Programmierkurs
Dr. Steffen Müthing
IWR, Universität Heidelberg

Probeklausur
Abgabe 08. Februar 2019
ipk-exercises:2018-ws-52-g208e25b

Probeklausur

Allgemeine Hinweise:

- Dieses Übungsblatt wird **nicht** bewertet.
- Die Aufgaben auf diesem Blatt sollen Ihnen einen Eindruck von Umfang und Schwierigkeit einzelner Klausuraufgaben geben, die Klausur selbst wird aber mehr Aufgaben enthalten.
- Wir werden eine Musterlösung zu diesem Blatt nach der Vorlesung am 06.02. online stellen.

Aufgabe 1

Geben Sie kurze Antworten auf die folgenden Fragen:

(a) Welche Sichtbarkeits-Arten gibt es in einer Klasse, und was bedeuten sie?

(b) Was sind Referenzen, und wofür nutzt man sie (Anwendungsbeispiel)?

(c) Worin unterscheiden sich `std::array` und `std::vector`?

(d) Was ist der Hauptgrund, Templates zu verwenden, d.h. was ist ihre wichtigste Aufgabe?

(e) Welche Regeln sollten beim Nutzen von virtuellen Methoden immer eingehalten werden?

(f) Welchen grossen Vorteil bietet die generische Programmierung (Templates), und welche Nachteile?

Aufgabe 2

Lesen Sie sich folgendes Programm durch:

```
#include <iostream>

// changes the vector v in place by multiplying all its entries by n
// does not return anything
void multiply(std::vector<int> v, int n)
{
  for (auto& i : v)
    i = i + n;
}

int main(int argc, char** argv)
{
  vector<int> v = {{ 1, 2, 3, 4, 5 }};
  v = multiply(v,3);
  // output all entries in the vector together with their index
  for (int i = 1 , i <= v.size() , i++)
    std::cout << i << ": ";
    std::cout << v[i] << std::endl;
  return 0
}
```

Dieses Programm enthält insgesamt 10 Fehler, von denen manche zu Compile-Fehlern führen und andere zu Laufzeit-Fehlern, bei denen das Programm sich nicht so verhält wie erwünscht.

Finden Sie alle 10 Fehler (Sie können die Fehler im Quellcode markieren) und benennen Sie kurz (wenige Worte), wo der Fehler liegt. Schreiben Sie ausserdem dazu, ob der Fehler zur Compile- oder zur Laufzeit auftritt.

Aufgabe 3

Gegeben sei folgende Klasse (nicht vollständig!):

```
1 class Building
2 {
3 public:
4   std::string name() const;
5   std::string city() const;
6   int height() const;
7 };
```

Sie sollen folgende Funktion implementieren:

```
1 std::vector<Building> sortedByHeight(const std::vector<Building>& vec);
```

Diese Funktion soll eine Kopie des Vektors zurückgeben, in der die Gebäude absteigend nach Höhe sortiert sind.

Implementieren Sie die Funktion!

- Zum Sortieren verwenden Sie `std::sort()`.
- Schreiben Sie eine vollständige Funktion inklusive der oben angegebenen Funktions-Signatur, nicht nur den Funktions-Body.

Aufgabe 4

Schreiben Sie eine Klasse `Student`, die die drei Attribute Vorname, Nachname und Matrikelnummer speichert (die ersten beiden als String, das letzte als Integer). Die Variablen in der Klasse sollen gekapselt und nicht extern zugänglich sein. Beachten Sie folgende Punkte:

- Einen Konstruktor, der Werte für die Attribute als Parameter akzeptiert und in der Klasse speichert.
- Öffentliche Zugriffsmethoden für Lese-Zugriff, die die Namen `firstName()`, `lastName()` und `idNumber()` tragen sollen. Methoden für Schreibzugriff sind nicht erforderlich.
- Es soll möglich sein, ein Objekt vom Typ `Student` auf das Terminal auszugeben:

```
1 Student s("Max","Mustermann",321542);
2 std::cout << s << std::endl;
```

  Dies soll folgende Ausgabe auf dem Terminal produzieren:

```
Mustermann, Max: 321542
```

- Beachten Sie die Richtlinien für Klassendesign aus der Vorlesung.
- Denken Sie an eventuell erforderliche includes!