

OPR Praktikum

Übung: Verkehrsmittel

Thomas Mahr

11. November 2019

1 Lernziele

- Vererbung benutzen
- Elementfunktionen überschreiben
- Klassendiagramm entwerfen

2 Voraussetzungen

Kapitel Vererbung

3 Beschreibung von Verkehrsmitteln

1. Ein Verkehrsmittel besitzt eine Position.
2. Ein Verkehrsmittel kann zu einer neuen Position bewegt werden.
3. Die Position besteht aus ganzzahligen x- und y-Werten.
4. Der Vorgabewert der Position ist (0,0).
5. Es gibt zwei spezielle Verkehrsmittel: Luftfahrzeuge und Radfahrzeuge.
6. Ein Luftfahrzeug besitzt eine maximal Flughöhe.
7. Ein Radfahrzeug besitzt eine bestimmte Anzahl Räder.
8. Ein Pkw ist ein spezielles Radfahrzeug.
9. Ein Pkw besitzt eine bestimmte Anzahl Türen.
10. Ein Heißluftballon ist ein spezielles Luftfahrzeuge.
11. Ein Heißluftballon besitzt ein bestimmtes Ballonvolumen.

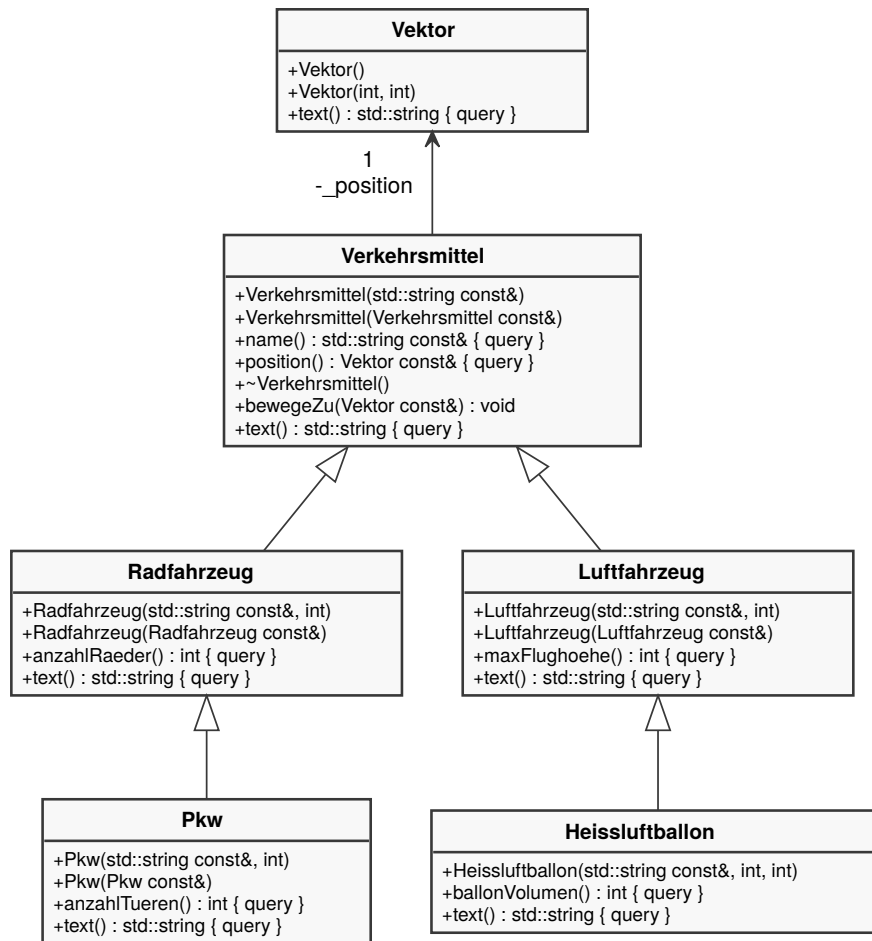


Abbildung 1: UML-Klassendiagramm

4 Aufgabe

Erstellen Sie die Abb. 1 gezeigten Klassen in jeweils einer eigenen Header-Datei und Übersetzungseinheit, so dass die vorgegebene `main()`-Funktion die unten gezeigte Ausgabe liefert.

```

1  #include "Pkw.h"
2  #include "Heissluftballon.h"
3  #include <iostream>
4  #include <cassert>
5  using namespace std;
6
7  void test01() {
8      cout << "test01:\n-----\n";
9      const Verkehrsmittel verkehrsmittel("Verkehrsmittel");
10     const Radfahrzeug radfahrzeug("Radfahrzeug", 3);
11     const Pkw pkw("VW Käfer", 2);
12     const Luftfahrzeug luftfahrzeug("Luftfahrzeug", 10000);
13     const Heissluftballon heissluftballon("Heissluftballon", 500,
        7500);
  
```

```

14     cout << verkehrsmittel.text() << endl;
15     cout << radfahrzeug.text() << endl;
16     cout << pkw.text() << endl;
17     cout << luftfahrzeug.text() << endl;
18     cout << heissluftballon.text() << endl;
19     assert(radfahrzeug.anzahlRaeder()==3);
20     assert(pkw.name()=="VW Käfer");
21     assert(pkw.anzahlRaeder()==4);
22     assert(pkw.anzahlTueren()==2);
23     assert(heissluftballon.maxFlughoehe()==500);
24     assert(heissluftballon.ballonVolumen()==7500);
25 }
26
27 void test02() {
28     cout << "test02:\n-----\n";
29     Pkw pkw("Pkw", 2);
30     cout << pkw.text() << endl;
31     pkw.bewegeZu(Vektor(50, 60));
32     cout << pkw.text() << endl;
33     cout << pkw.name() << " befindet sich nach der Bewegung an
        Position " << pkw.position().text() << endl;
34     assert(pkw.position().x==50);
35     assert(pkw.position().y==60);
36     Heissluftballon heissluftballon("Heissluftballon", 1000, 7500);
37     cout << heissluftballon.text() << endl;
38     heissluftballon.bewegeZu(Vektor(70, 80));
39     cout << heissluftballon.name() << " befindet sich nach der
        Bewegung an Position " << heissluftballon.position().text()
        << endl;
40     assert(heissluftballon.position().x==70);
41     assert(heissluftballon.position().y==80);
42 }
43
44 void test03() {
45     cout << "test03:\n-----\n";
46     Pkw pkw1("Pkw1", 5);
47     pkw1.bewegeZu(Vektor(77, 88));
48     cout << pkw1.text() << endl;
49     const Pkw pkw2 = pkw1;
50     cout << pkw2.text() << endl;
51     assert(pkw1.anzahlRaeder()==pkw2.anzahlRaeder());
52     assert(pkw1.anzahlTueren()==pkw2.anzahlTueren());
53     assert(pkw1.position().x==pkw2.position().x);
54     assert(pkw1.position().y==pkw2.position().y);
55     assert(pkw1.name()!=pkw2.name());
56 }
57
58 void test04() {
59     cout << "test04:\n-----\n";
60     Heissluftballon ballon1("Ballon1", 234, 5678);
61     ballon1.bewegeZu(Vektor(99, 111));
62     cout << ballon1.text() << endl;
63     const Heissluftballon ballon2{ballon1};
64     cout << ballon2.text() << endl;

```

```

65     assert(ballon1.maxFlughoehe()==ballon2.maxFlughoehe());
66     assert(ballon1.ballonVolumen()==ballon2.ballonVolumen());
67     assert(ballon1.position().x==ballon2.position().x);
68     assert(ballon1.position().y==ballon2.position().y);
69     assert(ballon1.name()!=ballon2.name());
70 }
71
72 int main() {
73     test01();
74     cout << endl;
75     test02();
76     cout << endl;
77     test03();
78     cout << endl;
79     test04();
80 }

```

Listing 1: main.cpp

Ausgabe:

test01:

```

Verkehrsmittel: Position=(0,0)
Radfahrzeug: Position=(0,0) Räder=3
VW Käfer: Position=(0,0) Räder=4 Türen=2
Luftfahrzeug: Position=(0,0), maxFlughoehe=10000m
Heissluftballon: Position=(0,0), maxFlughoehe=500m, ballonVolumen=7500m^3
Heissluftballon gelöscht
Luftfahrzeug gelöscht
VW Käfer gelöscht
Radfahrzeug gelöscht
Verkehrsmittel gelöscht

```

test02:

```

Pkw: Position=(0,0) Räder=4 Türen=2
Pkw: Position=(50,60) Räder=4 Türen=2
Pkw befindet sich nach der Bewegung an Position (50,60)
Heissluftballon: Position=(0,0), maxFlughoehe=1000m, ballonVolumen=7500m^3
Heissluftballon befindet sich nach der Bewegung an Position (70,80)
Heissluftballon gelöscht
Pkw gelöscht

```

test03:

```

Pkw1: Position=(77,88) Räder=4 Türen=5
Kopiere Verkehrsmittel (Kopie von Pkw1)
Kopiere Radfahrzeug
Kopiere Pkw
Kopie von Pkw1: Position=(77,88) Räder=4 Türen=5
Kopie von Pkw1 gelöscht
Pkw1 gelöscht

```

test04:

Ballon1: Position=(99,111), maxFlughoehe=234m, ballonVolumen=5678m³

Kopiere Verkehrsmittel (Kopie von Ballon1)

Kopiere Luftfahrzeug

Kopie von Ballon1: Position=(99,111), maxFlughoehe=234m, ballonVolumen=5678m³

Kopie von Ballon1 gelöscht

Ballon1 gelöscht

Hinweise:

1. Zur verwendeten Funktion `assert()` siehe <https://en.cppreference.com/w/cpp/error/assert>.
2. Gehen Sie bei der Implementierung in kleinen Schritten vor und bringen Sie die Testfunktionen eine nach der anderen zum Laufen.