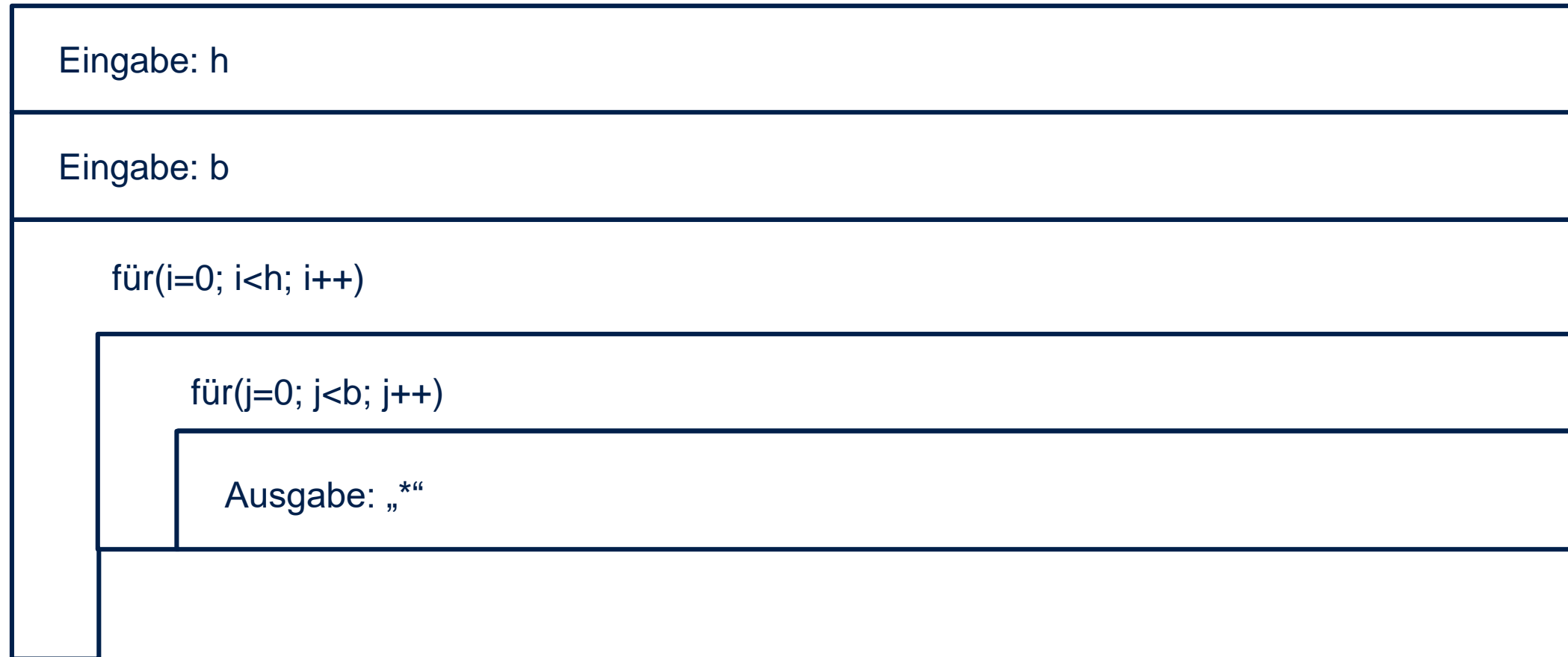
# Verschachtelte Schleifen – Beispielaufgabe – Struktogramm



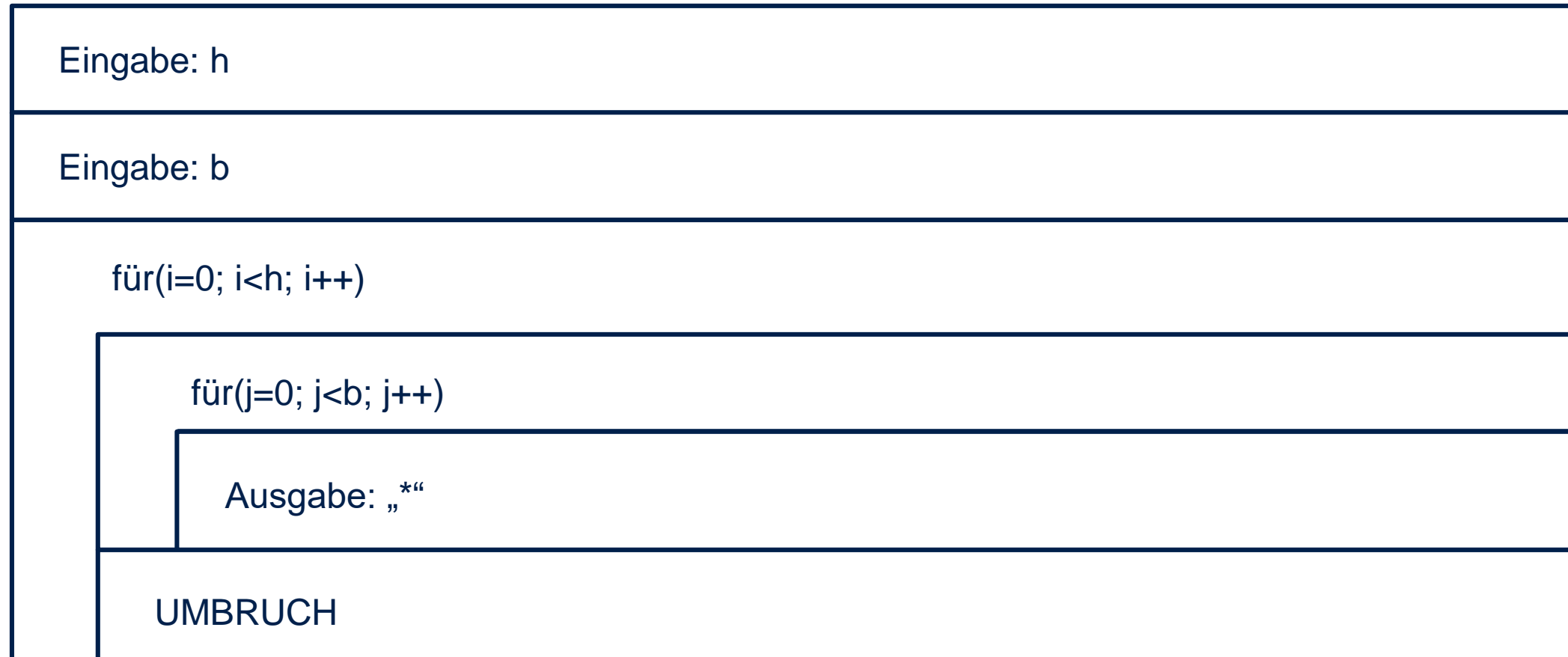
# Verschachtelte Schleifen – Beispielaufgabe – Struktogramm



# Verschachtelte Schleifen – Beispielaufgabe – Struktogramm



# Verschachtelte Schleifen – Beispielaufgabe – Struktogramm



# Verschachtelte Schleife – **Beispielaufgabe** – Pseudocode

**Programm** „Verschachtelte-Schleifen-Beispiel“

```
{  
    Eingabe: h  
    Eingabe: b  
    für(i=0;i<h;i++)  
    {  
        für(j=0;j<b;j++)  
        {  
            Ausgabe: „*“  
        }  
        UMBRUCH  
    }  
}
```

# Verschachtelte Schleife – Beispielaufgabe – Quellcode

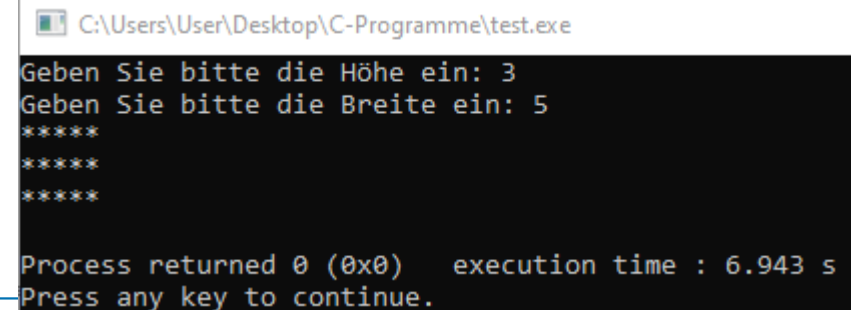
```
#include<stdio.h>

main()
{
    int h,b,i,j;

    printf("Geben Sie bitte die Höhe ein: ");
    scanf("%d",&h);

    printf("Geben Sie bitte die Breite ein: ");
    fflush(stdin);
    scanf("%d",&b);

    for(i=0;i<h;i++)
    {
        for(j=0;j<b;j++)
        {
            printf("*");
        }
        printf("\n");
    }
}
```



```
C:\Users\User\Desktop\C-Programme\test.exe
Geben Sie bitte die Höhe ein: 3
Geben Sie bitte die Breite ein: 5
*****
*****
*****

Process returned 0 (0x0)   execution time : 6.943 s
Press any key to continue.
```

# Sprunganweisung „break“ – Definition und Motivation

- Es kann gelegentlich bequem sein, ...
  - a) Schleifen „mitten aus ihrem Rumpf heraus“ zu verlassen (und nicht über deren Bedingung im Kopf oder Fuß)
  - b) den aktuellen Schleifendurchlauf abubrechen (und gegebenenfalls mit dem nächsten Durchlauf fortzufahren)Beides gelingt mit sogenannten „**Sprunganweisungen**“.
- Solche Sprunganweisungen könnten zwar immer auch „trickreich“ umgangen werden, so dass die Schleife letztlich doch erst über ihre Bedingung verlassen wird, aber eine solche Vorgehensweise ist in der Regel unelegant oder gar weniger performant.
- Es gibt mehrere Sprunganweisungen, von denen wir aber nur 2 näher kennenlernen wollen. Ihre Anwendung und Besonderheiten werden wir uns im Folgenden an Beispielen verdeutlichen.
- Der Vollständigkeit halber sei aber auch die Sprunganweisung „GOTO“ angesprochen, von der zwar jeder Programmierer gehört haben sollte, die aber möglichst ungenutzt bleibt, da sie als schlechter Stil gilt. Folgerichtig werden wir im Rest dieses Bausteines ihren Namen nicht einmal mehr aussprechen 😊
- Als erstes wollen wir uns mit der Sprunganweisung „**break**“ beschäftigen. Mit dieser verlässt man die Schleife und springt zum ersten Befehl hinter der Schleife.

# Sprunganweisung „break“ – **Beispielaufgabe**

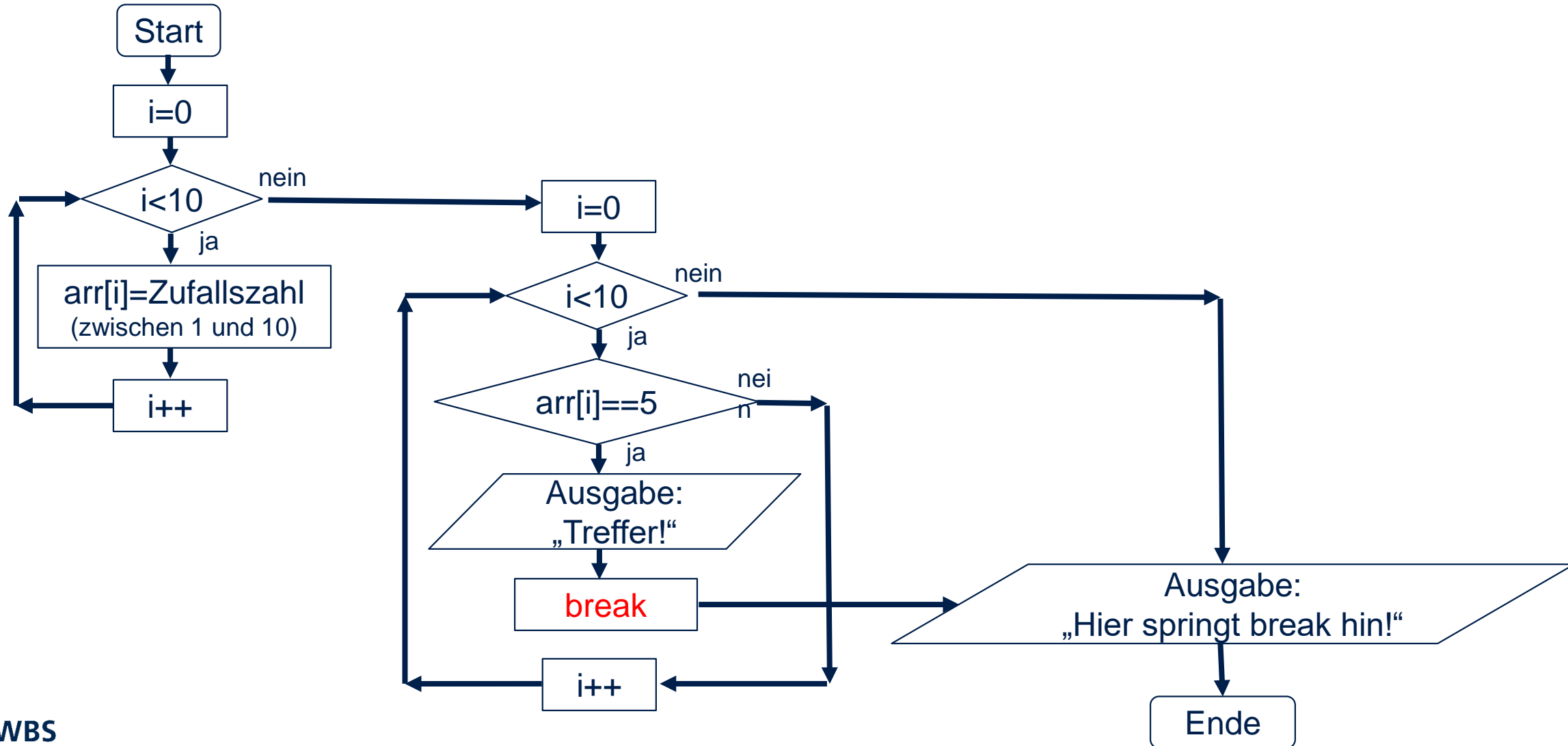
## Aufgabenstellung:

- Das Programm füllt zunächst ein 10 Felder umfassendes Integer Array mit Zufallszahlen zwischen 1 und 10.
- Anschließend soll untersucht werden, ob in dem Array der Wert 5 vorkommt.
- Zu diesem Zweck startet eine Schleife, die (im Prinzip) alle 10 Felder des Arrays durchläuft.
- Falls jedoch eine 5 gefunden wird, so soll „Treffer!“ auf der Konsole erscheinen und die Schleife sofort abbrechen.
- Nach der Schleife erscheint die Ausgabe: „Hier springt break hin!“ und das Programm endet

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



# Sprunganweisung „break“ – Beispielaufgabe – PAP



# Sprunganweisung „break“ – **Beispielaufgabe** – **Struktogramm**

# Sprunganweisung „break“ – Beispielaufgabe – Struktogramm

```
für(i=0; i<10; i++)
```

# Sprunganweisung „break“ – Beispielaufgabe – Struktogramm

```
für(i=0; i<10; i++)
```

```
arr[i]=Zufallszahl (zwischen 1 und 10)
```

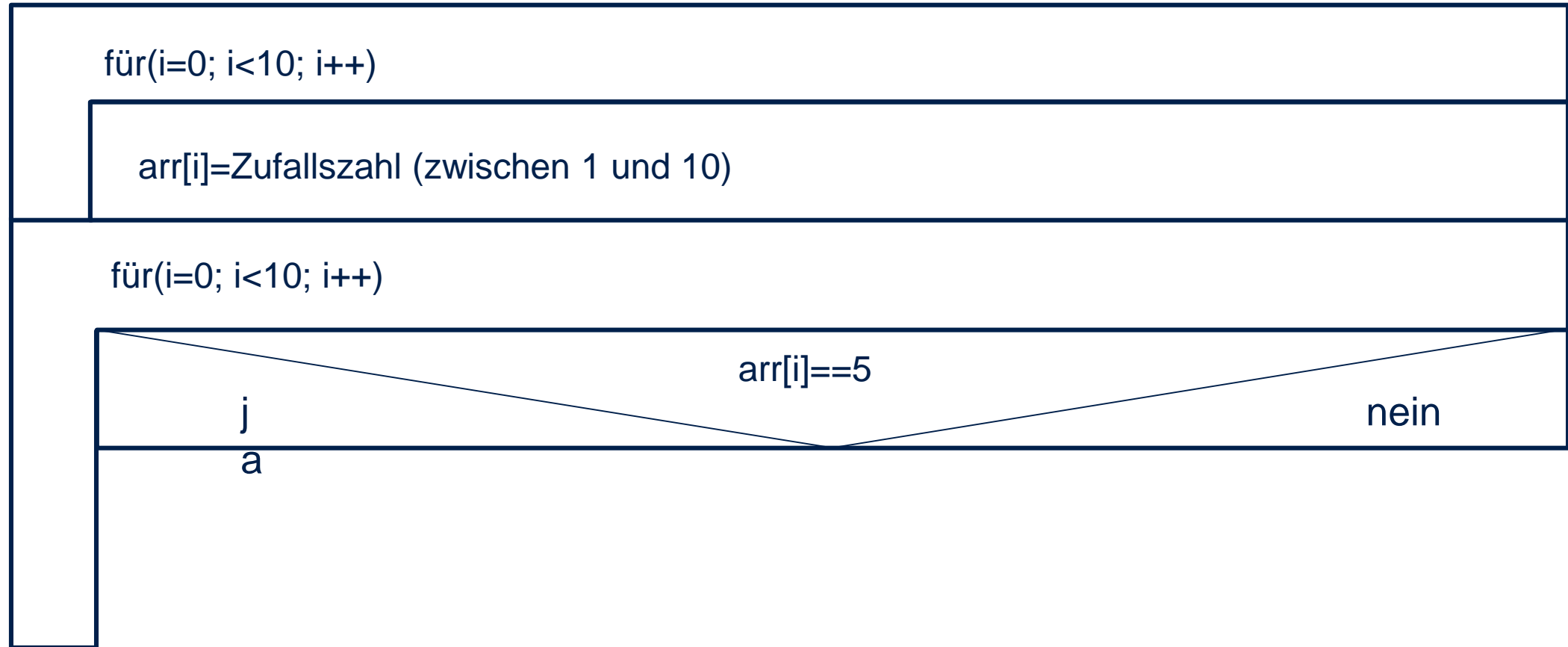
# Sprunganweisung „break“ – Beispielaufgabe – Struktogramm

für(i=0; i<10; i++)

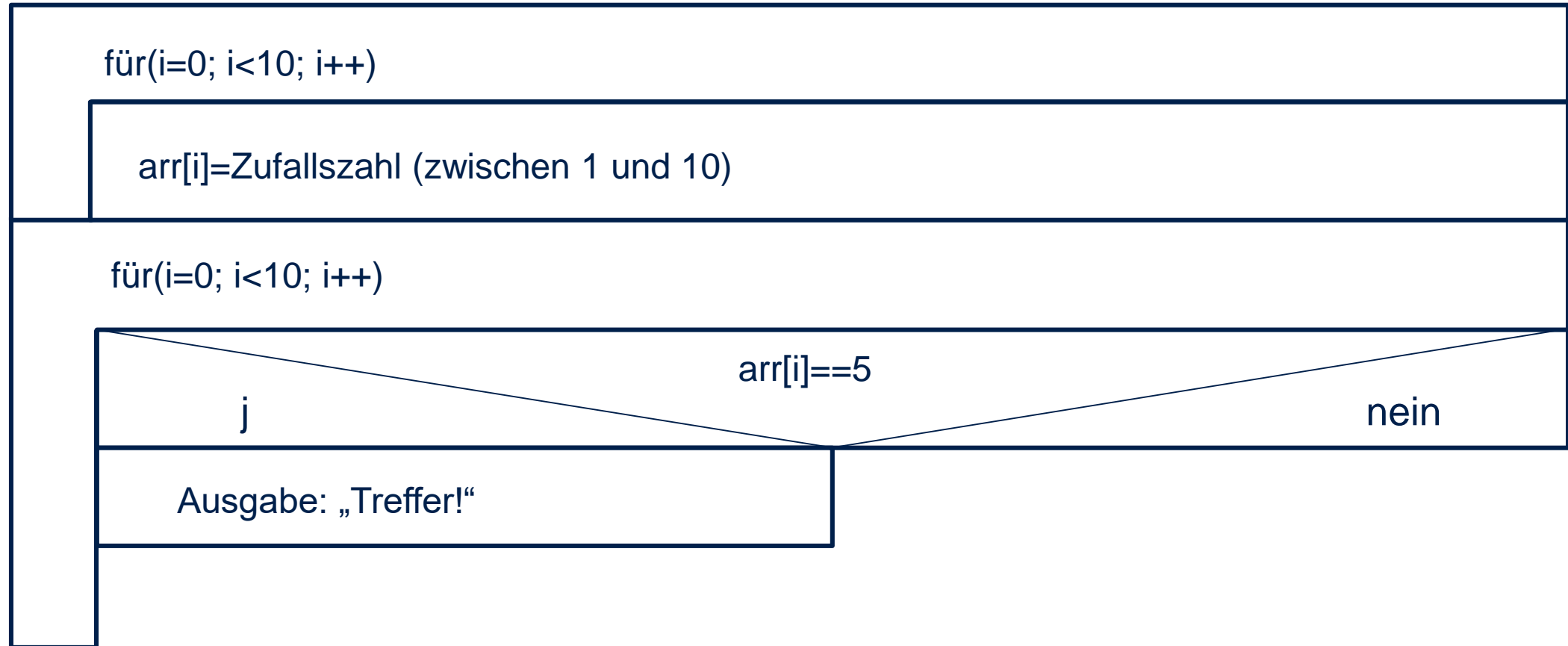
```
arr[i]=Zufallszahl (zwischen 1 und 10)
```

für(i=0; i<10; i++)

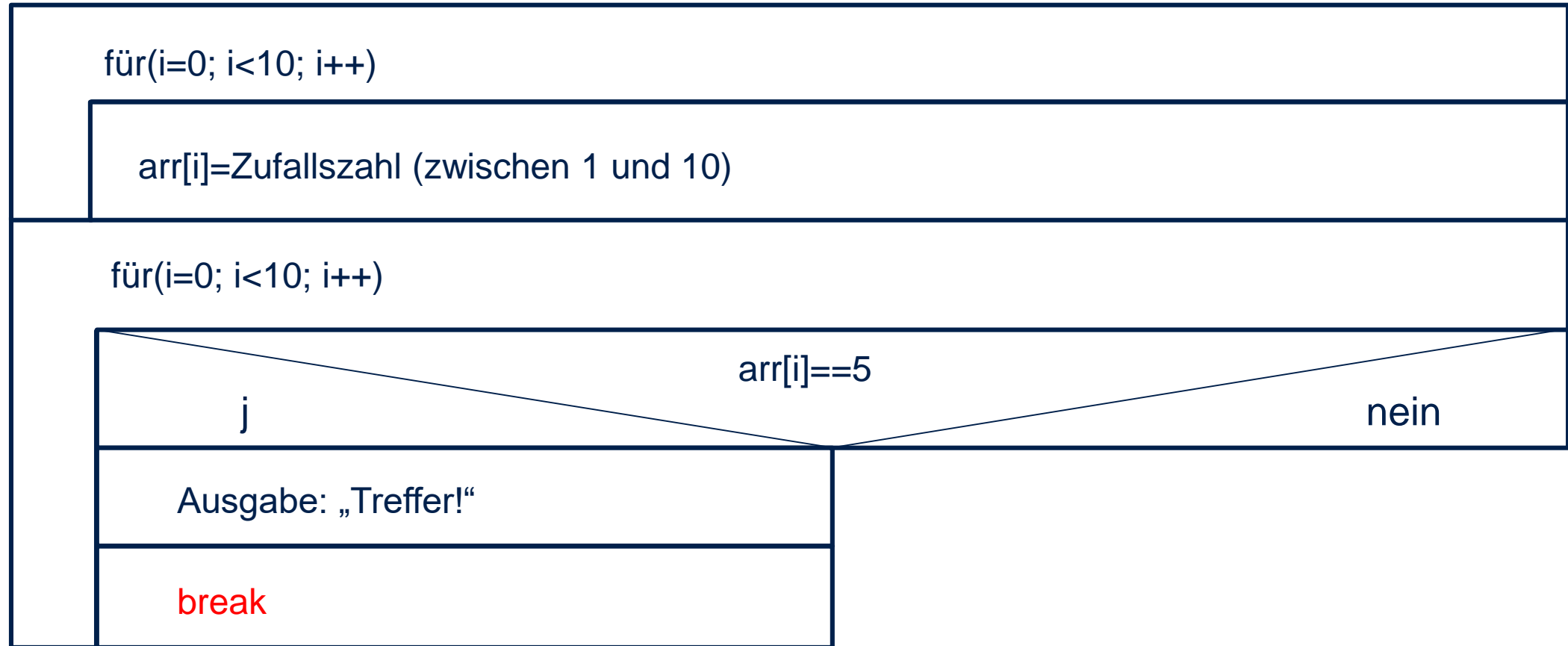
# Sprunganweisung „break“ – Beispielaufgabe – Struktogramm



# Sprunganweisung „break“ – Beispielaufgabe – Struktogramm

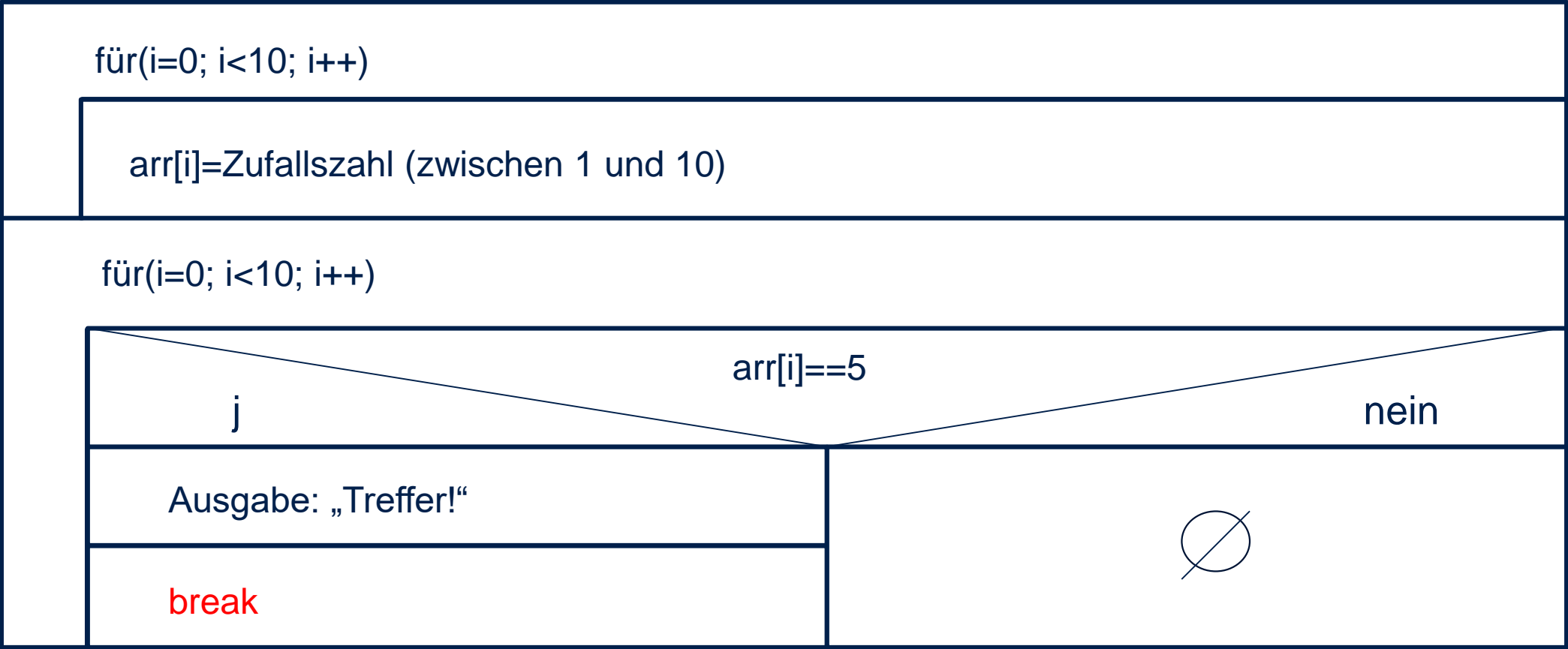


# Sprunganweisung „break“ – Beispielaufgabe – Struktogramm

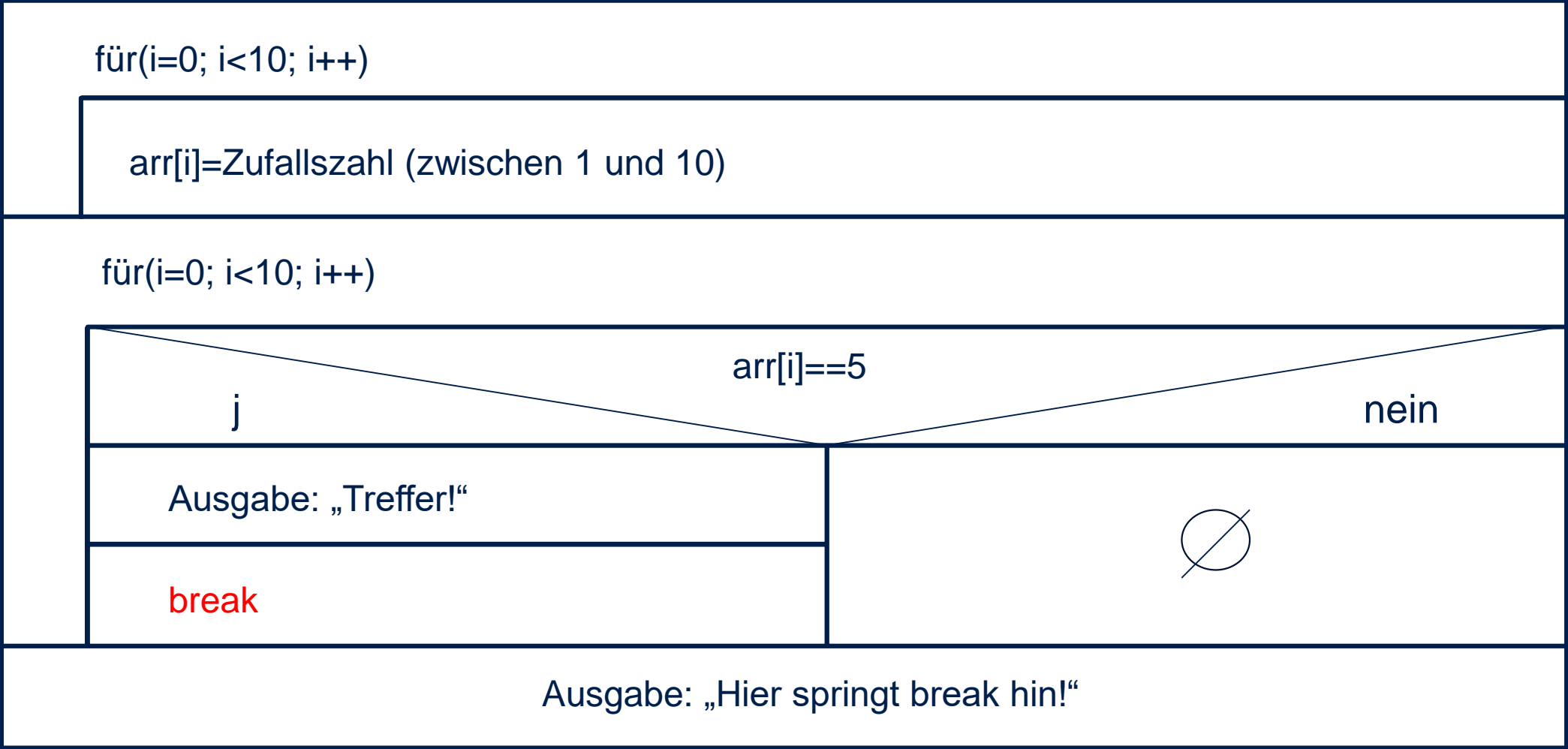




# Sprunganweisung „break“ – Beispielaufgabe – Struktogramm



# Sprunganweisung „break“ – Beispielaufgabe – Struktogramm



# Sprunganweisung „break“ – Beispielaufgabe – Pseudocode

## Programm „break-Beispiel“

```
{
    für(i=0;i<10;i++)
    {
        arr[i]=Zufallszahl (zwischen 1 und 10)
    }

    für(i=0;i<10;i++)
    {
        wenn(arr[i]==5)
        {
            Ausgabe: „Treffer!“
            break
        }
    }

    Ausgabe: „Hier springt break hin!“
}
```

# Sprunganweisung „break“ – Beispielaufgabe – Quellcode

```
#include<stdio.h>
#include<stdlib.h>
#include<time.h>

main()
{
    srand(time(NULL));
    int i,arr[10];

    for(i=0;i<10;i++)
    {
        arr[i]=rand()%10+1;
    }

    for(i=0;i<10;i++)
    {
        if(arr[i]==5)
        {
            printf("Treffer!");
            break;
        }
    }

    printf("Hier springt break hin!");
}
```

# Sprunganweisung „continue“ – Definition und Motivation

- Gelegentlich gibt es Gründe, einen aktuellen **Schleifendurchlauf abubrechen**, ohne die **gesamte Schleife zu beenden**, sondern den nächsten Schleifendurchlauf folgen zu lassen.  
=> Dies kann durch die Sprunganweisung „**continue**“ erreicht werden.
- Ob es tatsächlich einen weiteren Durchlauf geben wird, entscheidet natürlich die verwendete Schleifenbedingung, die unverändert vor jedem weiteren Durchlauf überprüft wird.
- Beim Einsatz von continue innerhalb von **for-Schleifen** ist ferner zu berücksichtigen, dass zwar kein einziger Befehl, der im Rumpf hinter dem continue folgt, noch ausgeführt wird, dass aber der Schleifenzähler vor der nächsten Bedingungsüberprüfung hochgezählt wird.
- Im Allgemeinen ist beim Einsatz von continue allerdings darauf zu achten, dass eine **Endlos-Schleife** entsteht, falls durch die Verwendung von continue jene Zeilen übersprungen werden, die für eine Änderung der Bedingungswerte von Bedeutung sind.

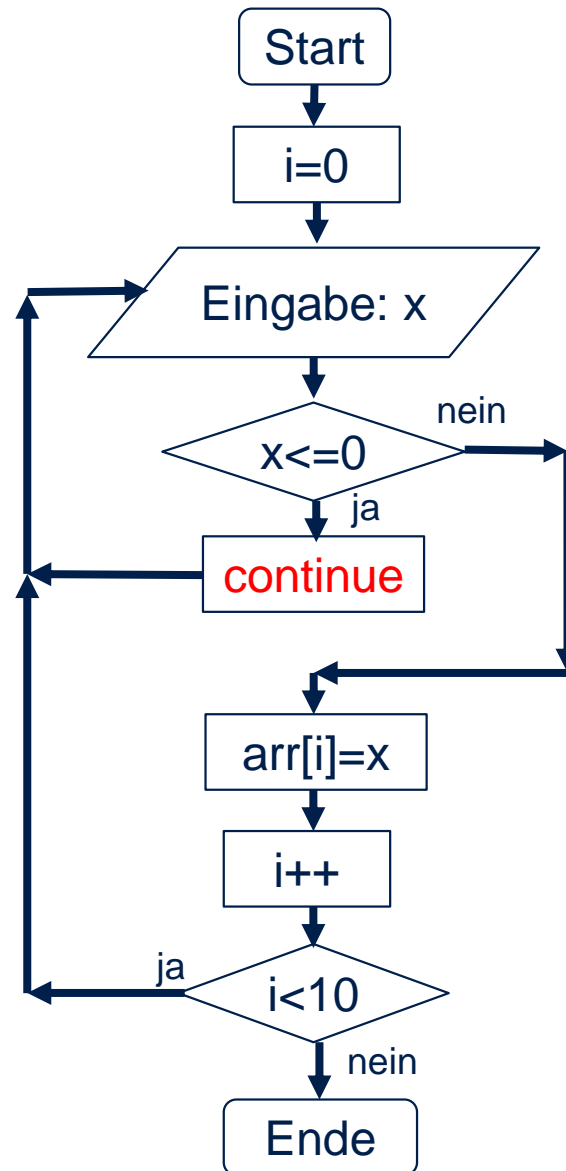
# Sprunganweisung „continue“ – **Beispielaufgabe**

## Aufgabenstellung:

- Das Programm startet mit einer fußgesteuerten Zähler-Schleife (Startwert 0), pro Durchlauf:
  - wird der User aufgefordert, eine ganze positive Zahl einzugeben.
  - falls die Eingabe jedoch kleiner/gleich 0 ist (und also nicht positiv), so wird der aktuelle Schleifendurchlauf abgebrochen
  - sonst wird ...
    - a) das Arrayfeld arr[Zählerwert] mit der aktuellen User-Eingabe gefüllt
    - b) der Zählerwert um 1 erhöht
  - Die Schleife endet, wenn alle 10 Felder von arr gefüllt sind.
- Nach der Schleife endet auch das Programm.

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

# Sprunganweisung „continue“ – Beispielaufgabe – PAP



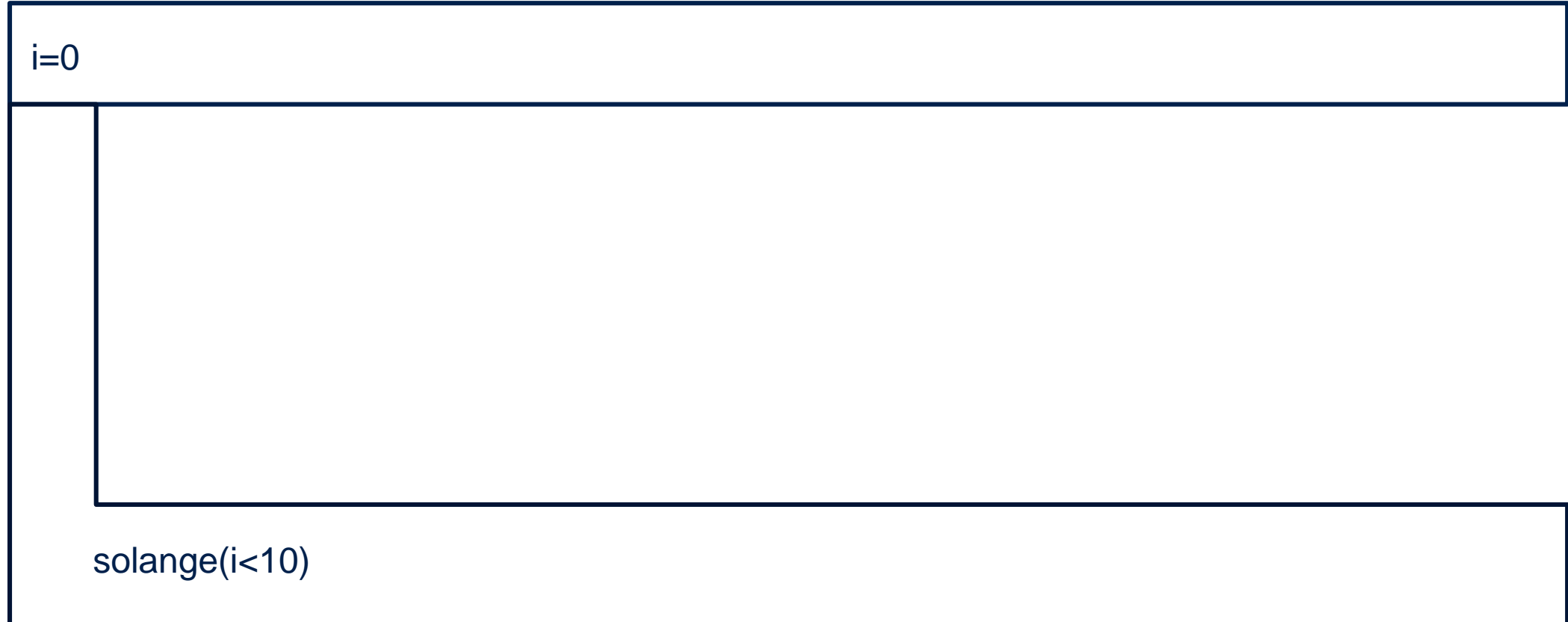
# Sprunganweisung „continue“ – Beispielaufgabe – **Struktogramm**



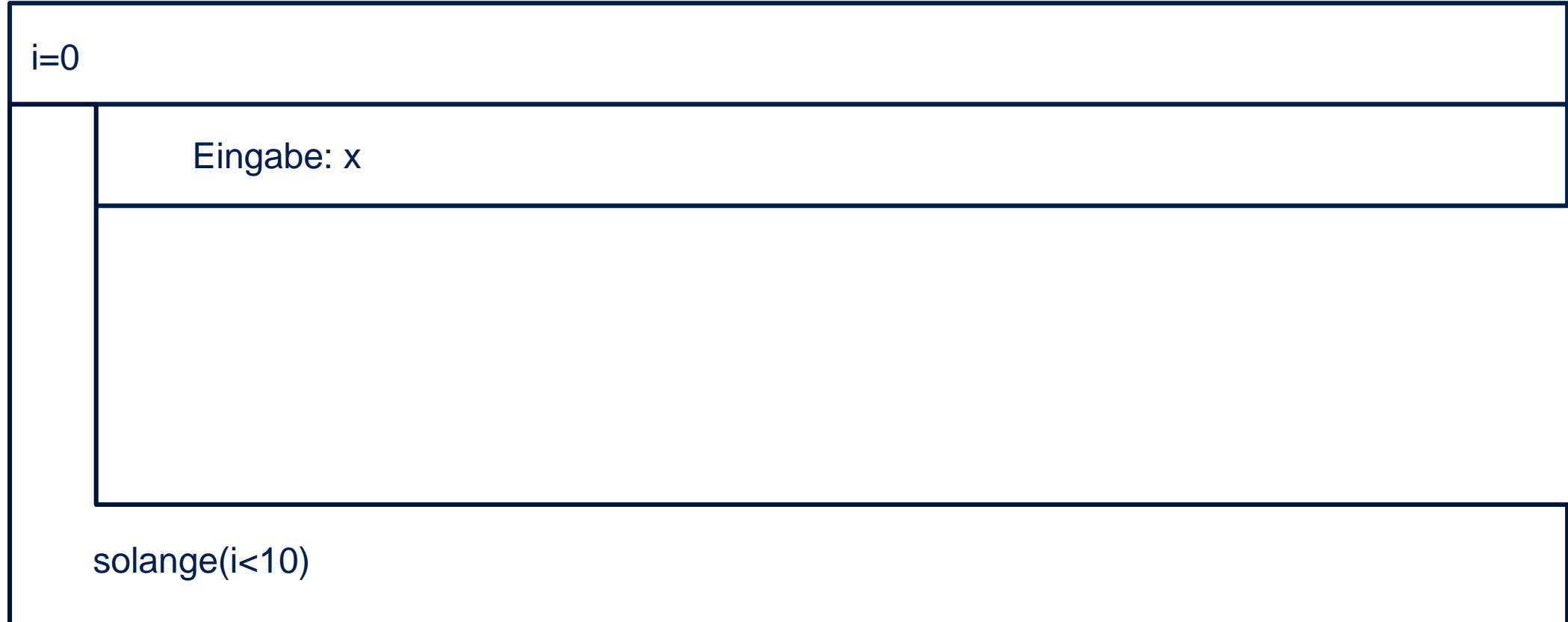
# Sprunganweisung „continue“ – Beispielaufgabe – **Struktogramm**

i=0

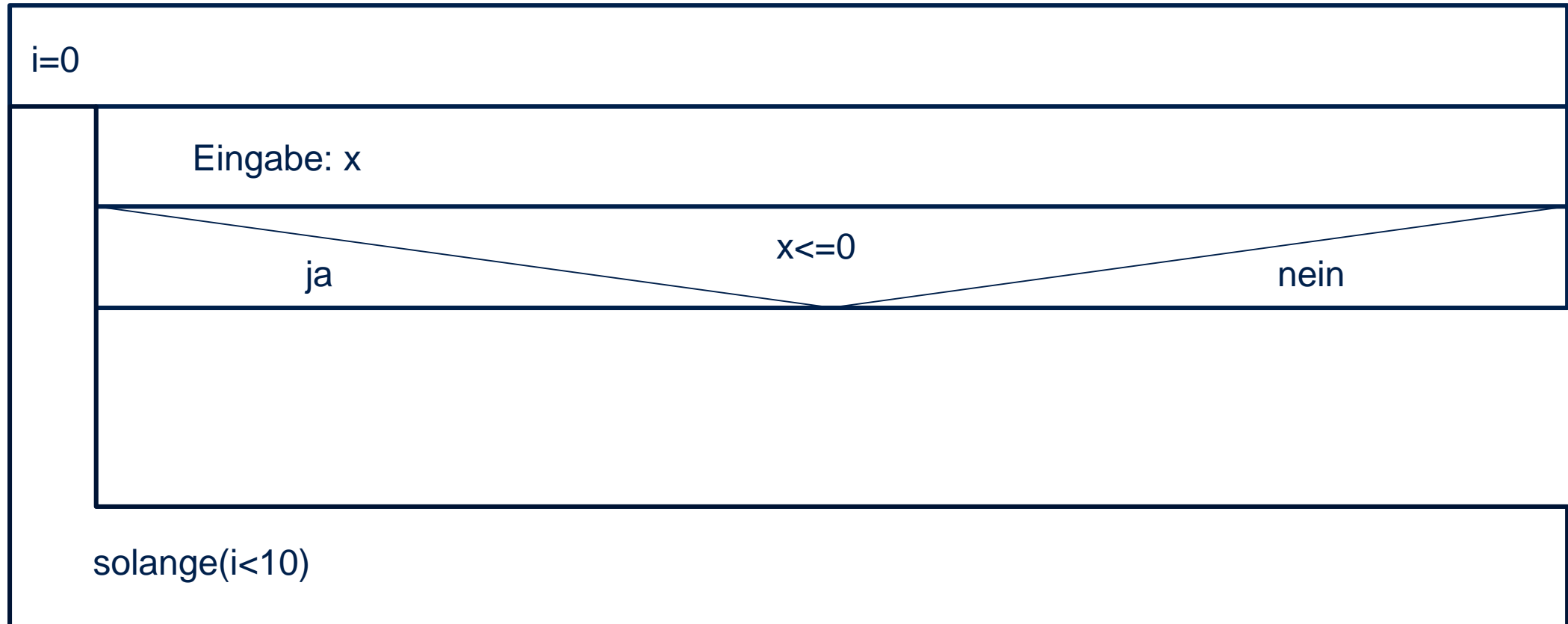
# Sprunganweisung „continue“ – Beispielaufgabe – Struktogramm



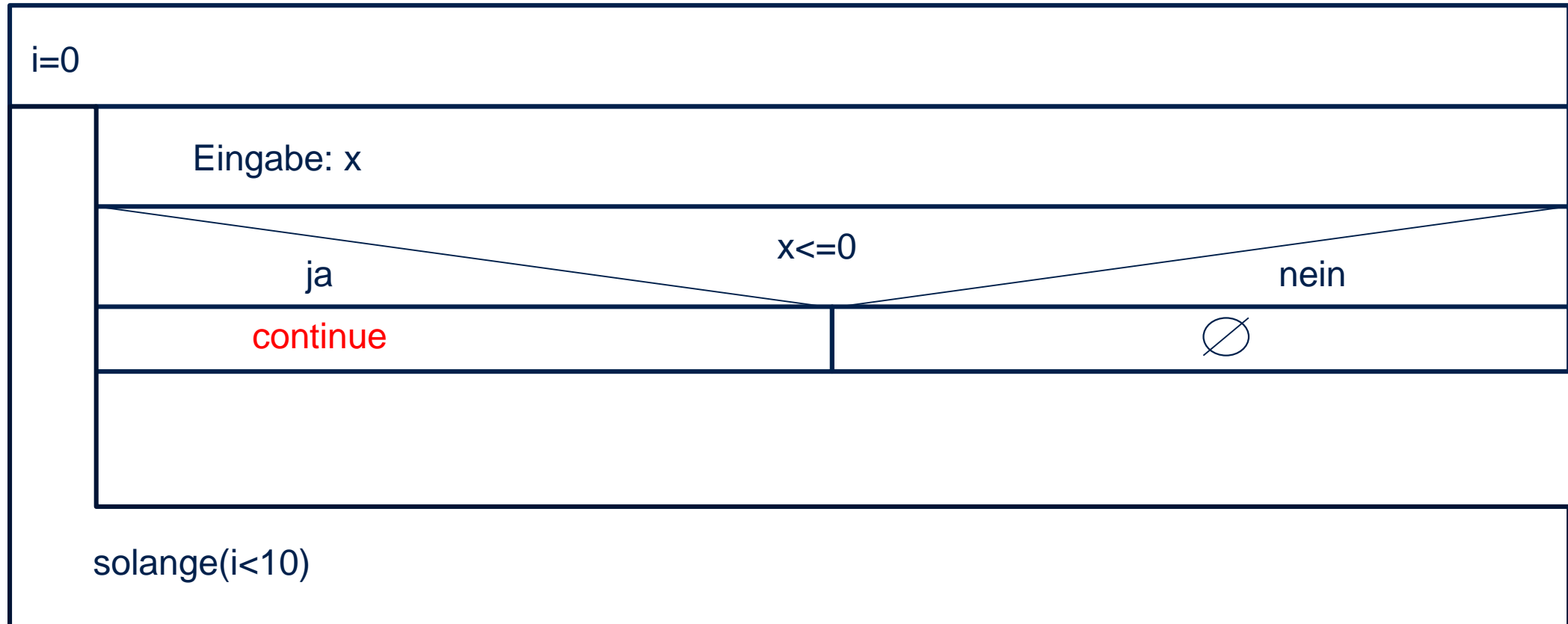
# Sprunganweisung „continue“ – Beispielaufgabe – Struktogramm



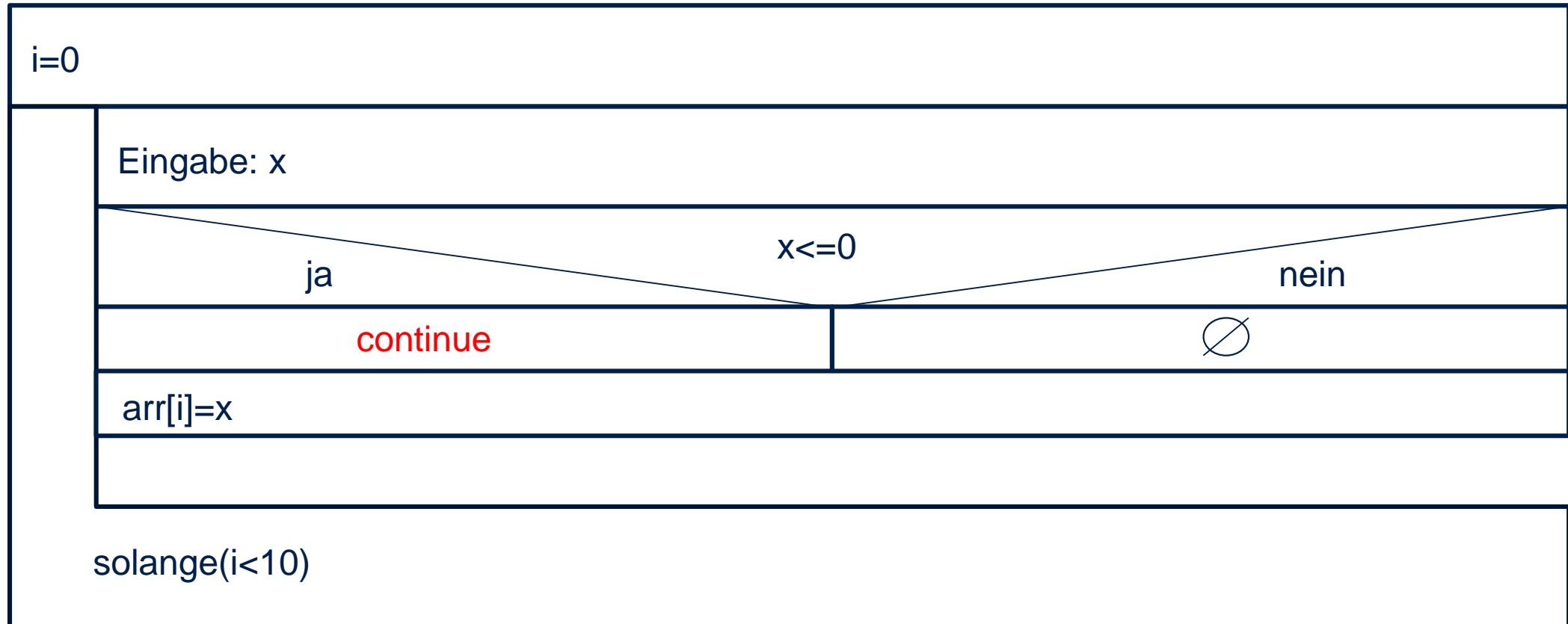
# Sprunganweisung „continue“ – Beispielaufgabe – Struktogramm



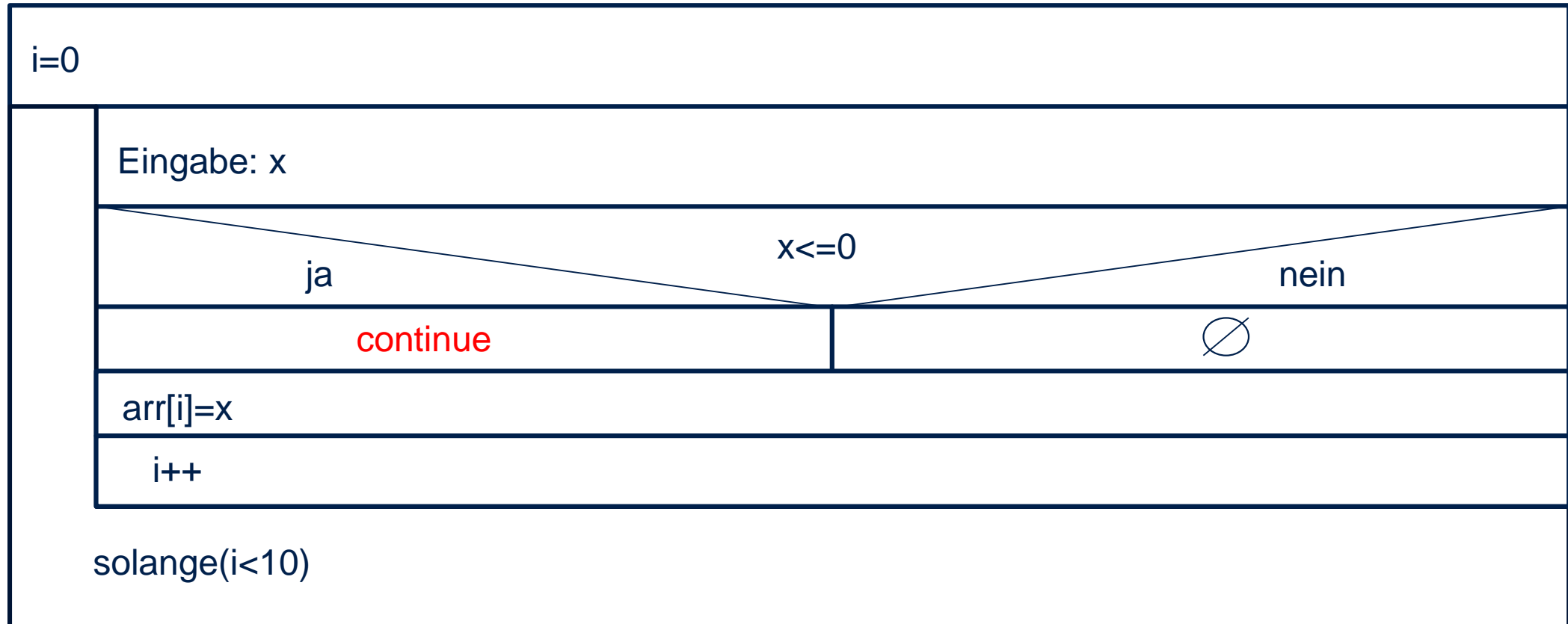
# Sprunganweisung „continue“ – Beispielaufgabe – Struktogramm



# Sprunganweisung „continue“ – Beispielaufgabe – Struktogramm



# Sprunganweisung „continue“ – Beispielaufgabe – Struktogramm



# Sprunganweisung „continue“ – Beispielaufgabe – Pseudocode

**Programm „continue-Beispiel“**

```
{  
    i=0  
  
    {  
        Eingabe: x  
  
        wenn(x<=0)  
        {  
            continue  
        }  
  
        arr[i]=x  
        i++  
    }  
    solange(i<10)  
}
```



# Sprunganweisung „continue“ – Beispielaufgabe – Quellcode

```
#include<stdio.h>

main()
{
    int i=0, x, arr[10];

    do
    {
        printf("Geben Sie bitte eine ganze positive Zahl ein:
");
        fflush(stdin);
        scanf("%d",&x);

        if(x<=0)
        {
            continue;
        }

        arr[i]=x;
        i++;
    }
    while(i<10);
}
```

# Verschachtelte Schleifen + Sprungbefehle – Gemeinsame Übung A\_04\_02\_01



## Aufgabe\_04\_02\_01

Gegeben sei der folgende Pseudocode:

```
Programm_04_02_01
{
    i=0
    solange (i<6)
    {
        // "Aktuelle Lottoszah!" (dieses Durchgangs)
        lotto[i]=Zufallszahl zwischen 1 und 49

        dublette=0
        for (j=0; j<i; j++)
        {
            wenn (lotto[i]==lotto[j])
            {
                // Eine zuvor gelöste Zahl ist identisch ...
                // ... mit der "Aktuellen Lottoszah!"
                dublette=1
                break // Springt aus der for-Schleife
            }
        }

        wenn (dublette==1)
        {
            // Aktueller Durchlauf wird abgebrochen, da i ...
            // ... (wegen der Dublette) nicht hochgezählt werden soll
            continue
        }

        i++
    }
}
```

### Hinweis:

Der dargestellte Pseudocode beschreibt lediglich den eigentlich interessierenden Algorithmus. Auf die Notation einer (möglichen) Kontrollausgabe wurde daher verzichtet.

### Aufgabenstellung:

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

## Hinweis:

Der Pseudocode beschreibt ausschließlich den eigentlich interessierenden **Algorithmus** zur Ermittlung eines Lottotipps.

Auf die Darstellung einer **Kontrollausgabe** wurde der Einfachheit halber verzichtet, da diese keiner weiteren Erklärung bedarf.

**VIELEN DANK  
FÜR IHRE  
AUFMERKSAMKEIT!**