

Praktikum zur Vorlesung Einführung in die Programmierung WS 16/17

Blatt 5

Es können 17 Punkte erreicht werden.

Allgemeine Hinweise

- Bitte lesen Sie vor der Bearbeitung **alle** Aufgaben sorgfältig durch! Dies erspart Ihnen unnötige Arbeit und somit auch Zeit!
- Die einzigen Header, die Sie zur Bearbeitung der Aufgaben verwenden dürfen, sind `iostream` und solche, die laut Aufgabenstellung explizit erlaubt werden.
- Lassen Sie sich fertiggestellte Aufgaben bitte möglichst **frühzeitig** testieren. In der letzten halben Stunde vor Schluss wird nur noch **eine** Aufgabe testiert!
- Wir akzeptieren ein Testat nur, wenn die Lösung eigenständig auf Anhieb erklärt werden kann. Andernfalls müssen wir die entsprechende Teilaufgabe mit 0 Punkten bewerten.

Grundlage: Compilereinstellungen für C++ 11

C++ 11 ist ein Standard für die Programmiersprache C++ aus dem Jahr 2011. Als erstes Element dieses Standards wurde in Kapitel 4 der Vorlesung das Schlüsselwort `nullptr` vorgestellt. Leider ist der Übersetzer (Compiler) von Eclipse beim Anlegen eines neuen Projekts nicht so eingestellt, dass C++ 11 verwendet wird. Die Verwendung von z.B. `nullptr` führt dann mindestens zu einer Warnung, eventuell auch zu einem Fehler.

Damit der Compiler die Quelltexte eines Projekts gemäß dem neuen Standards übersetzt, müssen folgende Schritte durchgeführt werden:

- Öffnen Sie die Projekteigenschaften (Rechtsklick auf den Projektnamen → **Properties**).
- Öffnen Sie die Seite **C/C++ Build** → **Settings**.
- Wählen Sie unter **Configuration** die Einstellung **Debug**.
- Wählen Sie **GCC C++ Compiler** → **Miscellaneous**.
- Fügen Sie im Eingabefeld **Other Flags** zusätzlich zum bereits bestehenden Text folgenden Text hinzu:

```
" -std=c++11 -pedantic"
```

Beachten Sie dabei: Die Anführungszeichen dürfen nicht mit eingegeben werden. Das erste Zeichen (vor dem Bindestrich) muss aber auf jeden Fall das Leerzeichen sein!

- Wiederholen Sie die Anweisungen ab Schritt c) für die Einstellung **Release**.

HINWEIS: Sie müssen diese Änderung einzeln für jedes neue Projekt vornehmen, in dem Sie Funktionalitäten aus C++ 11 verwenden wollen.

EINSCHRÄNKUNG: Im Rahmen des Praktikums möchten wir Sie dazu anregen, die Vorteile von C++ 11 zu nutzen. Deshalb wird ab sofort nur noch die Schreibweise `nullptr` als korrekt anerkannt. Alle anderen Schreibweisen zum Anlegen eines Null-Pointers (z.B. `int* p = 0`) führen fortan zu einer Abwertung der entsprechenden Teilaufgabe.

Aufgabe 1: Zeiger (9 Punkte)

Erstellen Sie ein neues Projekt mit dem Namen `Aufgabe_5_1` und darin eine Quelldatei `zeiger.cpp`.

a) Deklarieren und initialisieren Sie innerhalb Ihrer `main`-Funktion drei `int`-Variablen `x`, `y` und `z` mit den Werten 123, -1 und 0.

Deklarieren Sie danach drei Zeiger `a`, `b` und `c` und initialisieren Sie diese Zeiger so, dass `a` auf `x`, `b` auf `y` und `c` auf `z` zeigt.

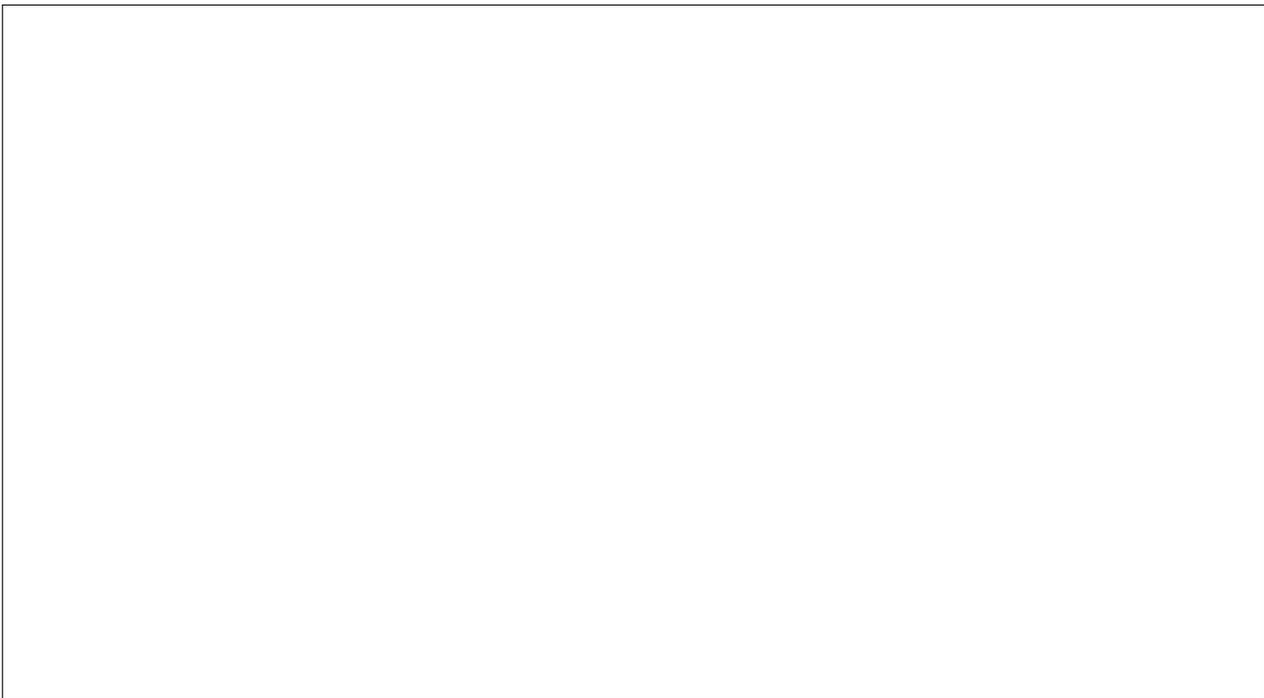
_____ (1)

b) Tauschen Sie die Zeigerwerte so, dass anschließend `a` auf `y` und `b` auf `x` zeigt.

EINSCHRÄNKUNG: Dabei dürfen weder die Inhalte von `x` und `y` geändert, noch dürfen die Variablen `x` und `y` selbst benutzt werden.

_____ (2)

c) Veranschaulichen Sie die Tauschoperationen mit Hilfe einer Zeichnung (Papier und Stift). Dabei ist es hilfreich, die Zustände der relevanten Variablen vor dem Vertauschen und nach jeder Zuweisung darzustellen.



_____ (1)

d) Erhöhen Sie den Wert der Variablen `y` um 2.

EINSCHRÄNKUNG: Dabei dürfen Sie nur eine der Zeigervariablen benutzen, nicht aber die Variable `y` selbst.

_____ (1)

e) Schreiben Sie eine Funktion `addiere()` welche zwei `int`-Werte als Parameter erhält. Die Funktion soll keinen Rückgabewert haben. Der erste Parameter soll dabei als Referenz übergeben werden und in der Funktion soll der zweite Parameter auf den ersten addiert werden.

Erhöhen Sie nun mit Hilfe von `addiere()` die `int`-Variable `y` um 5, so dass das Programmfragment

```
y = 1;
cout << "y=" << y << endl;
addiere(y, 5);
cout << "y=" << y << endl;
```

die folgende Ausgabe liefert:

```
y=1
y=6
```

_____ (2)

f) Geben Sie die Adressen und Inhalte der Variablen gemäß der untenstehenden Beispielausgabe aus.

HINWEIS: Verwenden Sie Leerzeichen und Tabulatoren (Zeichen: `\t`) zur Ausrichtung der Spalten.

```
--- Ausgabe nach Teilaufgabe a) ---
Variable x      Adresse      Inhalt      referenzierter Wert
Variable y      00E2F7A0    123
Variable z      00E2F794    -1
Variable z      00F2A0B6    0

Zeiger a        00E2F788    00E2F7A0    123
Zeiger b        00E2F77C    00E2F794    -1
Zeiger c        00E2E765    00F2A0B6    0
```

```
--- Ausgabe nach Teilaufgabe b) ---
Variable x      Adresse      Inhalt      referenzierter Wert
Variable y      00E2F7A0    123
Variable z      00E2F794    -1
Variable z      00F2A0B6    0

Zeiger a        00E2F788    00E2F794    -1
Zeiger b        00E2F77C    00E2F7A0    123
Zeiger c        00E2E765    00F2A0B6    0
```

```
--- Ausgabe nach Teilaufgabe d) ---
Variable x      Adresse      Inhalt      referenzierter Wert
Variable y      00E2F7A0    123
Variable y      00E2F794    1
Variable z      00F2A0B6    0

Zeiger a        00E2F788    00E2F794    1
Zeiger b        00E2F77C    00E2F7A0    123
Zeiger c        00E2E765    00F2A0B6    0
```

```
--- Ausgabe vor Teilaufgabe e) ---
Variable y      Adresse      Inhalt      referenzierter Wert
Variable y      00E2F794    1
```

```
--- Ausgabe nach Teilaufgabe e) ---
Variable y      Adresse      Inhalt      referenzierter Wert
Variable y      00E2F794    6
```

_____ (2)

Aufgabe 2: Approximation einer Zahl (4 Punkte)

Erstellen Sie zunächst ein neues Projekt `Aufgabe_5_2` und darin eine Quelltextdatei `approx.cpp`.

a) Schreiben Sie eine parameterlose Funktion `approx_pi`, die den Wert π approximiert. Grundlage hierfür ist eine Entdeckung des Mathematikers Leibniz, der folgendes herausgefunden hat:

$$\frac{\pi}{4} \approx 1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} - \frac{1}{11} + \dots$$

Setzen Sie innerhalb der Funktion ein `for`-Schleifenkonstrukt ein, um die Approximation durchzuführen. Insgesamt sollen 10.000 Annäherungsschritte durchgeführt werden. Der approximierte `double`-Wert für $\frac{\pi}{2}$ soll von der Funktion zurückgegeben werden.

HINWEIS: Das Benutzen der Schleifenvariablen in Verbindung mit einer `if-else`-Anweisung ist eine Möglichkeit zur Realisierung der abwechselnden Addition und Subtraktion.

_____ (3)

b) Schreiben Sie ein Hauptprogramm (eine `main`-Funktion), welches den approximierten Wert für $\frac{\pi}{2}$ mit 8 Stellen, den Wert der Konstanten `M_PI` sowie die Differenz, also das Ergebnis von `approx_pi * 2 - M_PI`, auf dem Bildschirm ausgibt.

HINWEIS:

- Inkludieren Sie zusätzlich zu `iostream` die Bibliothek `iomanip` sowie `cmath`.
- Die Anweisung `cout<<setprecision(n);` (mit integer-Argument `n`) legt die Anzahl der auf dem Bildschirm ausgegebenen Stellen (insgesamt vor und nach dem Komma) fest.
- Für den Vergleich kann dann die Konstante `M_PI` für π verwendet werden.

_____ (1)

Aufgabe 3: Palindrome (4 Punkte)

Erstellen Sie zunächst ein neues Projekt `Aufgabe_5_3` und fügen Sie das Codegerüst `palindrome.cpp` hinzu.

Bei einem Palindrom handelt es sich um eine Zeichenkette, die von vorne und hinten gelesen gleich ist. Ein Umdrehen der Zeichenkette würde wieder die gleiche Zeichenkette ergeben, wie z.B. beim Wort „LAGERREGAL“. Implementieren Sie eine Funktion `istPalindrom`, die beliebige Zeichenketten auf diese Eigenschaft hin überprüft. Der Funktionskopf sieht wie folgt aus:

```
bool istPalindrom (char zeichenkette[])
```

Die Funktion `istPalindrom` soll die Zeichenkette durchlaufen und im Falle eines festgestellten Palindroms `true`, andernfalls `false` zurückgeben.

Testen Sie die Funktion mit den in der `main`-Funktion vorbereiteten Beispielen.

_____ (4)