

5. Aufgabenblatt

Der Abgabetermin für Ihre Lösungen ist Montag, der **12. Juni 2017** um 23⁵⁵ Uhr. Die Bewertung erfolgt auf der Grundlage Ihres `git`-Repositories zu diesem Zeitpunkt. Stellen Sie sicher, dass Sie bis zu diesem Zeitpunkt alle Änderungen in Ihr Repository übertragen haben und vergessen Sie nicht das `makefile`!

Aufgabe 12

Schreiben Sie eine Funktion, die das Bitmuster einer Zahl vom Typ `char` ausgibt. Die Funktion soll das Ergebnis nicht direkt anzeigen, sondern einen String mit dem Bitmuster an die aufrufende Funktion zurückgeben:

```
1 string bitmuster (char test)
2 {
3     string muster;
4     ... // Bitmuster "abtasten"
5     return muster;
6 }
```

- ▶ Lagern Sie die `bitmuster`-Funktion in eine separate `cpp`-Datei aus und erstellen Sie eine passende Headerdatei.
- ▶ Rufen Sie Ihre Funktion aus der `main`-Funktion mit einfachen Werten auf und lassen Sie das Ergebnis ausgeben, um die Funktionsweise zu überprüfen.
- ▶ Informieren Sie sich – falls nötig – über die Darstellung von Ganzzahlen im binären Zahlenraum und den bitweisen UND-Operator (`&`).

(6 Punkte)

Aufgabe 13

Informieren Sie sich zunächst mittels einer Informationsquelle Ihrer Wahl über den ASCII-Zeichensatz und schreiben Sie ein Programm, das *alle druckbaren* Zeichen dieses Zeichensatzes und die dazugehörigen (dezimalen) Zahlenwerte in tabellarischer Form ausgibt.

- ▶ Geben Sie die Werte in zwei zusätzlichen Spalten außerdem auch in hexadezimaler Schreibweise und binärer Darstellung aus (siehe Abbildung 1 auf Seite 2).
- ▶ Nutzen Sie die Funktion `bitmuster` aus Aufgabe 12 für die Binärdarstellung.

(4 Punkte)

1	32	20	00100000
2	!	33	21 00100001
3	"	34	22 00100010
4	#	35	23 00100011
5	\$	36	24 00100100
6	%	37	25 00100101
7	&	38	26 00100110

Abbildung 1: Ein Teil der druckbaren Zeichen des ASCII-Zeichensatzes (mit Position im Zeichensatz in dezimaler, hexadezimaler und binärer Schreibweise).

Aufgabe 14

Definieren Sie analog zur Aufgabe 11 eine Klasse namens `Polynom` zur Berechnung eines Polynoms fünften Grades.

- ▶ Die Koeffizienten a_n sollen als private Variablen deklariert sein. Sie werden vom Konstruktor der Klasse initialisiert und sind alle optional. Verwenden Sie die gleichen Standardwerte wie in Aufgabe 11.
- ▶ Implementieren Sie eine Methode zur Ableitung des Polynoms. Der Rückgabewert der Methode soll ebenfalls vom Typ `Polynom` sein.
- ▶ Überladen Sie den `()`-Operator der Polynomklasse, um den Funktionswert $f(x)$ berechnen zu lassen.
- ▶ Erzeugen sie zwei hübsche Polynomobjekte mit den Koeffizienten aus Aufgabe 11 und zwei weitere Polynomobjekte mit den Ableitungen der hübschen Polynome, indem Sie Ihre Ableitungsmethode anwenden.
- ▶ Schreiben Sie ein Programm, das die beiden hübschen Polynome und ihre Ableitungen im Intervall $[-2, 2]$ mit einer Schrittweite von 0,01 zusammen mit den x -Wert ausgibt. Die Ausgabe erfolgt analog zu Aufgabe 11, allerdings mit fünf Spalten.

(8 Punkte)