

Einführung in die Programmierung

(B.Sc. CES, B.Sc. Physik; WS17/18)

J. Deussen, M.Sc., J. Hüser, M.Sc., und Prof. Dr. U. Naumann

LuFG Informatik 12: Software and Tools for Computational Engineering
RWTH Aachen



Übungsblatt 1

(Erste Schritte mit C++, g++ und GitLab (git))

Abgabe per git bis zum 24.10.2017 um 23:59 Uhr

Benutzung der Rechner des Rechenzentrums (Cluster)

- Überprüfen Sie über den TIM Account Manager¹, ob der Dienst *Hochleistungsrechnen RWTH Aachen* aktiviert ist.
- Der Zugang zum RWTH Cluster ist nur aus dem RWTH Netzwerk möglich. Sind Sie auf dem Gelände der RWTH können Sie das WLAN *eduroam*² nutzen. Möchten Sie von Zuhause aus arbeiten müssen Sie VPN³ nutzen.
- Unter MS Windows ist der Client *Putty*⁴ zu empfehlen.
- Unter Linux können Sie sich mit Hilfe des Befehls `ssh` auf dem RWTH Cluster einloggen. Ihre persönliche Netzadresse um sich auf dem RWTH Cluster anzumelden lautet

`TIMID@cluster.rz.rwth-aachen.de`

wobei *TIMID* Ihrer TIM Kennung, z.B. ab123456 entspricht.

`ssh -X TIMID@cluster.rz.rwth-aachen.de`

- Ihre Abgaben werden ausschließlich auf dem RWTH Cluster getestet. Falls Sie eine lokale Entwicklungsumgebung nutzen, sollten Sie Ihre Implementierungen vor der Abgabe auf dem RWTH Cluster testen.
- Wir sind genau wie Sie nur Nutzer der RWTH Infrastruktur. Wir haben keine Antworten auf Fragen wie
 - “Warum funktioniert mein TIM Account nicht?”, oder
 - “Wieso kann ich mich nicht ins Eduroam einwählen?”

Bei solchen Fragen wenden Sie sich bitte an den ServiceDesk⁵.

¹<https://www.rwth-aachen.de/selfservice>

²<https://doc.itc.rwth-aachen.de/display/WLAN/Home>

³<https://doc.itc.rwth-aachen.de/display/VPN/Home>

⁴<http://www.putty.org/>

⁵<https://doc.itc.rwth-aachen.de/display/SD/Home>

In dieser Veranstaltung benutzen wir den GitLab Dienst des IT Center der RWTH Aachen. Rufen Sie die Webseite

`http://git.rwth-aachen.de/`

auf und melden sich dort mit Ihrer TIM Kennung an. Falls noch nicht geschehen erstellen Sie unter *Profile Settings* im Reiter *Password* ein Passwort für den GitLab Zugang.

Ihr Tutor wird für Sie ein Repository mit folgendem Namen anlegen:

`programmierung_ws1718_X/Vorname_Nachname`

wobei 'X' Ihre Gruppennummer ist und 'Vorname' sowie 'Nachname' durch Ihren Namen zu ersetzen sind.

Alle Ihre Lösungen sollen über dieses Repository angelegt werden.

Sie können das Repository auf den Cluster-Rechnern herunterladen indem Sie auf der Konsole folgenden Befehl verwenden:

`git clone https://git.rwth-aachen.de/programmierung_ws1718_X/Vorname_Nachname.git`

Erstellen Sie im Repository mittels `mkdir` den Ordner 1. Kopieren Sie in diesen Ordner Ihre Lösung aus Aufgabe 1.1 (nur `hello.cpp` und `README`). Mit dem Befehl

`git add hello.cpp README`

fügen Sie die Dateien dem Repository hinzu. Achten Sie darauf, welche Dateien Sie ins Repository übernehmen. Generierte oder kompilierte Dateien (z.B. *.o Dateien) sollen **nicht** ins Repository übernommen werden. Übermitteln Sie die Dateien mittels `git commit` und `git push` an den Server. Überprüfen Sie den Status des Repository mit:

`git status`

Ein `git rm` löscht lediglich die Datei in der derzeitigen Revision. Schaut man sich vergangene Versionen an, ist die gelöschte Datei dort vorhanden. Nichtsdestotrotz können Sie mit Hilfe von `git reset` und `git revert` Änderungen rückgängig gemacht werden.

Sind Sie mit Ihrer Lösung zufrieden dann sollten Sie den Stand von Ihrem Rechner auf den Server im Rechenzentrum bringen. Das geschieht mit Hilfe von:

`git commit -m "Dieser_Text_beschreibt_die_Aenderungen"`
`git push`

Geben Sie als *Commit*-Kommentar (Option `-m`) eine kurze, aber verständliche Beschreibung der getätigten Modifikationen ein.

1 Erste Schritte mit C++

Erstellen Sie die Datei `hello.cpp` mit folgendem Inhalt:

```
#include <iostream>
int main()
{
    std::cout << "Willkommen_zur_Einfuehrung_in_die_Programmierung!" << std::endl;
    return 0;
}
```

Versehen Sie die einzelnen Zeilen mit erklärenden Kommentaren. Wenden Sie die folgenden Compileraufrufe an und lesen Sie in der Beschreibung⁶ des g++ Compilers nach, was die Aufrufe bedeuten bzw. erzeugen. Schreiben Sie die Erklärungen in eine Textdatei namens `README`.

```
g++ -E hello.cpp
g++ -c hello.cpp
g++ hello.o
g++ -o hello hello.cpp
g++ -Wall -Werror hello.cpp
```

Das Programm soll nun so abgeändert werden, dass nach dem Start eine Eingabe des Benutzers abgefragt wird. Die Eingabe des Benutzers erfolgt über Verwendung von `std::cin`. Gibt der Benutzer z.B. `Klaus` ein soll der Programmablauf so aussehen:

```
\$. / hello
Name? Klaus
Hallo Klaus, willkommen zur Einfuehrung in die Programmierung!
```

Erweitern Sie die Textdatei mit dem Namen `README`: Erläutern Sie die Schritte, die nötig sind um eine ausführbare Datei zu erhalten und wie diese dann aufgerufen werden muss.

2 Versionskontrolle mit GitLab

Im Dokumentationsportal⁷ ist eine Beschreibung von Git zu finden. Schlagen Sie dort die folgenden Optionen des Befehls `git` nach:

- add
- clone
- commit
- pull
- push
- diff
- status
- reset

⁶Mit dem Konsolen-Kommando `g++ --help | less` erhält man eine Übersicht der wichtigsten g++ Optionen. Die Angabe `| less` sorgt für eine seitenweise Ausgabe auf der Konsole.

⁷<https://doc.itc.rwth-aachen.de/display/GIT/GitLab+Startseite>

3 Ein alternatives Modell

Julia und Gretchen haben in den letzten Wochen festgestellt, dass das Vorhersagemodell bisher nur bedingt gute Resultate erzielt hat. Die ursprüngliche Vermutung, dass die Herren nach dem Alkoholkonsum einen leichten Linksdrift haben, scheint nicht allgemein gültig zu sein. Daher entscheiden die beiden, ein alternatives Modell auszutesten:

$$y = p \cdot \sin(0.75x^3) \quad (1)$$

Die Summe der Fehlerquadrate ist dabei durch

$$f(p, \mathbf{x}, \mathbf{y}) = \sum_{i=1}^m \left(p \cdot \sin(0.75x_i^3) - y_i \right)^2$$

gegeben. Zur Minimierung der Summe der Fehlerquadrate muss die partielle Ableitung nach p die Bedingung

$$\frac{\partial f(p, \mathbf{x}, \mathbf{y})}{\partial p} = \sum_{i=1}^m 2 \cdot \sin(0.75x_i^3) \cdot \left(p \cdot \sin(0.75x_i^3) - y_i \right) = 0$$

erfüllen. Daraus ergibt sich die folgende Schätzung für den Parameter p :

$$p = \frac{\sum_{i=1}^m y_i \cdot \sin(0.75x_i^3)}{\sum_{i=1}^m \sin(0.75x_i^3)^2} \quad .$$

Helfen Sie Julia und Gretchen, eine Parameterschätzung für das obige Modell mit Hilfe der kleinsten Fehlerquadrate zu implementieren. Lesen Sie dafür die Textdatei `points.dat` ein, in der sich in der ersten Spalte die x - und in der zweiten Spalte die y -Werte befinden, also die Punkte, an denen Faust und Romeo in den letzten Monaten gesichtet wurden. Erweitern Sie zusätzlich die `README` Datei, in der Sie kurz erläutern, wie Sie das Programm kompilieren und ausführen. Geben Sie außerdem die Schätzung für den Parameter p , sowie den erwarteten Treffpunkt für $x = 1.5$ an.

Hinweis: Um mathematische Funktionen wie zum Beispiel `sin` zu verwenden, sollten Sie die numerische Bibliothek mittels `#include<cmath>` einbinden. Ein Beispielprogramm für die Auswertung von y in Gleichung (1) könnte wie folgt aussehen:

```
#include <iostream>
#include <cmath>
using namespace std;

int main()
{
    float p, x, y;
    cin >> p;
    cin >> x;
    y = p * sin(0.75 * x * x * x);
    cout << y;
    return 0;
}
```