

Aufgabenblatt 11

Hinweise zur Klausur:

- Die Klausur wurde um einen Tag verschoben und findet jetzt am **20.02.2018 um 14:00 im grossen Hörsaal der Chemie** statt.

Melden Sie sich bitte bis zum **12.02.2018** im MÜSLI für die Klausur an. Danach ist der Menüpunkt gesperrt und weder eine An- noch Abmeldung möglich.

- Melden Sie sich bitte vor dem Ende der Anmeldefrist bei uns, wenn bei Ihnen noch Punkte nachgetragen werden müssen.
- Die Klausureinsicht findet am 23.02. von 14:00 bis 15:00 Uhr im Raum 1/414 im Mathematikon statt.
- Sie müssen für die Klausur angemeldet sein und teilgenommen haben, um an der Nachprüfung teilnehmen zu können. Sollten Sie am Klausurtermin krank sein, schicken Sie uns bitte eine kurze Email und reichen baldmöglichst ein ärztliches Attest nach.
- Eventuelle Nachprüfungen werden in der Woche vom 09.04.2018 stattfinden.

Allgemeine Hinweise:

- Dieses Übungsblatt wird **nicht** bewertet.
 - Die Aufgaben auf diesem Blatt sollen Ihnen einen Eindruck von Umfang und Schwierigkeit einzelner Klausuraufgaben geben.
 - Bei Fragen zu einzelnen Aufgaben können Sie in die Übungen am 12.02. und 16.02. kommen.
 - Wir werden eine Musterlösung zu diesem Blatt nach der Übung am 16.02. online stellen.
-

Aufgabe 1

Geben Sie kurze Antworten auf die folgenden Fragen:

- (a) Welche Kontrollstrukturen kennt C++?

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

(c) Was sind die zentralen Unterschiede zwischen `std::vector` und `std::map` (konzeptuell und in der Nutzung)?

(d) Was ist der Hauptgrund, Klassen zu definieren, d.h. was ist ihre Aufgabe?

(e) Welche Bedingung sollte beim Nutzen von Vererbung immer eingehalten werden?

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

Aufgabe 2

Lesen Sie sich folgendes Programm durch:

```

1  #include <iostream>
2
3  // changes the vector v in place by multiplying all its entries by n
4  // does not return anything
5  void multiply(std::vector<int> v, int n)
6  {
7      for (auto& i : v)
8          i = i + n;
9  }
10
11 int main(int argc, char** argv)
12 {
13     vector<int> v = {{ 1, 2, 3, 4, 5 }};
14     v = multiply(v,3);
15     // output all entries in the vector together with their index
16     for (int i = 1 , i <= v.size() , i++)
17         std::cout << i << ": ";
18         std::cout << v[i] << std::endl;
19     return 0
20 }
```

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 Funktions-Signatur:

```
1 int longestString(const std::vector<std::string>& vec);
```

Diese Funktion soll den längsten String in dem übergebenen `vector` finden und die Länge dieses Strings ausgeben.

Implementieren Sie die Funktion!

- Um die Länge eines Strings herauszufinden, können Sie die Member-Funktion `size()` der String-Klasse verwenden.
- 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.
- Beachten Sie die Richtlinien für Klassendesign aus der Vorlesung.

- Denken Sie an eventuell erforderliche includes!