

C-ÜBUNGSAUFGABEN STRUKTUREN 1

1)

Ein Wagen wird durch seinen Markennamen, die Anzahl der Türen und den Hubraum beschrieben.

Realisieren Sie dies durch eine Struktur in der Programmiersprache C.

2)

Initialisieren Sie eine Variable mit dem Datentyp Wagen mit von Ihnen ausgesuchten Daten.

3)

Schreiben Sie die Funktion `printWagen`, die als Parameter eine Variable mit dem Datentyp Wagen bekommt und den Namen, den Hubraum und die Anzahl der Türen des Wagens auf dem Bildschirm ausgibt.

4)

a) Schreiben Sie die Funktion `setWagen` mit den folgenden 4 Parametern:

eine Variable mit dem Datentyp Wagen, den zu setzenden Markennamen, die zu setzende Anzahl der Türen und den zu setzenden Hubraum.

b) Verändern Sie mit Hilfe dieser Funktion `setWagen` den Wert der Variablen bei 2) auf "Opel", 2000 ccm, 5 Türen.

5)

In einem Verwaltungsprogramm soll über Tastatur die Anzahl der Wagen in einem Fuhrpark eingegeben und von diesen Wagen jeweils der Markennamen, der Hubraum und die Anzahl der Türen eingegeben werden.

Realisieren Sie das C-Programm.

6)

6.1)

Ein Wagen wird durch seinen Markennamen, die Anzahl der Türen und den Hubraum beschrieben.

Eine Person wird durch ihre Personalnummer und den Nachnamen beschrieben.

Eine Garage wird durch ihren Besitzer (Person), die in ihr stehenden maximal 10 Wagen und den monatlichen Mietbetrag beschrieben.

Realisieren Sie Person und Wagen jeweils durch eine C-Struktur.

Realisieren Sie auch Garage unter Zuhilfenahme der Strukturen Person und Wagen.

6.2)

Initialisieren Sie eine Variable mit dem Datentyp Garage mit von Ihnen ausgesuchten Daten.

6.3)

Schreiben Sie die Funktion `printWagen`, die als Parameter eine Variable mit dem Datentyp Wagen bekommt und den Markennamen, die Anzahl der Türen und den Hubraum des Wagens auf dem Bildschirm ausgibt.

6.4)

Schreiben Sie die Funktion `setWagen` mit den folgenden 4 Parametern:

eine Variable mit dem Datentyp Wagen, den zu setzenden Markennamen, die zu setzende Anzahl der Türen und den zu setzenden Hubraum.

6.5)

Schreiben Sie die Funktion `printPerson`, die als Parameter eine Variable mit dem Datentyp `Person` bekommt und die Personalnummer und den Nachnamen einer Person auf dem Bildschirm ausgibt.

6.6)

Schreiben Sie die Funktion `setPerson` mit den folgenden 3 Parametern:
eine Variable mit dem Datentyp `Person`, der zu setzenden Personalnummer und den zu setzenden Nachnamen.

6.7)

Schreiben Sie die Funktion `printGarage`, die als Parameter eine Variable mit dem Datentyp `Garage` bekommt und den Garagenbesitzer (speziell: Personalnummer und Nachnamen), die Autos (speziell: Markennamen, die Anzahl der Türen und den Hubraum) in der Garage und die Garagenmiete auf dem Bildschirm ausgibt.

6.8)

Schreiben Sie die Funktion `setGarage` mit den folgenden 2 Parametern:
eine Variable mit dem Datentyp `Garage` (und den sich dort befindlichen Daten) und eine Variable mit dem Datentyp `Garage`, die eine Kopie davon ist.

6.9)

In einem Verwaltungsprogramm soll über Tastatur die Anzahl der Garagen in einem Ort und von diesen Garagen jeweils die Garagenbesitzer (speziell: Personalnummer und Nachnamen), die Autos (speziell: Markennamen, die Anzahl der Türen und den Hubraum) in der Garage und die Garagenmieten eingegeben werden können.

6.10)

Die in dem vorigen Verwaltungsprogramm eingegebenen Daten sollen dann für jede Garage auf dem Bildschirm ausgegeben werden. Benutzen Sie dazu u.a. die Funktion `printGarage`.

6.11)

Die in dem vorigen Verwaltungsprogramm eingegebenen Mieten sollen für jede Garage um 100 Euro erhöht werden. Benutzen Sie dazu die Funktion `setGarage`.

6.12)

Geben Sie mit Hilfe der Funktion `printGarage` die Daten aller Garagen auf dem Bildschirm aus.

C-ÜBUNGSAUFGABEN STRUKTUREN 2

7)

7.1)

Ein Zimmer wird durch seine Länge, Breite, Höhe und seine Nutzung (Schlafzimmer, Wohnzimmer, usw.) beschrieben.

Eine Person wird durch ihre Personalnummer und den Nachnamen beschrieben.

Eine Wohnung wird durch seine maximal 10 Zimmer, seinen Besitzer und seinen Mieter beschrieben. Realisieren Sie jeweils Person und Zimmer durch eine Struktur.

Realisieren Sie auch Wohnung unter Zuhilfenahme der Strukturen Person und Zimmer.

7.2)

Initialisieren Sie eine Variable mit dem Datentyp Wohnung mit von Ihnen ausgesuchten Daten.

7.3)

Schreiben Sie die Funktion `printZimmer`, die als Parameter eine Variable mit dem Datentyp Zimmer bekommt und die Länge, Breite und Höhe des Zimmers auf dem Bildschirm ausgibt.

7.4)

Schreiben Sie die Funktion `setZimmer` mit den folgenden 4 Parametern:
eine Variable mit dem Datentyp Zimmer, die zu setzende Länge, die zu setzende Breite und die zu setzende Höhe.

7.5)

Schreiben Sie die Funktion `printPerson`, die als Parameter eine Variable mit dem Datentyp Person bekommt und die Personalnummer und den Nachnamen der Person auf dem Bildschirm ausgibt.

7.6)

Schreiben Sie die Funktion `setPerson` mit den folgenden 3 Parametern:
eine Variable mit dem Datentyp Person, die zu setzende Personalnummer und den zu setzenden Nachnamen.

7.7)

Schreiben Sie die Funktion `printWohnung`, die als Parameter eine Variable mit dem Datentyp Wohnung bekommt und den Wohnungsbesitzer (speziell: Personalnummer und Nachnamen), den Wohnungsmieter (speziell: Personalnummer und Nachnamen) und die Zimmer (speziell: Länge, Breite und Höhe) in der Wohnung auf dem Bildschirm ausgibt.

7.8)

Schreiben Sie die Funktion `setWohnung` mit den folgenden 2 Parametern:
eine Variable mit dem Datentyp Wohnung (und den sich dort befindlichen Daten) und eine Variable mit dem Datentyp Wohnung, die eine Kopie davon ist.

7.9)

In einem Verwaltungsprogramm soll ein Verwaltungsbeamter über Tastatur die Anzahl der Wohnungen in einem Stadtteil und von diesen Wohnungen jeweils der Wohnungsbesitzer (speziell: Personalnummer und Nachnamen), der Mieter (speziell: Personalnummer und Nachnamen), die Zimmer (speziell: Länge, Breite, Höhe) eingeben werden können. Realisieren Sie das C-Programm.

7.10)

Die in dem vorigen Verwaltungsprogramm eingegebenen Daten sollen dann für jede Wohnung auf dem Bildschirm ausgegeben werden. Benutzen Sie dazu u.a. die Funktion `printWohnung`.

7.11)

Durch eine "staatliche Förderung" werden alle Wohnungsmieter zu Wohnungsbesitzern. Benutzen Sie dazu die Funktion `setWohnung`.

7.12)

Geben Sie mit Hilfe der Funktion `printWohnung` die Daten aller Wohnungen auf dem Bildschirm aus.

C-ÜBUNGSAUFGABEN STRUKTUREN + ZEICHNEN

Bemerkung:

a) Benutzen Sie die IGEL-Grafik.

b) Erstellen Sie für jede der folgenden Aufgaben ein eigenes Projekt, oder besser: packen Sie jede Aufgabe jeweils insgesamt in eine Funktion wie z.B. my_zeichnung1, my_zeichnung2, my_zeichnung3, usw.

1) Zeichnen Sie innerhalb eines Grafikfensters ein Männchen, dessen Beine und Hände aus Linien, dessen Kopf aus einem Kreis und dessen Rumpf aus einem Rechteck besteht:

Kopf :	Mittelpkt: 200, 50 Radius: 30	Arm2 :	Anfang: 230, 80, Ende: 280, 130
Rumpf :	links oben: 170, 80, rechts unten: 230, 180	Bein1 :	Anfang: 170, 180 Ende: 170, 250
Arm1 :	Anfang: 170, 80 Ende: 120, 30	Bein2 :	Anfang: 230, 180 Ende: 230, 250

2) Deklarieren Sie die Strukturen "punkt", "kreis", "rechteck", "linie":

Der Mittelpunkt des Kreises, die Begrenzungspunkte links oben bzw. rechts unten des Rechtecks und der Anfangs- und Endpunkt der Linie sollen jeweils durch die Struktur "punkt" realisiert werden. Der Anfangs- und der Endpunkt der Linie sollen sich in einem Feld der Länge 2 befinden.

Die Randfarbe und Füllfarbe sollen auch jeweils als Daten in der Struktur enthalten sein.

Erstellen Sie die Funktionen "zeichneKreis", "zeichneRechteck", "zeichneLinie", wobei der jeweilige Parameter die entsprechende Struktur enthält.

Zeichnen Sie das Männchen mit Hilfe dieser Funktionen.

3) Verbessern (übersichtlichere Gestaltung) Sie das Programm, indem Sie die Funktion "zeichneMann" implementieren.

4) Erstellen Sie die Funktionen "bewegeKreis", "bewegeRechteck", "bewegeLinie", die jeweils das entsprechende Objekt um den Wert dx in x-Richtung und den Wert dy in y-Richtung verschiebt.

Beispiel:

bewegeKreis(kr, 20, 30) verschiebt den Kreismittelpunkt (und damit den Kreis) um dx = 20 in x-Richtung und dy = 30 in y-Richtung.

5) Das Männchen soll sich von der Mitte des Grafikfensters aus zufällig torkelnd

(Fachsprache: "random walk", umgangssprachlich "Schlagseite") mit der Wahrscheinlichkeit $p=0,5$ nach links oder rechts und gleichzeitig mit der Wahrscheinlichkeit $p=0,5$ nach oben oder unten jeweils um eine Längeneinheit fortbewegen. Die Bewegung des Männchens soll als "Spur" auf dem Bildschirm dargestellt werden, d.h. das noch nicht verschobene Männchen soll auch noch auf dem Bildschirm erscheinen (umgangssprachlich: "das Männchen legt eine Spur").

Erstellen Sie dazu zuerst eine Funktion, die bestimmt, um wieviel Längeneinheiten (1 oder -1) das Männchen torkelt.

6) Verbessern (übersichtlichere Gestaltung) Sie das Programm, indem Sie die Struktur "mann" deklarieren, die Funktion "bewegeMann" implementieren und obige Funktion "zeichneMann" so abändern, dass sie eine Strukturvariable vom Typ "mann" als Parameter hat.

ZUSATZAUFGABEN

- 1) Es soll noch berücksichtigt werden, dass das Männchen nicht ausserhalb, sondern nur innerhalb des Grafikfensters bewegen soll.
- 2) Nur das verschobene Männchen (nicht auch noch seine "Spur") soll auf dem Bildschirm erscheinen.
- 3) Das Männchen soll immer wieder mit zufälligen Grössen, zufälligen Farben an zufällige Stellen des Grafikfensters gezeichnet werden.
- 4) Das Männchen soll bei jeder Verschiebung (Bewegung) zur besseren Beobachtung (zufällig oder systematisch) seine Farbe ändern.
- 5) Jedesmal, wenn das Männchen an den Fensterrand kommt, soll es seine Farbe ändern.
- 6) Entwickeln Sie zu der torkelnden Fortbewegungsmethode noch eine "gerichtete" torkelnde Fortbewegungsmethode: Der Punkt bewegt sich so lange (mit einer bestimmten Wahrscheinlichkeit) in eine bestimmte Richtung (z.B. x-Richtung: nach rechts, y-Richtung: nach unten) bis er an einen Fensterrand kommt. Dann ändert er seine Richtung: Wenn er z.B. am unteren Fensterrand ankommt, ändert er die y-Richtung nach oben.
- 7) Erweitern Sie die Funktion so, dass die Längeneinheit, um die das Männchen verschoben wird, nicht nur 1 oder -1 sein kann, sondern allgemein n , $-n$.
In einer anderen Fassung kann man die Funktion auch noch so erweitern, dass die Längeneinheit nicht nur 1 oder -1 sein kann, sondern allgemein irgendein Wert zwischen 1 und n bzw. der negative Wert davon.

Überlegen Sie sich selbst noch weitere Features !!