

28. OKTOBER 2020

# PROGRAMMIEREN, ALGORITHMEN UND DATENSTRUKTUREN I, WS 2020/21 AUFGABENBLATT 1

## 1.1 Installieren einer Entwicklungsumgebung

Willkommen bei Ihrem ersten Praktikum zu Programmieren, Algorithmen und Datenstrukturen I. Die nachfolgenden Aufgaben beziehen sich auf folgende Vorlesungsinhalte: Kapitel 1 *Einleitung* sowie Kapitel 2 *Einführung in die Programmierung*, Abschnitte 1-13.

Für das PAD-Praktikum benötigen Sie eine lauffähige, korrekt eingerichtete *integrierte Entwicklungsumgebung* (IDE, von englisch *integrated development environment*). Es gibt viele kostenfreie gute Alternativen, für diesen Kurs empfehlen wir jedoch die IDE Qt. Die Installation wird kurz in den Vorlesungs-Videos angesprochen, bei Problemen werfen Sie bitte einen Blick auf das hochgeladene PDF *Installationsanleitung\_Qt.pdf*.

Wichtig ist zudem die Installation eines Compilers. Installieren Sie daher zusammen mit Qt die Compilersuite MinWG. Wie in dem pdf gezeigt, können Sie in dem Installationsschritt *Komponenten auswählen* von Qt unter *Qt 5.15.1* den Unterpunkt *MinWG 8.1.0* anwählen. Wählen Sie 32 oder 64 Bit entsprechend Ihrer Rechner-Architektur. MinWG lässt sich jedoch auch separat installieren.

Sollte die Qt-Installation einfach nicht gelingen, lohnt es sich vielleicht, einen Blick auf andere IDEs wie CLion von JetBrains oder Visual Studio von Microsoft zu werfen (vor allem für Mac-User empfehlenswert, hier scheint die Installation von Qt nicht ganz trivial).

Um die erfolgreiche Installation zu testen, gehen Sie wie folgt vor:

- Legen Sie in Qt ein neues Projekt an. Achten Sie dabei darauf, ein *Projekt ohne Qt* zu wählen, und im Untermenü eine *reine C++-Anwendung*.
- Als Build-System wählen Sie *qmake*.
- Bei der Kit-Auswahl sollte das installierte MinGW aufgelistet sein, dieses per Häkchen aktivieren.
- Die Einträge in der Projektverwaltung können leer gelassen werden. Es wird nun ein Projekt erstellt, welches bereits eine Projektdatei sowie eine `main.cpp` enthält.
- Machen Sie sich mit dem generierten Sourcecode vertraut. Ändern Sie dann das vorhandene Hello-World-Programm in der `main.cpp` so ab, dass es Sie bei Programmstart persönlich begrüßt!
- Compilieren Sie das Projekt und führen Sie das Programm aus.

Bitte speichern Sie zukünftig jede bearbeitete Aufgabe separat ab (oder kommentieren Sie die Teillösungen aus), damit Sie alle Ergebnisse bei der Abnahme schnell zur Hand haben und präsentieren können.

## 1.2 Einlesen und Ausgeben

Als erstes richtiges Übungsprogramm lesen Sie nun bitte jeweils eine *Integer-Zahl* per Tastatur ein, transformieren diese in einen *Character* und geben diesen wieder auf der Konsole aus. Integrieren Sie in Ihr Programm zudem sinnvolle Text-Ausgaben, damit der Nutzer weiß, was er zu tun hat und wie die berechnete Ausgabe zu interpretieren ist.

Vergessen Sie nicht, Ihr Programm sinnvoll zu strukturieren und zu kommentieren!

**Achtung:** Das Qt-Fenster *Ausgabe der Anwendung* kann nicht verwendet werden, um eine Tastatureingabe auszuwerten. Hierfür müssen Sie das Programm in einer Konsole starten - gehen Sie daher links auf den Reiter *Projekte*, dann unter *Erstellung und Ausführung* auf *Ausführen* und aktivieren den Punkt *im Terminal ausführen*.

## Debugging

Überlegen Sie zuerst, testen Sie danach: In welchem Wertebereich muss die eingegebene Zahl liegen? Was passiert, wenn der Wertebereich überschritten wird? Wie verhält sich das Programm, wenn Sie bei der Eingabe keine Integer verwenden (sondern Zeichenketten oder Gleitkommazahlen)?

Nutzen Sie den Debugger: Setzen Sie in der ersten Zeile der `main()`-Methode einen Breakpoint. Starten Sie das Programm und gehen Sie zeilenweise Ihr Programm durch. Was passiert jeweils mit Ihrer Eingabe?

## 1.3 Einfache Berechnungen

In dieser Aufgabe wollen wir einfachste Berechnungen durchführen lassen. Lesen Sie hierfür zwei Integer-Zahlen  $n$  und  $m$  ein und geben Sie die Ergebnisse folgender Rechnungen wieder aus:

- $n + m$
- $n - m$
- $n * m$
- $n / m$
- $n \bmod m$

Achten Sie auf korrekte Ergebnisse der Division! Wo ist das Problem?

Debugging: Was passiert bei der Eingabe von 0 als 2. Zahl?

## 1.4 Schleifen

In der Vorlesungen haben wir verschiedene Möglichkeiten kennengelernt, Berechnungen mehrfach hintereinander durchzuführen. Testen Sie Ihr Wissen, indem Sie mithilfe der `for`-, der `while`- und der `doWhile`-Schleife folgende Aufgabe lösen:

Lesen Sie einen positiven Integer  $n$  ein, und geben Sie für jede zweite Zahl von 1 bis  $n$  dessen Quadrat aus.

Hier ein kurzes Beispiel:  $n = 8$ , Ausgabe: 1 ( $= 1^2$ ), 9 ( $= 3^2$ ), 25 ( $= 5^2$ ), 49 ( $= 7^2$ ) .

Was passiert bei negativen Eingaben?

## 1.5 Bonusaufgabe: Binomialkoeffizient

Lesen Sie zwei positive Integer  $k$  und  $n$  ein (mit  $n \geq k$ ) und verwenden Sie eine Schleifenart Ihrer Wahl, um den Binomialkoeffizienten zu berechnen! Der Binomialkoeffizient ist definiert als:

$$\binom{n}{k} = \frac{n!}{k! * (n - k)!} \quad (1)$$

wobei  $n!$  die Fakultät einer Zahl  $n$  beschreibt und als Produkt der Zahlen 1 bis  $n$  definiert ist:

$$n! = 1 * 2 * \dots * (n - 1) * n \quad (2)$$

(Achtung: der Wertebereich eines Integers ist begrenzt, normalerweise auf 32 Bit. Die Werte der Fakultätsberechnung steigen jedoch rasant an, sodass bei 12! schon Schluss ist. Höhere Eingaben für  $n$  oder  $k$  sorgen für falsche Ergebnisse. )