B Aufgabenblätter

Aufgabenblatt 1 Klasse 10 a/b/c

Aufgabe 1
Erstellen Sie die Zeitmatrix zu dem Fahrzeitengraph der Umgebung von Wertingen.

Aufgabe 2
Beschreiben Sie umgangssprachlich die Ausgabe der Ortsliste. Erstellen Sie dann das Struktogramm der zugehörigen Prozedur Ausgabe.Orte

Aufgabe 3
Beschreiben Sie umgangssprachlich die Ausgabe der Wegeliste. Erstellen Sie dann das Struktogramm der Prozedur Ausgabe.Wege

Aufgabe 4
Beschreiben Sie umgangssprachlich das Bestimmen der Ortskennung aus dem Ortsnamen und formulieren Sie das Struktogramm der Prozedur Berechne.Kennung

Aufgabe 5
Welche zusätzliche Informationen müsste man noch abspeichern, um eine Karte des Weges zeichnen zu können ?

Aufgabe 6
Beschreiben Sie umgangsprachlich, wie man überprüft, ob man einen Ort bei dem zurückgelegten Weg bereits passiert hat (Prozedur Pruefe) und formulieren Sie das Struktogramm.

Aufgabe 7
Formulieren Sie das Struktogramm der Prozedur Schneller_Weg, die den schnellsten Weg zwischen zwei Orten ermittelt.

Aufgabe 8
Verändern Sie das Struktogramm von Suche_Weg so, dass der verbesserte Backtracking - Algorithmus berücksichtigt wird.

Aufgabenblatt 2 Klasse 10a/b/c

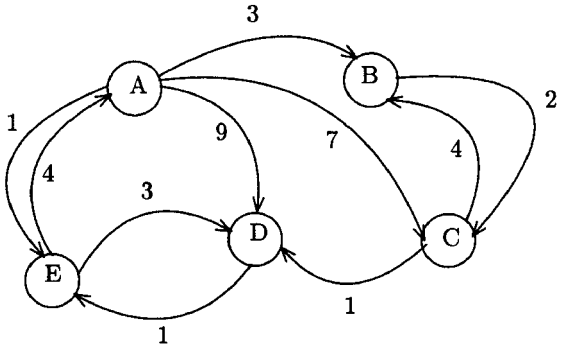
Aufgabe 1
Nebenstehende Adjazenzmatrix stellt die Fahrzeiten zwischen den einzelnen Orten dar.

a) Erstellen Sie zur Adjazenzmatrix den zugehörigen Graphen.
b) Erstellen Sie den Suchbaum für den schnellsten Weg von C nach D

	A	B	C	D	E
A	0	7	2	13	8
B	12	0	/	/	/
C	/	5	0	/	/
D	/	4	/	0	/
E	/	/	/	3	0

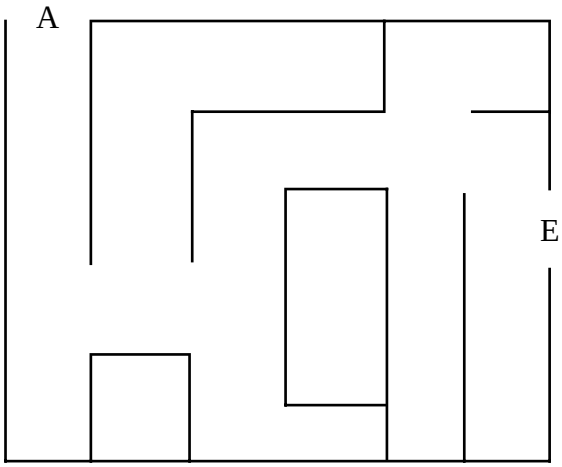
Aufgabe 2

a) Erstellen Sie zum nebenstehenden Graphen die zugehörige Adjazenzmatrix.
b) Erstellen Sie den Suchbaum für den Weg mit der kleinsten Bewertung von E nach C.



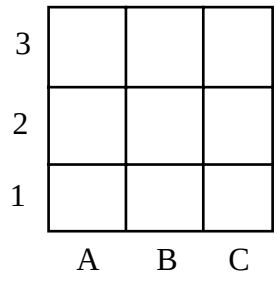
Aufgabe 3
Es soll ein Weg durch das nebenstehende Labyrinth von A nach E gefunden werden

a) Erstellen Sie den zugehörigen Graphen und die Adjazenzmatrix.
b) Entwickeln Sie den Suchbaum für den Weg durch das Labyrinth
c) Formulieren Sie das Struktogramm des entsprechenden Backtracking - Verfahrens



Aufgabe 4
Springerproblem
Von einem beliebigen Punkt eines Schachfeldes sollen mit einem Springer alle Felder besucht werden, ohne ein Feld doppelt zu berühren.

a) Erstellen Sie den Graphen für das nebenstehende 3x3 - Schachfeld.
b) Erstellen Sie den Suchbaum, wenn der Springer am Anfang auf dem Feld A1 steht. Unter welchen Bedingungen endet der Suchbaum ?
c) Formulieren Sie das Struktogramm des entsprechenden Backtracking - Verfahrens



Aufgabe 5

Rucksackproblem

Ein Wanderer packt seinen Rucksack. Er nimmt einige Nahrungsmittel in Dosen mit. Damit der Rucksack nicht zu schwer wird, darf das Gesamtgewicht der Dosen 4 kg nicht überschreiten. Bei der Auswahl der Dosen soll der gesamte Nährwert möglichst groß werden.

Dose	A	B	C	D	E	F
Gewicht	1,5	2	0,7	1,9	0,4	1,2
Nährwert	20	17	10	25	14	13

a) Erstellen Sie den Suchbaum für die optimale Kombination, wenn als erste Dose A und als zweite Dose B, C oder D eingepackt wird. Unter welchen Bedingungen endet der Suchbaum ?

b) Formulieren Sie das Struktogramm des entsprechenden Backtracking - Verfahrens

Aufgabenblatt 3 Klasse 10 a/b/c

Aufgabe 1
Erstellen Sie den Graphen für die Wegesuche von Meitingen nach Wertingen nach dem Dijkstra - Verfahren.

Aufgabe 2
Beim Dijkstra - Verfahren wird der Weg rekursiv aus der Liste Mengen ausgelesen. Veranschaulichen Sie den Ablauf der rekursiven Prozedur Weg_ausgeben für den Weg Biberbach - Meitingen - Langenreichen - Hohenreichen.

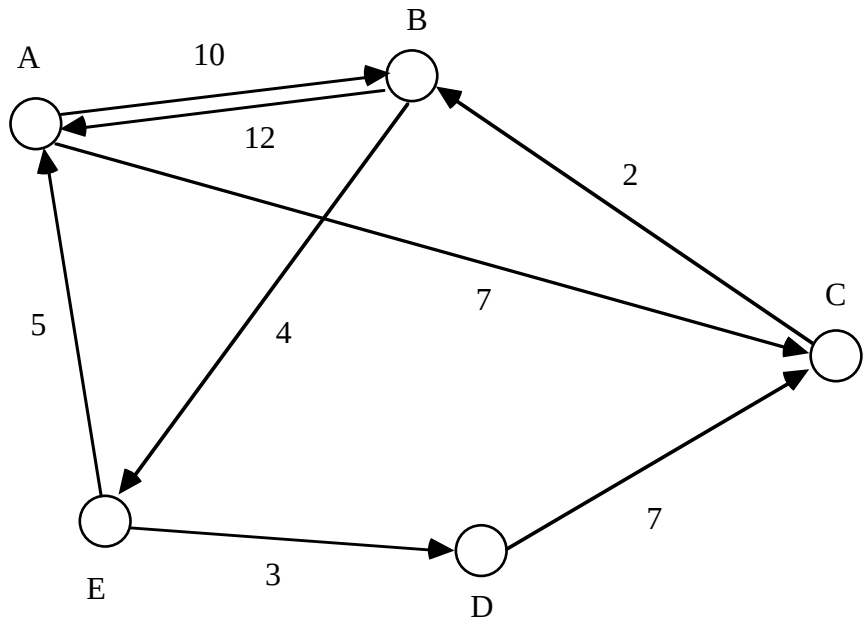
Aufgabe 3
Erstellen Sie das Struktogramm der Prozedur Minimale_Gesamtstrecke_finden, die aus der Menge der erreichbaren Orte in der Liste Mengen, den mit der kürzesten Gesamtstrecke auswählt.

Aufgabe 4
Wie muss die Abbruchbedingung im Dijkstra-Verfahren formuliert werden, damit man die minimalen Wege von einem Startknoten zu allen anderen Knoten findet ?

Aufgabe 5
Ermitteln Sie für das Beispiel des Skripts die folgenden Matrizen mit den Flügen über London, Amsterdam und Boston.

Aufgabe 6
Entwickeln Sie die restlichen Wegematrizen nach dem Verfahren von Warshall zu dem Beispiel des Skripts.

Aufgabe 7
Bestimme zu nebenstehendem Graphen die Matrix der minimalen Verbindungen nach dem Warshall - Verfahren sowie die zugehörige Wegematrix. Ermittle dann daraus den minimalen Weg von A nach B, bzw. von D nach A.



Aufgabenblatt 4 Klasse 10 a/b/c

Aufgabe 1
Entwickeln Sie einen Graph für den Spielverlauf des Nimm-Spieles für n = 8 und k = 2. Die Knoten sind dabei die Hölzchenzahlen.

Aufgabe 2
Überlegen Sie sich eine Spielstrategie für das Nimm-Spiel. Unterscheiden Sie dazu im Graphen Verlust- und Gewinnpositionen. Zeichnen Sie den Graphen so, dass die Verlust- und die Gewinnpositionen übereinander liegen. Entwickeln Sie eine rechnerische Methode mit der sich die Gewinn- und Verlustpositionen erkennen lassen.

Aufgabe 3
a) Entwickeln Sie ein Struktogramm für ein Nimmspiel, bei dem der Rechner gegen einen Benutzer spielt. In dem Struktogramm sollen für den Zug des Computers die Prozedur Computerzug, für den Zug des Benutzers die Prozedur Benutzerzug, für das Wechseln des Spielers zwischen Computer und Benutzer die Prozedur Wechseln und für die Ausgabe des Gewinners die Prozedur Gewinner_ausgeben verwendet werden. Die Anfangszahl der Hölzchen und der Spieler der beginnt werden durch den Zufallszahlengenerator bestimmt.
b) Entwickeln Sie ein Struktogramm für die Prozedur Computerzug mit dem Parameter Gesamtzahl.
c) Entwickeln Sie ein Struktogramm für die Prozedur Benutzerzug mit dem Parameter Gesamtzahl.
d) Entwickeln Sie ein Struktogramm für die Prozedur Wechseln mit dem Parameter Am_Zug.
e) Entwickeln Sie ein Struktogramm für die Prozedur Gewinner_ausgeben mit dem Parameter Am_Zug.

Aufgabe 4
Transportieren Sie die Scheiben von A nach B. Lösen Sie die Aufgabe für n = 2 und n = 3. Erstellen Sie den dazugehörigen Spielegraph. Ein Knoten gibt die Position der Scheiben an, z.B. (A / B / A) für 1. Scheibe auf A, 2. Scheibe auf B und 3. Scheibe auf A. Wie lässt sich die Lösung für n = 2 in der Lösung n = 3 verwenden ? Wie lässt sich damit die Lösung für n = 4, 5 ..... ermitteln. ?

Aufgabe 5
Erstellen Sie ein Struktogramm für die Lösung der Türme von Hanoi. Entwerfen Sie hierzu die Prozedur Verschiebe (Anzahl : INTEGER, Start, Ziel, Hilfe : CHAR), die ausgibt , wie die Scheiben verschoben werden müssen.
Anzahl gibt die Zahl der Scheiben an, Start ist der Ausgangsstab, Ziel der Zielstab und Hilfe der drittte Stab.

Aufgabenblatt 5 Klasse 10a/b/c

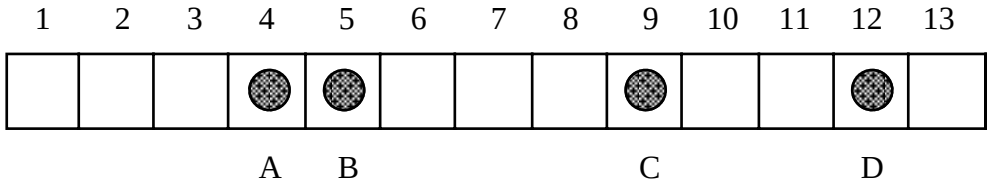
Aufgabe 1

Betrachten Sie das Nimmspiel in folgender Variante. Die maximale Zahl der wegzunehmenden Stäbchen sei 4, die minimale Zahl sei 2. Der Spieler, der den letzten Zug durchgeführt hat, habe gewonnen. (Ist nur noch ein Stäbchen vorhanden, so kann kein Zug mehr durchgeführt werden.)

a) Erstellen Sie den Spielgraph für eine Anfangszahl von 10 Stäbchen
b) Unterscheiden Sie zwischen Gewinnpositionen und Verlustpositionen. Entwickeln Sie eine Spielstrategie und ein Erkennungsmerkmal für Gewinnpositionen.

Aufgabe 2

Das Spiel „Wandernde Steine“ hat folgende Anleitung. Auf einem nach rechts unbegrenzten Streifen, der in Felder eingeteilt ist, liegen n Spielsteine. Ein Zug besteht darin, einen der Steine dem linken Streifenende zu nähern, ohne dabei ein besetztes Feld erneut zu besetzen oder zu überspringen. Die beiden Spieler ziehen abwechselnd und der letzte Zug gewinnt.
Beispiel für n = 4



a) Erstellen Sie den Spielgraph für n = 2. Jede Position wird als Kreis mit zwei Zahlen dargestellt, welche die Feldindices der beiden Steine A und B angeben. Beginnen Sie mit der Anfangsposition (4 / 6).
b) Unterscheiden Sie Gewinn- und Verlustpositionen. Entwickeln Sie eine Gewinnstrategie und ein Erkennungsmerkmal für Gewinnpositionen
c) Erstellen Sie den Spielgraphen für den Fall n = 4. Beginnen Sie mit der Anfangsstellung (2 / 3 / 5 / 7)
d) Unterscheiden Sie Gewinn- und Verlustpositionen. Entwickeln Sie eine Gewinnstrategie und ein Erkennungsmerkmal für Gewinnpositionen
(Hinweis: Betrachten Sie die Abstände zwischen A und B bzw. zwischen C und D)

Aufgabe 3

Betrachten Sie folgendes Spiel. Zwei Spieler nennen abwechselnd eine Zahl zwischen 1 und 10. Diese Zahlen werden nun aufsummiert. Der Spieler, der mit seiner Zahl die Summe 100 erreicht, hat gewonnen.

a) Erstellen Sie den Spielgraphen für die ersten beiden Spielzüge
b) Entwickeln Sie eine geeignete Spielstrategie und ein Erkennungsmerkmal für Gewinnpositionen.

Aufgabenblatt 6 Klasse 10a/b/c

Aufgabe 1
Erzeugen Sie eine 2. Instanz der Klasse cPerson mit Namen Obelix und lasse diesen auf dem Bildschirm erscheinen und wieder verschwinden.

Aufgabe 2
Implementieren Sie in der Klasse Person eine neue Methode „Bewege_dich (Delta_X, Delta_Y)“, die das Objekt um Delta_X und Delta_Y verschiebt. Beachte dabei die Bildschirmränder . x kann Werte zwischen 1 und 80 annehmen, y zwischen 1 und 25. Bei Erreichen des Bildschirmrandes soll das Objekt reflektiert werden.

Aufgabe 3
Implementieren Sie in der Klasse Person eine neue Methode „Lasse_dich_Steuern (Taste)“ mit der sich je nach Tastendruck die Person nach oben, links, rechts oder unten steuern lässt. Es sollen folgende Tastenzuordnungen gelten: w = oben, a = links, s = rechts und y = unten.

Aufgabe 4
Implementieren Sie in der Klasse Person folgende neue Methoden „Sage_X“, „Sage_Y“ und „Sage_Form“, die die Koordinate bzw. den Buchstaben als Ergebnis zurückgibt.
(Hinweis: Dies wird mit Hilfe von RETURN erreicht.)

Aufgabe 5
Implementieren Sie in der Klasse Person eine neue Methode Folge (Person) bzw. Fliehe_vor (Person) mit denen das Objekt sich einem anderen Objekt annähern, bzw. vor ihm fliehen kann.

Aufgabe 6
Entwerfen Sie die Klasse cRoemer und die zugehörige Methode „Anlegen“, die Anfangsort und Zeichen festlegt und jedem Roemer ein Schwert und einen Helm gibt.

Aufgabe 7
Entwerfen Sie eine Methode „trifft (Person)“ der Klasse Person, die überprüft ob zwei Objekte benachbart sind.

Aufgabe 8
Zwischen Galliern und Römern kommt es normalerweise zum Kampf. Hat der Gallier einen Zaubertrank getrunken, so flieht der Römer, sonst versteckt sich der Gallier.
Entwerfen Sie eine Methode „Kämpfe_mit“ der Klasse Gallier , die obige Beschreibung der Kämpfe realisiert.

Aufgabe 9
Entwerfen Sie eine Klasse der Druiden als Unterklasse der Gallier. Sie haben zusätzlich einen Vorrat von Zaubertrank, mit dem sie andere Gallier versorgen können. Die Methode „Anlegen“ soll die Anfangswerte festlegen. Dabei hat ein Druide selbstverständlich einen Zaubertrank getrunken und kann 10 Gallier versorgen.

Aufgabe 10

Entwerfen Sie eine Methode „Gib_Zaubertrank_an“ der Klasse Druiden, die beim Treffen mit einem Gallier diesem einen Zaubertrank gibt, sofern der Vorrat des Druiden noch nicht aufgebraucht und der Gallier keinen Zaubertrank hat. Welche Methoden benötigt man noch dazu ? Entwerfen Sie diese Methoden.

Aufgabenblatt 7 Klasse 10a/b/c

Aufgabe 1

Bei einer Bank gibt es verschiedene Arten von Konten, Girokonten und Sparkonten. Beide sind durch eine Kontonummer und einen Konteninhaber gekennzeichnet. Bei Sparkonten werden die Einlagen im Gegensatz zum Girokonto verzinst. Dafür kann das Sparkonto nicht überzogen werden, was beim Girokonto bis zu einer gewissen Grenze möglich ist.

a) Entwerfen Sie die Objekte Konto, Girokonto und Sparkonto mit den nötigen Attributen, wobei die beiden Objekte Girokonto und Sparkonto als Unterklassen des Objekts Konto realisiert werden sollen.
b) Entwickeln Sie für das Objekt Girokonto eine Methode Abheben, die einen Geldbetrag vom Konto abhebt, solange damit die Grenze nicht überschritten wird.
c) Entwickeln Sie für das Objekt Girokonto eine Methode Überzogen, die überprüft, ob das Konto überzogen ist
d) Entwickeln Sie eine Methode Zins_zahlen für das Objekt Sparkonto, die den Zins ermittelt und dem Konto gutschreibt.

Aufgabe 2

In einem Auslieferungslager werden Waren gespeichert. Von jeder Warenart werden folgeden Daten gespeichert: Artikelnummer, Stückpreis, Platzbedarf (in ganzzahligen Platzeinheiten) und die Anzahl der vorhandenen Stücke. Das Auslieferungslager hat einen maximalen Platz und zur Vereinfachung der Verwaltung wird der belegte Platz gespeichert. Zur Modellierung der Warenverwaltung werden zwei Modelle entworfen, eines für eine Warenart und das andere für das Warenlager.

a) Entwerfen Sie die beiden Objekte mit den entsprechenden Attributen und der Methode Init, die die Attribute auf die Anfangswerte setzt.
b) Zur Lagerverwaltung benötigt man zwei Methoden des Objekts Warenlager, die den freien Platz ausgeben, bzw. ändern. Entwerfen Sie die beiden Methoden Sag_Freier_Platz und Lagere_ein, die als Parameter jeweils die Zahl der freien bzw. benötigten Lagereinheiten haben.
c) Entwerfen Sie die Methode Anliefern des Objekts Warenart, das als Parameter die angelieferte Stückzahl hat. Diese Methode prüft, ob im Lager noch genügend Platz ist und ändert gegebenenfalls den freien Platz und die Anzahl der vorhandenen Stücke. Als Ergebnis wird gemeldet, ob die Ware gespeichert werden kann.
d) Entwerfen Sie die Methode Ausliefern des Objekts Warenart, das als Parameter die auszuliefernde Stückzahl hat. Diese Methode prüft, ob im Lager noch genügend Stücke vorhanden sind, berechnet den Gesamtpreis und ändert gegebenenfalls den freien Platz und die Anzahl der vorhandenen Stücke. Als Ergebnis wird gemeldet, ob die Ware geliefert werden kann. Der Gesamtpreis wird als VAR - Parameter übergeben.

Aufgabe 3

Ein Roboter soll auf Reize aus der Umgebung reagieren können. Mit ihm soll nun der Weg durch ein Labyrinth gesucht werden. Zur Modellierung werden zwei Objekte verwendet, eines für den Roboter und eines für das Labyrinth. Für den Roboter benötigt man seine Koordinaten, sowie seine Bewegungsrichtung, während das Labyrinth durch ein Feld beschrieben wird, in dem angegeben ist, ob ein bestimmter Punkt (x, y) frei ist oder ob dort ein Hindernis steht.

a) Entwerfen Sie die beiden Objekte mit den entsprechenden Attributen
b) Entwerfen Sie die Methode Init für das Objekt Roboter, die die Anfangswerte für die Position (x, y) und die Bewegungsrichtung festlegt. Position und Bewegungsrichtung werden mit Parameter übergeben.
c) Entwerfen Sie die Methode Sag_frei des Objekts Labyrinth. Das Ergebnis dieser Methode sagt ob das Feld mit den, als Parameter übergebenen Koordinaten (x, y) frei ist.
d) Entwerfen Sie die Prozeduren Sage_x, Sage_y und Sage_Richtung des Objekts Roboter, die als Ergebnis die aktuelle x- bzw. y-Koordinate, bzw. die Bewegungsrichtung liefern.
e) Als Strategie zur Wegsuche wird folgendes Konzept angewendet. Jedes Mal wenn der Roboter an ein Hindernis stösst, dreht er sich um 90° nach links. Entwerfen Sie eine Methode Bewege_Dich mit der sich der Roboter um ein Feld in seiner Bewegungsrichtung vorwärts bewegt, bzw. falls dies nicht möglich sich um 90° dreht.
f) Entwerfen Sie ein Hauptprogramm mit dem eine Roboter mit Hilfe der obigen Methode Bewege_Dich ein Labyrinth vom Startpunkt (1 / 1) bis zum Endpunkt (1 / 14) durchlauft. Bei welcher Art von Labyrinthen versagt die obige Strategie ?

Aufgabenblatt 8 Klasse 10a/b/c

Entwerfen Sie eine Objektstruktur mit Klassen, Attributen und Methoden zu folgenden Beschreibungen

Aufgabe 1
Beschreibung eines einarmigen Banditen (Spielautomat)

- In dem Spielautomat werden in drei Fenstern drei Ziffern dargestellt.
- Ein Spiel kostet 50 Cent
- In einen Geldschlitz kann Geld eingeworfen werden
- Ist das Kapital größer gleich dem Einsatz, kann das Spiel durch einen Hebel gestartet werden. In den Fenstern drehen dann die Zahlräder. Das Kapital wir dabei um den Einsatz vermindert.
- Durch eine weitere Taste wird das Spiel gestoppt. Die Drehung der Zahlräder kommt dabei zum Stehen.
- Je nach Stellung der Zahlräder gibt es unterschiedliche Gewinnmöglichkeiten.
  - Zeigen alle drei Räder die Null, so gewinnt man den vierfachen Einsatz.
  - Zeigen zwei Räder die Null, so gewinnt man den zweifachen Einsatz
  - Zeigt ein Rad die Null, so gewinnt man den einfachen Einsatz.
  - Zeigen alle drei Räder die gleiche Ziffer, so gewinnt man den doppelten Einsatz
  - Bei allen anderen Stellungen der Räder, gewinnt man nichts
- Ein Gewinn wird sofort zum Kapital addiert.
- Mit der Taste „Auszahlen“ wird das Kapital ausgezahlt.

Aufgabe 2
Beschreibung eines Taschenrechners

- Der Taschenrechner besteht aus mehreren Tasten und einer Anzeige
- Für jede Ziffer gibt es eine Taste
- Es gibt Tasten für Addition, Subtraktion, Multiplikation und Division
- Es gibt eine = - Taste zur Berechnung des Ergebnisses und eine Löschtaste zum Löschen der Anzeige
- Wird eine Zifferntaste gedrückt, so wird die Ziffer rechts an die dargestellte Zahl angefügt.
- Wird eine Rechentaste gedrückt, so wird die angezeigte Zahl mit dem internen Ergebnis entsprechend verrechnet und die Anzeige auf Null gesetzt..
- Wird die Löschtaste gedrückt so wird die Anzeige gelöscht und auf Null gesetzt.
- Bei Drücken der = - Taste wird der Wert des internen Ergebnisses dargestellt und anschließend auf Null gesetzt.

Aufgabe 3
Beschreibung des Spiels „Schiffe versenken“

- Zwei Personen spielen gegeneinander
- Jede Person hat für das Spiel zwei 10 x 10 Gitter. Eine eigenes Gitter auf dem die eigenen Schiffe dargestellt werden und ein Feindgitter auf dem die Schüsse auf den Gegner eingetragen werden.
- Jeder Spieler hat am Anfang 10 Schlachtschiffe, die er auf seinem eigenen Gitter plaziert.
- Die Spieler sind abwechselnd an der Reihe und geben einen Schuß ab. Dieser Schuß wird dabei in das Feindgitter eingetragen. Wurde dabei ein Treffer erzielt, so kann ein weiterer Schuß abgegeben werden. Außerdem wird bei einem Treffer angegeben ob das Schiff versenkt wurde oder nicht. Wurde kein Treffer erzielt, so ist der andere Spieler an der Reihe.
- Das Spiel endet, wenn ein Spieler alle Schiffe des Gegners versenkt hat

Aufgabenblatt 9 Klasse 10a/b/c

Aufgabe 1

Entwerfen Sie die Struktogramme der Methoden der Klasse Labyrinth

a) PROCEDURE (Mein_Labyrinth : cLabyrinth) Init;
   Die Methode Init soll die die einzelnen Felder des Labyrinths festlegen und den Anfangswert der Punktezahl bestimmen.
b) PROCEDURE (Mein_Labyrinth : cLabyrinth) Zeige_Feld (x, y : INTEGER);
   Die Methode zeichnet den Inhalt des Feldes mit den Koordinaten x,y
c) PROCEDURE (Mein_Labyrinth : cLabyrinth) Zeige_Dich;
   Die Methode zeichnet das ganze Labyrinth
d) PROCEDURE (Mein_Labyrinth : cLabyrinth) Sage_Weg_frei (x, y : INTEGER)
   : BOOLEAN;
   Die Methode gibt an, ob das Feld mit den Koordinaten x,y frei ist.
e) PROCEDURE (Mein_Labyrinth : cLabyrinth) Sage_Punkt (x, y : INTEGER)
   : BOOLEAN;
   Die Methode gibt an, ob sich an der Stelle x,y ein Punkt befindet.
f) PROCEDURE (Mein_Labyrinth : cLabyrinth) Sage_dicker_Punkt (x, y : INTEGER)
   : BOOLEAN;
   Die Methode gibt an, ob sich an der Stelle x,y ein dicker Punkt befindet.
g) PROCEDURE (Mein_Labyrinth : cLabyrinth) Loesche_Punkt (x, y : INTEGER);
   Die Methode löscht den Punkt an der Stelle x,y
h) PROCEDURE (Mein_Labyrinth : cLabyrinth) Sage_Punkteanzahl (): INTEGER;
   Die Methode liefert die Anzahl der Punkte
i) PROCEDURE (Mein_Labyrinth : cLabyrinth) Sage_xGroesse () : INTEGER;
   Die Methode liefert die Größe des Labyrinths in x-Richtung
k) PROCEDURE (Mein_Labyrinth : cLabyrinth) Sage_yGroesse () : INTEGER;
   Die Methode liefert die Größe des Labyrinths in y-Richtung

Aufgabe 2

Entwerfen Sie die Struktogramme der Methoden der Klasse Spiel

a) PROCEDURE (Mein_Spiel : cSpiel) Init (Anzahl : INTEGER);
   Die Methode legt die Anfangswerte des Punktestandes und der Leben fest. Die Zahl der Leben wird über den Parameter Anzahl festgelegt.
b) PROCEDURE (Mein_Spiel : cSpiel) Erhoehe_Punktestand (Anzahl : INTEGER);
   Die Methode erhöht den Punktestand um Anzahl
c) PROCEDURE (Mein_Spiel : cSpiel) Sage_Punktestand () : INTEGER;
   Die Methode liefert den Punktestand.
d) PROCEDURE (Mein_Spiel : cSpiel) Reduziere_Leben;
   Die Methode reduziert den Punktestand um 1.
e) PROCEDURE (Mein_Spiel : cSpiel) Sage_Leben () : INTEGER;
   Die Methode liefert die Anzahl der verfügbaren Leben.

Aufgabe 3

Entwerfen Sie die Struktogramme der Methoden der Klasse Tastatur

a) PROCEDURE (Meine_Tastatur : cTastatur) Init;
   Die Methode belegt die Attribute mit den Anfangswerten
b) PROCEDURE (Meine_Tastatur : cTastatur) Sage_gedrueckt () : BOOLEAN;
   Die Methode gibt an, ob die Taste gedrückt wurde und speichert das Zeichen in dem Attribut Taste.
c) PROCEDURE (Meine_Tastatur : cTastatur) Sage_Taste () : CHAR;
   Die Methode liefert das Zeichen der gedrückten Taste

Aufgabe 4

Entwerfen Sie die Struktogramme der Methoden der Klasse Figur

a) PROCEDURE (Meine_Figur : cFigur) Init (x, y : INTEGER; Form : CHAR);
   Die Methode belegt die Anfangswerte der Figur
b) PROCEDURE (Meine_Figur : cFigur) Zeige_Dich;
   Die Methode zeichnet das Zeichen an der entsprechenden Stelle und in der entsprechenden Farbe auf den Bildschirm
c) PROCEDURE (Meine_Figur : cFigur) Verstecke_Dich (Mein_Labyrinth : cLabyrinth);
   Die Methode versteckt die Figur
d) PROCEDURE (Meine_Figur : cFigur) Sage_x () : INTEGER;
   Die Methode liefert die x-Koordinate der Figur
e) PROCEDURE (Meine_Figur : cFigur) Sage_y () : INTEGER;
   Die Methode liefert die y-Koordinate der Figur
f) PROCEDURE (Meine_Figur : cFigur) Bewege_dich (dx, dy : INTEGER;
   Mein_Labyrinth : cLabyrinth)
   Die Methode bewegt die Figur durch das Labyrinth. Dabei muss immer überprüft werden, ob der Weg frei ist.
g) PROCEDURE (Meine_Figur : cFigur) Lasse_Dich_Steuern (Taste : CHAR;
   oben, unten, links, rechts : CHAR;Mein_Labyrinth : cLabyrinth)
   Die Methode steuert die Figur über Tastendruck
h) PROCEDURE (Meine_Figur : cFigur) Folge (Zweite_Figur : cFigur
   Mein_Labyrinth : cLabyrinth)
   Die Methode lässt Meine_Figur der zweiten Figur folgen.
i) PROCEDURE (Meine_Figur : cFigur) Fliehe_vor (Zweite_Figur : cFigur
   Mein_Labyrinth : cLabyrinth)
   Die Methode lässt Meine_Figur vor der zweiten Figur fliehen.
k) PROCEDURE (Meine_Figur : cFigur) Trifft (Zweite_Figur : cFigur) : BOOLEAN;
   Die Methode ermittelt ob Meine_Figur mit der zweiten Figur zusammengetroffen ist. In diesem Fall heißt dies dass die Koordinaten gleich sein müssen.

Aufgabe 5

Entwerfen Sie die Struktogramme der Methoden der Klasse PacMan

a) PROCEDURE (Mein_PacMan : cPacMan) Init (x, y, : INTEGER; Form : CHAR);
   Die Methode belegt die Anfangswerte des PacMan
b) PROCEDURE (Mein_PacMan : cPacMan) Reduziere_Uhr;
   Die Methode stellt die Uhr zurück und falls die Uhr auf null steht, so wird PacMan verwundbar.
c) PROCEDURE (Mein_PacMan : cPacMan) Sag_unverwundbar () : BOOLEAN;
   Die Methode gibt an ob PacMan unverwundbar ist.
d) PROCEDURE (Mein_PacMan : cPacMan) Werde_unverwundbar;
   Die Methode macht PacMan unverwundbar, ändert seine Farbe und stellt die Uhr auf 50 Einheiten.
e) PROCEDURE (Mein_PacMan : cPacMan) Werde_verwundbar;
   Die Methode macht PacMan verwundbar und setzt die Farbe zurück.
f) PROCEDURE (Mein_PacMan : cPacMan) Stirbt (Mein_Labyrinth : cLabyrinth);
   Die Methode lässt PacMan sterben, in dem er sich versteckt.

Aufgabe 6

Entwerfen Sie die Struktogramme der Methoden der Klasse Geist

a) PROCEDURE (Mein_Geist : cGeist) Init (x, y, : INTEGER; Form : CHAR);
   Die Methode belegt die Anfangswerte des Geists
b) PROCEDURE (Mein_Geist : cGeist) Sage_Punkte () : INTEGER;
   Die Methode liefert die Anzahl der Punkte die PacMan bein Tod dieses Geistes erhält.
c) PROCEDURE (Mein_Geist : cGeist) Sage_am_Leben () : BOOLEAN;
   Die Methode überprüft, ob der Geist am Leben ist.
d) PROCEDURE (Mein_Geist : cGeist) Stirbt (Mein_Labyrinth : cLabyrinth);
   Die Methode lässt den Geist sterben, in dem sie ihn versteckt und seine Koordianten außerhalb des Bildschirms setzt.