

Übungsblatt 5

Hinweise

- Reichen Sie Ihre **Lösung fristgerecht** und **ausschließlich** über das **E-Learning** ein. Beachten Sie bitte die Abgaberegularien! Der Abgabetermin ist bei der Aufgabenstellung angegeben.
 - Unter U5 : Abgabe ist **eine PDF-Datei** (Aufgaben 1 bis 3) hochzuladen.
 - **VPL-Programmieraufgaben:** Jedes **Speichern** im VPL-Editor (mittels Disketten-Symbol) erzeugt **automatisch** eine aktualisierte **Abgabe** der jeweiligen Aufgabe. Es sind keine weiteren Aktionen nötig, insbesondere **NICHT** im Reiter **Abgabe**! **Nur** die zuletzt getätigte Abgabe wird bewertet!
-

Theorie

(8 Punkte)

Arbeiten Sie die Vorlesungsfolien *Dateiarbeit* sowie zur Wiederholung *Zeiger und Referenzen* und *OOP* durch und beantworten Sie folgende Fragen.

1. Dateiarbeit / Ein- und Ausgabestreams (2.5 P)

- a) Nennen Sie die 3 wichtigsten Stream-Klassen. Wofür werden sie jeweils verwendet?
- b) Welche 3 Schritte sind bei der Nutzung von Filestreams immer auszuführen?
- c) Welche Öffnungsmodi für Dateien gibt es? Was bedeuten Sie jeweils?
- d) Was bedeuten die Methoden `good()`, `bad()` und `fail()`?
- e) Nennen Sie Beispiele für *schwere* und für *korrigierbare* Fehler.

2. Operatoren `new` und `delete` (1.5 P):

- a) Was steht rechts vom `new`-Operator? Was muss in diesem Argument *immer* enthalten sein?
- b) Was liefert der `new`-Operator zurück?
- c) Für den `delete`-Operator gibt es zwei grundlegende Syntaxvarianten: `delete ptrVar1;` und `delete [] ptrVar2;`. Worin besteht der Unterschied?

3. OOP: dynamische Komponenten (4.0 P):

- a) Notieren Sie die zwei Codezeilen, mit denen (1) eine `double`-Variable dynamisch erzeugt und (2) anschließend deren Speicher wieder freigegeben wird.
- b) Notieren Sie die zwei Codezeilen, mit denen (1) ein `int`-Vektor mit 20 Elementen dynamisch erzeugt und (2) anschließend dessen Speicher wieder freigegeben wird.
- c) Notieren Sie die zwei Codezeilen, mit denen (1) ein Objekt der Klasse `MESSWERT` dynamisch erzeugt und (2) anschließend dessen Speicher wieder freigegeben wird.
- d) Notieren Sie die zwei Codezeilen, mit denen (1) ein Vektor mit 10 Elementen der Klasse `MESSWERT` dynamisch erzeugt und (2) anschließend dessen Speicher wieder freigegeben wird.

4. Messreihen mit elektrischen Messwerten: VPL-Programmieraufgabe (19 P)

Die Funktionalität des Programms wird durch zwei Klassen realisiert:

- Mit der Klasse `MESSWERT` werden elektrische Messwerte mit den Attributen `Zahlenwert` und `Einheit` in Objekten abgebildet.
- Mit der Klasse `MESSREIHE` werden Vektoren aus `MESSWERT`-Objekten gebildet. Die Länge einer solchen Messreihe ist beliebig, die Reihe wird dynamisch erzeugt.

Im E-Learning finden Sie die folgenden vorbereiteten Dateien:

- `Messungen.h` : Deklarationen aller Klassen (`MESSWERT` und `MESSREIHE`)
- `Messungen.cpp` : Definitionen aller Funktionen (`main` und ggf. Leistungsberechnung)
- `MessWert.cpp` : Implementierung der Klasse `MESSWERT`
- `MessReihe.cpp` : Implementierung der Klasse `MESSREIHE`

- a) Implementieren Sie in `MessWert.cpp` alle Methoden der Klasse `MESSWERT`. Sie können dafür Ihre Lösung aus Übungsblatt 4, Aufgabe 4 nutzen. Ändern Sie in der Klassendeklaration in `Messungen.h` den Namen der Normierungs-Methode in den von **Ihnen** gewählten.
- b) Definieren Sie in `Messungen.h` eine Klasse `MESSREIHE` mit den folgenden Attributen:
 - (1) Anzahl der Messwerte (in der jeweiligen Messreihe)
 - (2) Zeiger auf ein Objekt der Klasse `MESSWERT`

Definieren Sie die folgenden Methoden der Klasse `MESSREIHE` in `MessReihe.cpp`:

- Verwenden Sie aussagekräftige Namen für alle Methoden und Attribute.
- Prüfen Sie in allen Methoden, ob die übergebenen Parameter gültig sind.
- Schützen Sie in allen Methoden die übergebenen Parameter vor Modifikationen innerhalb der Methode (wo es möglich und sinnvoll ist).
- In keiner der Methoden darf `cout` verwendet werden! Alle Ausgaben erfolgen über die zu definierende Ausgabemethode (siehe i)).

c) Konstruktor

- Die Anzahl der Messwerte wird als Argument übergeben.
- Erzeugen Sie dynamisch einen Vektor mit der *Anzahl* von `MESSWERT`-Objekten.
- Das Zeiger-Attribut zeigt auf den Anfang (auf das 0. Element) von diesem Vektor.

d) Kopierkonstruktor

- Kopieren Sie die gesamte Messreihe!
- Achten Sie darauf, dass das Zeiger-Attribut des neuen Objektes auf einen **eigenen** Vektor zeigt und nicht auf den Vektor der Kopiervorlage!

- e) Destruktor
 - Achten Sie darauf, dynamisch erzeugte Komponenten löschen!
- f) Abfrage eines Messwertes an beliebiger Position in der Messreihe
 - Die Position (der Index) wird als Argument übergeben.
 - Der Messwert wird als `MESSWERT`-Objekt zurückgegeben.
- g) Setzen eines Messwertes an beliebiger Position in der Messreihe
 - Der Messwert und die Position (der Index) werden als Argumente übergeben.
 - Der Messwert wird als `MESSWERT`-Objekt übergeben.
- h) Abfrage der Anzahl der Messwerte in der Messreihe
- i) Ausgabe aller Messwerte auf dem Bildschirm
 - Rufen Sie die Ausgabe-Methoden der `MESSWERT`-Objekte auf.
- j) Bestimmung des kleinsten Messwertes aus der Messreihe
 - Zurückgegeben wird die Position (der Index) des Minimums, nicht das Objekt selbst!
- k) Bestimmung des größten Messwertes aus der Messreihe
 - Zurückgegeben wird die Position (der Index) des Maximums, nicht das Objekt selbst!

Realisieren Sie die folgenden Funktionalitäten in der `main`-Funktion in `Messungen.cpp` :

- l) Erzeugen Sie eine Messreihe mit 5 Elementen und belegen Sie die Elemente mit folgenden Messwerten: 10 mA, 20 mA, 30 mA, 40 mA, 50 mA.
- m) Erzeugen Sie mit dem Kopierkonstruktor eine Kopie dieser Messreihe. Skalieren Sie in dieser Kopie alle Messwerte mit dem Faktor 2. Verwenden Sie dafür eine Schleife, die für beliebige Längen der Messreihe korrekt funktioniert.
- n) Geben Sie die beiden Messreihen aus mittels der Methode aus i) aus.
- o) Erzeugen Sie eine dritte Messreihe mit 5 Elementen und belegen Sie diese mit den folgenden Werten: 5 V, 2.3 kV, 4.53 uV, 27 mV, 10 nV.
- p) Erzeugen Sie eine vierte Messreihe mit 5 Elementen zur Aufnahme von Leistungswerten. Die einzelnen Leistungswerte berechnen sich aus den Elementen der Messreihen aus l) und o) mit jeweils gleichem Index. *Sie können hierfür Ihre Funktion aus Übungsblatt 4, Aufgabe 4e) nutzen. Kopieren Sie Ihre Funktion vor die `main`-Funktion in `Messungen.cpp`.*
- q) Ermitteln Sie mit den Methoden aus j) und k) den kleinsten und den größten Leistungswert und geben Sie diese beiden Messwerte aus.