

Hochschule Deggendorf Dr. Peter Jüttner	
Vorlesung: Objektorientierte Programmierung	SS 2019
Übung 7	Termin 9.5.19

Klassen, Aufruf Oberklassenkonstruktor, Mehrfachvererbung

1. Memberinitialisierung

Auf dem Laufwerk finden Sie ein Programm(fragment) Übung 7-1.cpp mit zwei Klassen Fahrzeug und Kfz_Versicherung. Leider lässt sich das Programm nicht übersetzen. Finden Sie heraus, warum das so ist und korrigieren Sie den Fehler durch eine Memberinitialisierung

Der Konstruktor der Klasse Kfz_Versicherung wird ergänzt um eine Memberinitialisierung, die den Kontruktor der Klasse Fahrzeug aufruft. Dabei wird das Attribut V_Fahrzeug initialisiert.

```
Kfz_Versicherung(char Hersteller[H_LAENGE], char Modell[M_LAENGE],
unsigned int Leistung, unsigned int Nr, char Kd[K_LAENGE])
:V_Fahrzeug(Hersteller, Modell, Leistung)
{
    Nummer = Nr;
    if (strlen(Kd) < K_LAENGE)
        strcpy(Kunde,Kd);
    else printf("Falsche Kundenangabe\n");
};
```

2. Aufruf des Oberklassenkonstruktors in der Unterklasse

- Ändern Sie die Klasse Kfz dahingehend ab, dass alle Attribute private sind und deren Werte im Konstruktor per Parameter übergeben und gesetzt werden.
- Entfernen Sie die Methoden zum Berechnen der Steuer.
- Schreiben Sie eine Methode zum Ausgeben der Attribute eines Kfz (sofern nicht schon vorhanden)
- Erstellen Sie in einer der Unterklassen Wohnmobil, Pkw und Lkw einen entsprechenden Konstruktor, der den Oberklassenkonstruktor mit Parametern aufruft.
- Entfernen Sie die anderen Unterklassen.

```

#include <stdio.h>
#include <stdlib.h>
#include <string.h>

const unsigned int pauschale_Womo_Steuer = 500;

class Kfz
{ private:
    unsigned int Leistung;
    unsigned int Hubraum;
    unsigned int Gewicht;
    char Kennzeichen[11]; /* 1-3 Buchstaben, Bindestrich, 1-2 Buchstaben, 1-4
Ziffern */
    char Halter[20];

public:
    Kfz(unsigned int l, unsigned int h, unsigned int g, char k[11], char ha[20])
    { Leistung = l;
      Hubraum = h;
      Gewicht = g;
      strcpy(Kennzeichen,k);
      strcpy(Halter,ha);
    };

    void schreibekfz()
    { printf("Ausgabe eines Kfzs\n");
      printf("%s %s\n", "Kennzeichen:", Kennzeichen);
      printf("%s %s\n", "Halter:", Halter);
      printf("%s %d\n", "Leistung:", Leistung);
      printf("%s %d\n", "Hubraum:", Hubraum);
      printf("%s %d\n", "Gewicht:", Gewicht);
    };
};

class Pkw : public Kfz
{
public:
    Pkw(unsigned int l, unsigned int h, unsigned int g, char k[11], char ha[20]) :
    Kfz(l,h,g,k,ha)
    {};
};

int main()
{ // Kfz F1; /* Objekt einer rein virtuellen Klasse laesst sich nicht vereinbaren
*/
    unsigned int le;
    unsigned int hu;

```

```

unsigned int ge;
char ke[11];
char hal[20];

printf("Bitte Leistung eingeben:");
scanf("%d",&le);
printf("Bitte Hubraum eingeben:");
scanf("%d",&hu);
printf("Bitte zul Gesamtgewicht in kg eingeben:");
scanf("%d",&ge);
printf("Bitte Kennzeichen eingeben:");
scanf("%s", ke);
printf("Bitte Halter eingeben:");
scanf("%s", hal);

Pkw p(le, hu, ge, ke, hal);
p.schreibekfz();

system("PAUSE");
};

```

3. Aufgabe: Mehrfachvererbung

- a.) Übernehmen Sie die Klasse Kfz aus der vorherigen Übung, entfernen Sie die rein virtuelle Methode und führen Sie ein neues unsigned int Attribut Anzahl_Sitzplätze ein. Schreiben Sie, sofern noch nicht vorhanden einen Konstruktor, der alle Attribute einliest.

```

class Kfz
{ protected:
    unsigned int Leistung;
    unsigned int Hubraum;
    unsigned int Gewicht;
    char Kennzeichen[11]; /* 1-3 Buchstaben, Bindestrich, 1-2 Buchstaben, 1-4
Ziffern */
    char Halter[20];
    unsigned int Anzahl_Sitzplaetze;

public:
    Kfz()
    { printf("Einlesen eines Kfzs\n");
      printf("Bitte Leistung eingeben:");
      scanf("%d",&Leistung);
      printf("Bitte Hubraum eingeben:");
      scanf("%d",&Hubraum);
      printf("Bitte zul Gesamtgewicht in kg eingeben:");
      scanf("%d",&Gewicht);
      printf("Bitte Kennzeichen eingeben:");
      scanf("%s", Kennzeichen);
      printf("Bitte Halter eingeben:");

```

```

scanf("%s", Halter);
printf("Bitte Anzahl der Sitzplaetze eingeben:");
scanf("%s", &Anzahl_Sitzplaetze);
};
void schreibekfz()
{ printf("Ausgabe eines Kfzs\n");
  printf("%s %s\n", "Kennzeichen:", Kennzeichen);
  printf("%s %s\n", "Halter:", Halter);
  printf("%s %d\n", "Leistung:", Leistung);
  printf("%s %d\n", "Hubraum:", Hubraum);
  printf("%s %d\n", "Gewicht:", Gewicht);
  printf("%s %d\n", "Sitzplaetze:", Anzahl_Sitzplaetze);
};
};

```

- b.) Definieren Sie eine Klasse Verkehrsmittel mit den Attributen unsigned int Anzahl_Sitzplätze, Einsatzbereich (Aufzählungstyp mit den Elementen Stadt, Land, Stadt_und_Land, Kontinental, Land_und_Kontinental, Interkontinental, Kontinental_und_Interkontinental, Land_und_Kontinental_und_Interkontinental), (kürzen Sie geeignet ab)
Schreiben Sie einen Konstruktor, der alle Attribute einliest.

```

enum Einsatz {S=1, L=2, S_u_L=3, K=4, L_u_K=5, I=6, K_u_I=7,
L_u_K_u_I=8};

```

```

class Verkehrsmittel
{ protected:
  Einsatz Einsatzgebiet;
  unsigned int Anzahl_Sitzplaetze;

public:
  Verkehrsmittel()
  { printf("Bitte Einsatzgebiet für Verkehrsmittel eingeben (1, .. , 8)\n");
    scanf("%d",&Einsatzgebiet);
    printf("Bitte Anzahl der Sitzplaetze für Verkehrsmittel eingeben\n");
    scanf("%d",&Anzahl_Sitzplaetze);
  };

  void schreibeverkehrsmittel()
  { printf("Ausgabe Verkehrsmittel:\n");
    printf("Einsatzgebiet:");
    switch(Einsatzgebiet)
    { case 1: printf("Stadt");
      break;
      case 2: printf("Land");
      break;
      case 3: printf("Stadt und Land");
      break;

```

```

        case 4: printf("Kontinental");
                break;
        case 5: printf("Land und Kontinental");
                break;
        case 6: printf("Interkontinental");
                break;
        case 7: printf("Kontinental und Interkontinental");
                break;
        case 8: printf("Land und Kontinental und Interkontinental");
                break;
        default: printf("Unbekanntes Einsatzgebiet");
    };
    printf("\n");
    printf("%s%d\n", "Anzahl der Sitzplaetze:", Anzahl_Sitzplaetze);
};
};

```

- c.) Definieren Sie eine Klasse `Autobus` als Unterklasse der Klassen `Kfz` und `Verkehrsmittel`. Vereinbaren Sie im Hauptprogramm ein Objekt vom Typ `Autobus`. Was passiert, wenn Sie das Programm compilieren und ausführen?

```

class Autobus : public Kfz, public Verkehrsmittel
{ public:
    Autobus() // : Kfz(), Verkehrsmittel()
    { printf("Konstruktor von Autobus aufgerufen\n");
    };
};

```

Der Konstruktor der Klasse `Autobus` ruft implizit die Konstruktoren der Oberklassen. So werden die Sitzplätze zweimal eingelesen.

- d.) Definieren Sie in der Klasse `Autobus` eine Methode, die alle Attribute (inkl. der ererbten) ausgibt.

```

void Ausgabe_Autobus()
{ printf("Ausgabe eines Autobus\n");
  printf("%s %s\n", "Kennzeichen:", Kennzeichen);
  printf("%s %s\n", "Halter:", Halter);
  printf("%s %d\n", "Leistung:", Leistung);
  printf("%s %d\n", "Hubraum:", Hubraum);
  printf("%s %d\n", "Gewicht:", Gewicht);
  printf("%s %d\n", "Sitzplaetze:", Kfz::Anzahl_Sitzplaetze);
  printf("Einsatzgebiet:");
  switch(Einsatzgebiet)
  { case 1: printf("Stadt");
        break;
    case 2: printf("Land");
        break;
    case 3: printf("Stadt und Land");
  };
};

```

```
        break;
    case 4: printf("Kontinental");
        break;
    case 5: printf("Land und Kontinental");
        break;
    case 6: printf("Interkontinental");
        break;
    case 7: printf("Kontinental und Interkontinental");
        break;
    case 8: printf("Land und Kontinental und Interkontinental");
        break;
    default: printf("Unbekanntes Einsatzgebiet");
};
printf("\n");
printf("%s %d\n", "Sitzplaetze:", Verkehrsmittel::Anzahl_Sitzplaetze);
};
};
```