

## **A Arbeits - und Informationsblätter**

## **Arbeitsblatt: Routenplaner**

- Schalte den Computer ein und melde dich am Schulnetz an.
- Öffne das Projekt „**Route2**“ (Laufwerk H:)
- Ergänze in dem Modul „**Ausgabe**“ den fehlenden Quelltext der Prozeduren „**Orte**“ und „**Wege**“.
- Teste dein Programm. Die Ausgabe aller Orte und aller Wege sollte nun funktionieren.
- Speichere dein Projekt ab
- Melde dich am Schulnetz ab

## Arbeitsblatt: Backtracking - Verfahren 1

- Schalte den Computer ein und melde dich am Schulnetz an.
- Öffne das Projekt „**Route3**“ (Laufwerk H:)
- Ergänze in dem Modul „**Berechne**“ den fehlenden Quelltext der Prozedur „**KurzerWeg**“.
- Anschließend muss noch die Option des schnellsten Weges realisiert werden. Hierzu muss in dem Modul „**Berechne**“ die Prozedur „**Schneller\_Weg**“ formuliert werden. Überlege dir dazu, wie sich die Suche nach dem kürzesten Weg von der Suche nach dem schnellsten Weg unterscheidet. Kopiere dann die Prozedur Kurzer\_Weg und ändere sie entsprechend ab.
- Teste anschließend dein Programm.
- Speichere dein Projekt ab
- Melde dich am Schulnetz ab

## Arbeitsblatt: Backtracking - Verfahren 2

- Schalte den Computer ein und melde dich am Schulnetz an.
- Öffne das Projekt „**Route4**“ (Laufwerk H:)
- Ergänze in dem Modul „**Berechne**“ die fehlenden Quelltext der Prozedur „**Suche\_Weg**“.
- Anschließend müssen noch die Hilfsprozeduren Pruefe und Kennung in dem Modul „**Berechne**“ realisiert werden
- Teste anschließend dein Programm. Alle Optionen sollten nun funktionieren.
- Speichere dein Projekt ab
- Melde dich am Schulnetz ab

## Arbeitsblatt: Dijkstraverfahren 1

- Schalte den Computer ein und melde dich am Schulnetz an.
- Öffne das Projekt „**Route7**“ (Laufwerk H:)
- Ergänze in dem Modul „**Berechne**“ den fehlenden Quelltext der Prozedur „**Weg\_ausgeben**“
- Füge dann den Quelltext der Prozeduren „**Initialisiere**“ und „**Erreichbare\_Orte\_bestimmen**“ ein, die die drei Mengen für den Dijkstra - Algorithmus einführen und durchsuchen.
- Ergänze schließlich noch den Quelltext der Prozedur „**Suche\_Weg**“
- Speichere dein Projekt ab (Auch auf Diskette)
- Melde dich am Schulnetz ab

## Arbeitsblatt: Dijkstraverfahren 2

- Schalte den Computer ein und melde dich am Schulnetz an.
- Öffne das Projekt „**Route8**“ (Laufwerk H:)
- Ergänze in dem Modul „**Berechne**“ den fehlenden Quelltext der Prozedur „**Minimale\_Gesamtstrecke\_bestimmen**“.
- Teste dein Programm. Die Option „Bestimmen des kürzesten Weges“ muss nun funktionieren.
- Anschließend muss noch die Option des schnellsten Weges realisiert werden. Hierzu muss in dem Modul „**Berechne**“ die Prozedur „**Schneller\_Weg**“ formuliert werden. Überlege dir dazu, wie sich die Suche nach dem kürzesten Weg von der Suche nach dem schnellsten Weg unterscheidet. Kopiere dann die Prozedur Kurzer\_Weg und ändere sie entsprechend ab.
- Teste anschließend dein Programm. Alle Optionen sollten nun funktionieren.
- Speichere dein Projekt ab (Auch auf Diskette)
- Melde dich am Schulnetz ab

## Arbeitsblatt: Warshallverfahren I

- Schalte den Computer ein und melde dich am Schulnetz an.
- Öffne das Projekt „**Route10**“ (Laufwerk H:)
- Ergänze in dem Modul „**Berechne**“ den fehlenden Quelltext der Prozedur „**Initialisieren**“
- Teste dein Programm. Die Option „Ausgabe der Entfernung\_Warshallmatrix“ bzw. die Ausgabe der Wegematrix muss nun funktionieren. In der Wegematrix dürfen nur direkte Wege eingezeichnet sein, d.h. -1 oder 10000 (kein Weg). Die Warshallmatrix muss der Entfernungsmatrix bzw. der Zeitmatrix entsprechen
- Anschließend muss in dem Modul „**Berechne**“ die Prozedur „**Warshall**“ formuliert werden.
- Teste anschließend dein Programm. Nun müssen obige Matrizen richtig berechnet werden.
- Speichere dein Projekt ab.
- Melde dich am Schulnetz ab

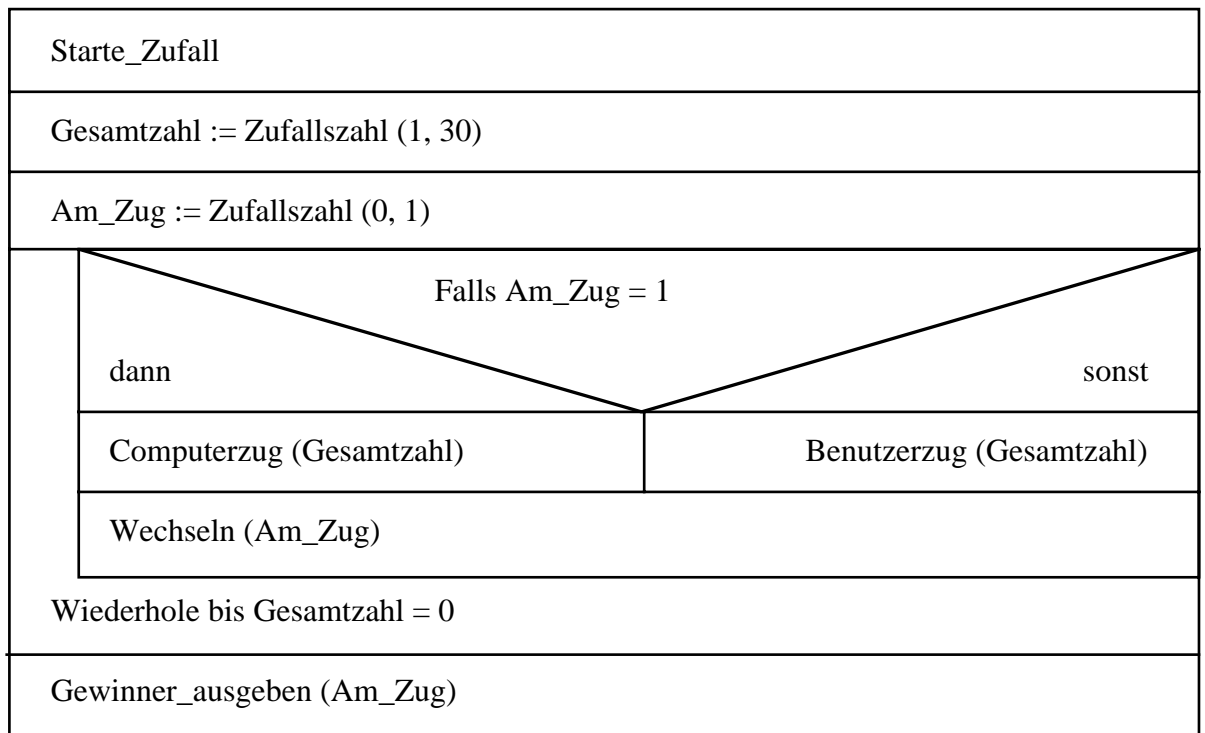
## Arbeitsblatt: Warshallverfahren II

- Schalte den Computer ein und melde dich am Schulnetz an.
- Öffne das Projekt „**Route11**“ (Laufwerk H:)
- Ergänze in dem Modul „**Berechne**“ den fehlenden Quelltext der Prozedur „**Weg\_ausgeben**“
- Anschließend muss in dem Modul „**Berechne**“ in der Prozedur „**KurzerWeg**“ die Ausgabe eingebaut werden.
- Das Gleiche muss nun auch in der Prozedur „**SchellerWeg**“ erfolgen
- Teste nun dein Programm. Alle Optionen müssen nun funktionieren.
- Speichere dein Projekt ab.
- Melde dich am Schulnetz ab



## Arbeitsblatt: Nimmspiel

Das untenstehende Struktogramm zeigt die Struktur des Programmes Nimm.



### **Erläuterung:**

- Starte\_Zufall startet den Zufallszahlengenerator
- Zufallszahl (a, b) liefert eine Zufallszahl zwischen den Zahlen a und b

### **Aufgaben:**

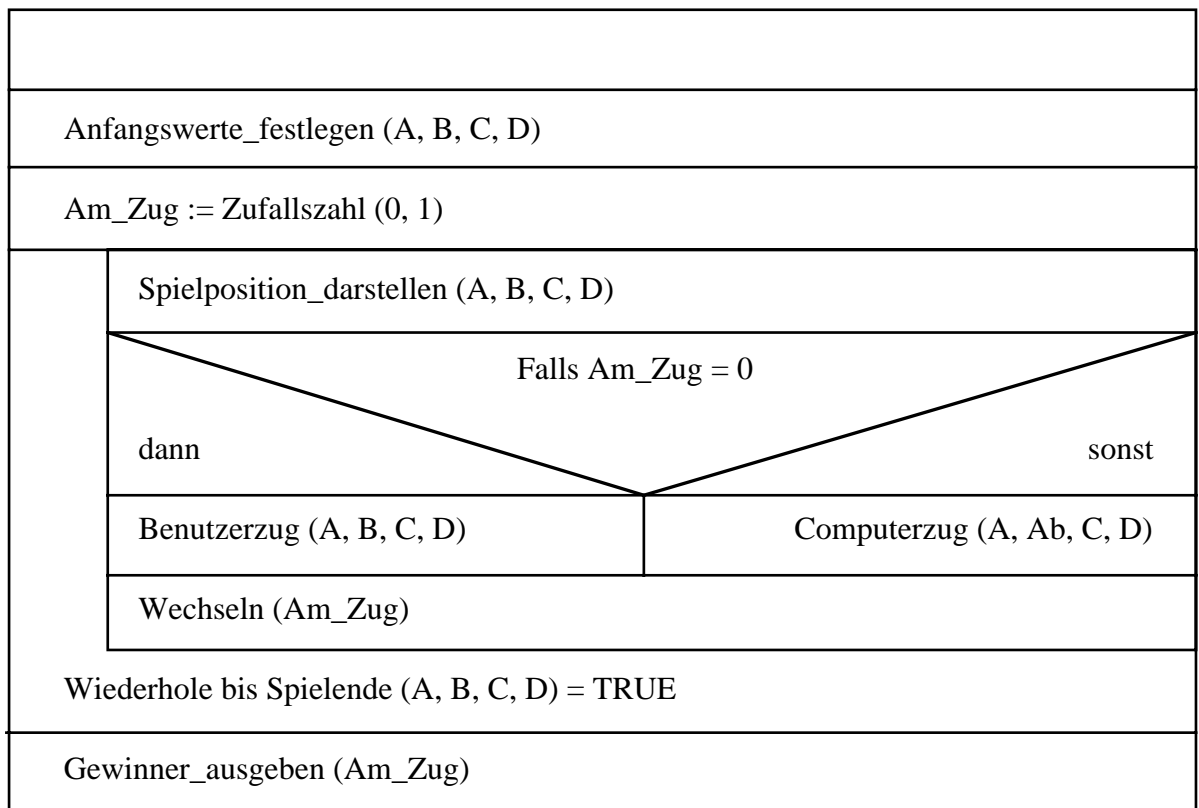
- Entwerfen Sie die Struktogramme der benötigten Prozeduren (Siehe Aufgabenblatt 4, Aufgabe 3 b), c), d) und e) ).
- Übertragen Sie Ihre Struktogramme in Oberon.
  - Öffnen Sie dazu das Projekt „Nimm1“ (Laufwerk H:)
  - Ergänzen Sie in dem Modul „Nimm“ den fehlenden Quelltext der Prozeduren „Computerzug“, „Benutzerzug“, „Wechseln“ und „Gewinner\_ausgeben“
  - Testen Sie nun Ihr Programm. Überprüfen Sie dabei ob Ihre Strategie richtig implementiert wurde,
  - Erstellen Sie einen Ausdruck Ihres Programms und geben Sie als Kommentar am Anfang Ihres Programmes die Namen der Gruppenmitglieder an.

## Arbeitsblatt: Türme von Hanoi

- Öffnen Sie das Projekt „**Hanoi1**“ (Laufwerk :)
- Ergänzen Sie nun die Prozeduren „**Eingabe**“ und „**Verschiebe**“ in dem Modul **Hanoi**  
In der Prozedur „Eingabe“ wird die Anzahl der Scheiben festgelegt, für die das Spiel durchgeführt wird.  
Die Prozedur „Verschiebe“ entspricht dem Struktogramm
- Testen Sie nun Ihr Programm. Lassen Sie sich dazu die Anleitung für vier Steine ausgeben und überprüfen Sie die Lösung.
- Erstellen Sie einen Ausdruck Ihres Programms und geben Sie als Kommentar am Anfang Ihres Programmes die Namen der Gruppenmitglieder an.

## Arbeitsblatt: Wandernde Steine

Das untenstehende Struktogramm zeigt die Struktur des Spieles Steine



### **Erläuterung**

Die Variablen A, B, C und D kennzeichnen die Positionen der Steine. Ihre Zahl ist die Nummer des Feldes. (Siehe Aufgabenblatt 5, Aufgabe 2)

In der Prozedur Anfangswerte\_festlegen werden die Anfangspositionen der Steine bestimmt.

### **Aufgaben**

Erstellen Sie die Struktogramm folgender Prozeduren:

– **Benutzerzug:**

In dieser Prozedur gibt der Benutzer den Namen des Steines ein, der gezogen wird, sowie die Anzahl der Felder, um die der Stein bewegt wird. In der Prozedur muss die Gültigkeit der Eingabe überprüft werden.

– **Computerzug**

Der Computer zieht nach einer eingegebenen Strategie.

– **Gewinner\_ausgeben**

In dieser Prozedur soll der Gewinner ermittelt und am Bildschirm ausgegeben werden.

– **Spielende**

Diese Prozedur stellt fest, ob das Spiel beendet ist und liefert das Ergebnis in einer Booleschen Variablen.

- **Spielposition darstellen**

In dieser Prozedur wird die Spielposition am Bildschirm dargestellt. Hierbei werden die Felder in einer Zeile gezeichnet. Leere Felder werden durch Punkte dargestellt. Besetzte Felder durch den entsprechenden Buchstaben des Steines:

Beispiel: .....A..B.....C.....D

- **Wechseln**

In dieser Prozedur wird bestimmt, wer als Nächster am Zug ist.

Übertragen Sie Ihre Struktogramme in Oberon.

- Öffnen Sie dazu das Projekt „**Steine1**“ (Laufwerk :)
- Ergänzen Sie in dem Modul „**Steine1**“ den fehlenden Quelltext der Prozeduren „**Benutzerzug**“, „**Computerzug**“, „**Gewinner\_ausgeben**“, „**Spielende**“, „**Spielposition\_darstellen**“ und „**Wechseln**“
- Testen Sie nun Ihr Programm. Überprüfen Sie dabei ob Ihre Strategie richtig implementiert wurde,
- Erstellen Sie einen Ausdruck Ihres Programms und geben Sie als Kommentar am Anfang Ihres Programmes die Namen der Gruppenmitglieder an.