

## Objekte

### Bewertungsaufgabe p03c

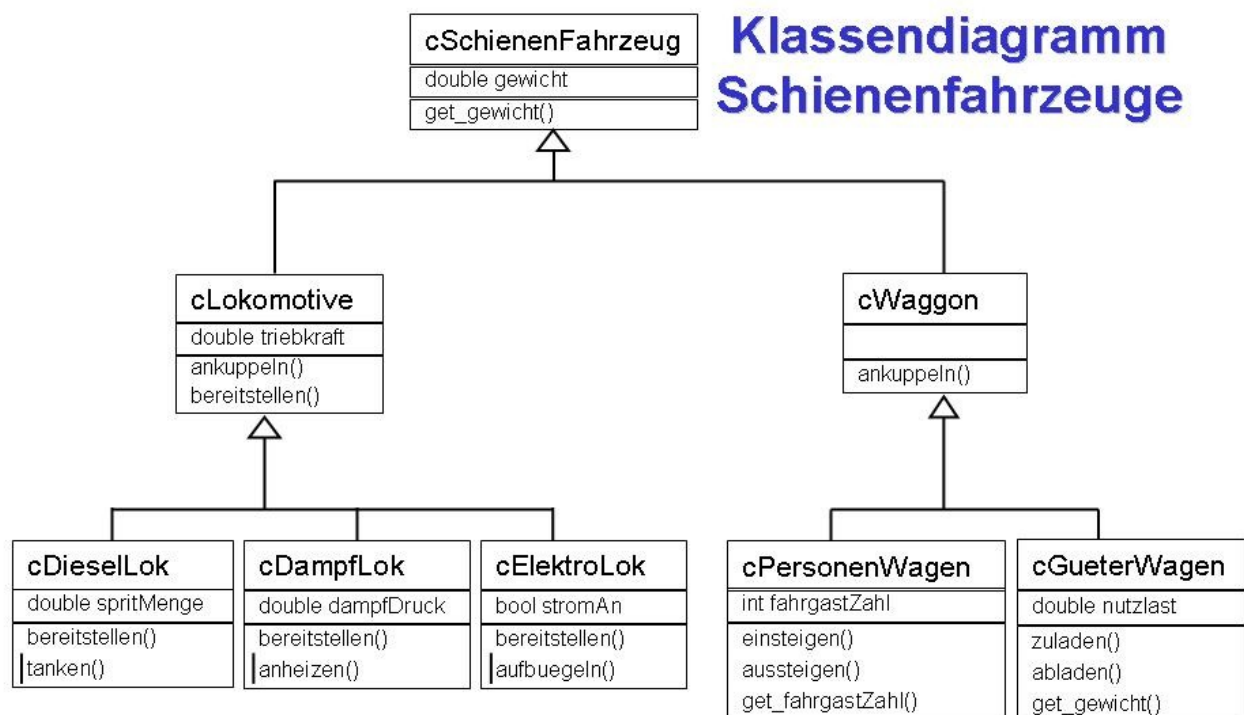
### Anlage mehrerer Klassenartefakte im MS Visual Studio

Lernziele: Klassenhierarchie, Vererbung, Überschreibung von Methoden, systematisches Vorgehen, Mehr-Dateien-Projekt in MS Visual Studio



Quelle: Trains-on-Want

Setzen Sie folgendes Klassendiagramm in ein C++-Programm um:



- Gehen Sie systematisch vor.
- Lesen Sie die Aufgabe genau durch.
- Arbeiten Sie schrittweise
- **Legen Sie für jede Klasse ein Pärchen aus Header- und Klassen-Quelldatei an (\*.cpp), und für das Hauptprogramm eine Quelldatei schienenfahrzeuge.cpp**

### Motivation:

Sie betreiben eine Eisenbahngesellschaft mit unterschiedlichen Lokomotiven- und Waggontypen.

Die Lokomotiven werden vor ihrem Einsatz bereitgestellt, dann können Waggon angehängt werden.

Jeder Lokomotiventyp hat seine spezielle („spezialisierte“) Version (Überschreibung) der Bereitstellungsmethode:

Bei der Diesellok wird zunächst getankt. Von der Tankmenge wird danach die Einsatzdauer in Stunden bestimmt.

Bei der Dampflok wird zunächst der Kessel angeheizt und der Druck aufgebaut. Vom aufgebauten Druck ist die Einsatzdauer abhängig.

Bei der Elektrolok wird zunächst der Stromabnehmer ausgefahren (es wird „aufgebügelt“), damit die Lok Strom bekommt. Die Einsatzdauer kann pauschal verdoppelt werden, der Strom kommt ja aus der Oberleitung. Allerdings ist der Einsatz davon abhängig, dass der Strom auch angeschaltet ist.

Dann kann ein Zug zusammengestellt werden, eine Lok und (mehrere) Waggon können angekuppelt werden.

In Personenwagen können Fahrgäste ein- und aussteigen, in Güterwagen können Lasten zugeladen und abgeladen werden.

Dann mal ran, und immer schön pünktlich fahren! Viel Spaß.

### Die Programmieraufgabe:

- Legen Sie alle Klassendefinitionen leer an.
- Stellen Sie die Ableitungen her (Vererbungen).
- Definieren Sie die Mitgliedsvariablen.
- Bauen Sie die Standard-Konstruktoren, die alle Werte auf Null setzen, außer gewicht=50000.0 und triebkraft=6500.0 in den betreffenden Klassen.
- Implementieren Sie die Methoden der Klassen.
- Erstellen Sie ein Hauptprogramm, das mit Instanzen der Klassen arbeitet. (Der Anweisungsteil wird vorgegeben, siehe unten).

### Die Klassen im Einzelnen:

#### **cSchienenfahrzeug:**

- public Methoden:  
double get\_gewicht(); // Liefert den Wert des Attributs  
// gewicht als Returncode zurueck

#### **cLokomotive:**

- public Methoden:  
void ankuppeln(); // Gibt aus "Ich zieh Euch alle!"  
  
double bereitstellen (int stunden); // Liefert den Rückgabewert:  
// gewicht + (stunden \* triebkraft)

#### **cWaggon:**

- public Methoden:  
void ankuppeln(); // Gibt aus: "Ich lasse mich ziehen"

### **cDieselLok:**

- public Methoden:  
double bereitstellen (int stunden); // Erhöht den Wert von spritMenge  
// um das Ergebnis von tanken(), und ruft  
// dann bereitstellen() der Basisklasse auf  
// mit dem Parameterwert  
// maximum (stunden, spritMenge / 200)
- private Methoden:  
double tanken(); // Gibt die Tankmenge 2000.0 zurueck

### **cDampfLok:**

- public Methoden:  
double bereitstellen (int stunden); // Erhöht den Wert von dampfDruck  
// um das Ergebnis von anheizen(), und ruft  
// dann bereitstellen() der Basisklasse auf  
// mit dem Parameterwert  
// stunden - dampfDruck / 10
- private Methoden:  
double anheizen(); // Gibt den Wert 85.3 zurück

### **cElektroLok:**

- public Methoden:  
double bereitstellen (int stunden); // Ruft aufbuegeln() auf und ruft  
// dann bereitstellen() der Basisklasse auf  
// mit dem Parameterwert  
// stunden \* 2 \* (int)stromAn
- private Methoden:  
void aufbuegeln(); // setzt das Attribut stromAn auf true

### **cPersonenwagen:**

- public Methoden:  
int einsteigen (int rein); // Erhoeht die Anzahl der Fahrgaeste;  
// Begrenzung auf <= 117;  
// gibt die neue FahrgastZahl zurueck  
  
int aussteigen (int raus); // Vermindert die Zahl der Fahrgaeste;  
// Begrenzung auf >=0;  
// gibt die neue FahrgastZahl zurueck  
  
int get\_fahrgastZahl(); // Liefert den Wert FahrgastZahl  
zurueck

### **cGueterwagen:**

- public Methode:  
double zuladen(double lastplus); // lastplus wird auf nutzlast addiert,  
// Begrenzung auf <=200000.0,  
// gibt neuen Wert der nutzlast zurück  
  
double abladen(double lastweg); // lastweg wird von nutzlast abgezogen,  
// Begrenzung auf >=0.0  
// gibt neuen Wert der nutzlast zurück  
  
double get\_gewicht(); // Ueberschreibt get\_gewicht() des  
// Schienenfahrzeugs, liefert den Wert  
// (gewicht + nutzlast) zurueck

### Das Hauptprogramm:

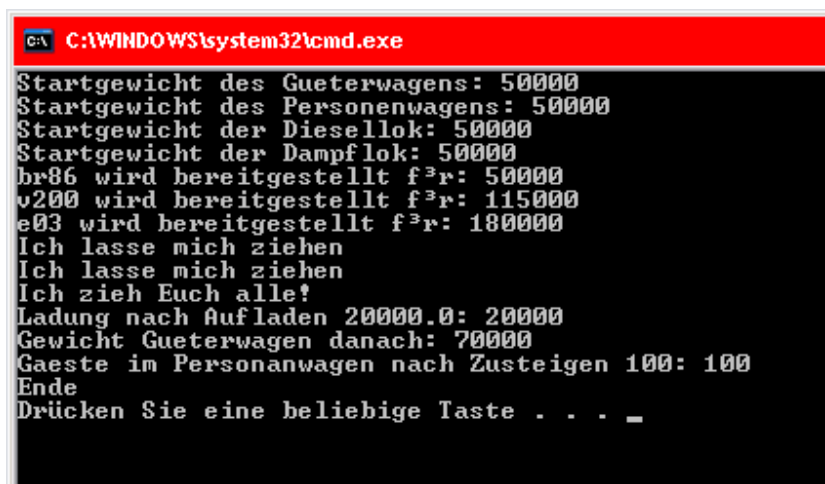
- Erstellen Sie folgende Instanzen:

```
cGueterWagen packdrauf;  
cPersonenWagen pullman;  
cDieselLok v200;  
cDampfLok br86;  
cElektroLok e03;
```

- Verwenden Sie folgende Anweisungen, um die Funktion Ihrer Klassen und Methoden nachzuweisen (verwenden Sie cut and paste):

```
-----  
// Mit den Objekten arbeiten:  
  
cout << "Startgewicht des Gueterwagens: " << packdrauf.get_gewicht() << endl;  
cout << "Startgewicht des Personenwagens: " << pullman.get_gewicht() << endl;  
cout << "Startgewicht der Diesellok: " << v200.get_gewicht() << endl;  
cout << "Startgewicht der Dampflokomotive: " << br86.get_gewicht() << endl;  
  
cout << "br86 wird bereitgestellt: " << br86.bereitstellen(8) << endl;  
cout << "v200 wird bereitgestellt: " << v200.bereitstellen(6) << endl;  
cout << "e03 wird bereitgestellt: " << e03.bereitstellen(10) << endl;  
  
pullman.ankuppeln();  
packdrauf.ankuppeln();  
v200.ankuppeln();  
  
cout << "Ladung nach Aufladen 20000.0: " << packdrauf.zuladen(20000.0) << endl;  
cout << "Gewicht Gueterwagen danach: " << packdrauf.get_gewicht() << endl;  
cout << "Gäste im Personenwagen nach Zusteigen 100: " <<  
pullman.einsteigen(100) << endl;  
  
cout << "Ende" << endl;  
-----
```

- Die Programmausgabe sollte dann wie folgt aussehen:



```
C:\WINDOWS\system32\cmd.exe  
Startgewicht des Gueterwagens: 50000  
Startgewicht des Personenwagens: 50000  
Startgewicht der Diesellok: 50000  
Startgewicht der Dampflokomotive: 50000  
br86 wird bereitgestellt f³r: 50000  
v200 wird bereitgestellt f³r: 115000  
e03 wird bereitgestellt f³r: 180000  
Ich lasse mich ziehen  
Ich lasse mich ziehen  
Ich zieh Euch alle!  
Ladung nach Aufladen 20000.0: 20000  
Gewicht Gueterwagen danach: 70000  
Gäste im Personenwagen nach Zusteigen 100: 100  
Ende  
Drücken Sie eine beliebige Taste . . . _
```

- Natürlich können Sie beliebig Lokomotiven einsetzen, beliebig viele Waggon anhängen, Fahrgäste ein- und aussteigen lassen, Lasten zu- und abladen, Varianten der Anweisungen im Hauptprogramm durchspielen, wie Sie möchten. viel Spaß dabei!

## **Bewertungsschema:**

**Übersetzungsfehler ==> 0 P !**

sonst:

Kommentare und Codeformatierung	4 P
Klassen / Artefaktentrennung (Dateien)	6 P
Konstruktoren und -kaskade	4 P
Methodenimplementierungen	4 P
Hauptprogramm	2 P
	-----
Summe	20 P