

Algorithmen und Programmierung

Übungsblatt 4

WS 2022/23

Dr. Felix Jonathan Boes

Benedikt Bastin, Ellen Bundschuh, Anna Höpfner, Gina Muuss, Adrian Oeyen, Felix Roth,
Thore Wolf

Ausgabe: 31.10.2022

Abgabe: keine Abgabe; Präsentation in der Übung

Aufgabe 1 (Git - Branches und Merge Requests). Das Werkzeug `git` ist der Quasistandard für die verteilte Zusammenarbeit von gut organisierten Naturwissenschaftler:innen. In der Vorlesung haben Sie mehr über `git` erfahren. Um sich an die Arbeit mit `git` zu gewöhnen, können Sie die Level bis inklusive Kapitel „Shit-Happens“ des Spiels OhMyGit durchspielen. In den Übungen müssen Sie Ihren Spielfortschritt nicht vorführen.

Lösen Sie folgende Teilaufgaben.

- (1) Jedes Gruppenmitglied soll sich `git` auf dem jeweiligen Endgerät einrichten oder `git` auf den Rechnern der Computerpools nutzen.
- (2) Klicken Sie in Flavius, dort wo auch ihr Submissiongrouptoken angezeigt ist, auf den Button **Gitlab-Repository**. So gelangen Sie, innerhalb der `gitlab`-Instanz der Informatik¹, in ihr Abgabe-Repository. In diesem Repository hat Flavius bereits einen Branch mit dem Namen `Zettel_4` angelegt.
- (3) Jedes Gruppenmitglied soll (lokal und mit den zugehörigen `git`-Befehlen) das angelegte Repository klonen. Anschließend wechseln Sie den Branch zu `Zettel_4`. Von dort aus soll ein Featurebranch mit dem Namen `Zettel_4_NACHNAME` lokal angelegt werden.
- (4) In jedem neu angelegten Branch `Zettel_4_NACHNAME` soll ein C++-Programm mit dem Namen `NACHNAME.cpp` angelegt werden welches den Text "`Hallo NACHNAME`" ausdrückt. Diese Änderung soll im besagten Featurebranch `commit`et werden. Der Branch soll anschließend in Ihr Abgabe-Repository `push`et werden.
- (5) Zu jedem Featurebranch soll in `gitlab` ein Merge Request² erstellt werden. Die Merge Requests sollen gemeinsam (an einem Gerät) bearbeitet werden.

Aufgabe 2 (Arrays, Range-based for-loops und Parameterübergabe). Entwerfen und implementieren Sie die folgenden Funktionen. Dabei sollen Sie entscheiden, ob die Parameter via *Call by Value*, *Call by Reference* oder *Call by Const Reference* übergeben werden. Um durch Arrays zu iterieren, sollen Sie Range-Based For-Loops verwenden. Entscheiden Sie, ob dabei Elementkopien, Elementreferenzen oder konstante Elementreferenzen verwendet werden. Begründen Sie Ihre Entscheidungen nachvollziehbar (durch aussagekräftige Kommentare an sinnvollen Stellen).

¹<https://gitlab.informatik.uni-bonn.de>

²https://docs.gitlab.com/ee/user/project/merge_requests/creating_merge_requests.html

- (1) Entwerfen und implementieren Sie eine Funktion, die ein `int`-Array auf der Konsole ausdruckt.
- (2) Entwerfen und implementieren Sie eine Funktion, die alle Werte eines übergebenen `int`-Arrays durch den jeweiligen Betrag ersetzt.
- (3) Entwerfen und implementieren Sie eine Funktion, welche die Einträge eines übergebenen `string`-Arrays zeilenweise auf der Konsole ausdrückt.
- (4) Entwerfen und implementieren Sie eine Funktion, die ganze Zahlen von der Standardeingabe liest, bis die Zahl 42 empfangen wird. Alle bis dahin gelesenen ganzen Zahlen sollen in einem Array gespeichert werden, welcher von der Funktion zurückgegeben wird.

Hinweis: In der Vorlesung haben Sie die Referenz <https://en.cppreference.com> kennen gelernt. Dort finden Sie alle verfügbaren Memberfunktionen von dynamische Arrays³ und Strings⁴

Aufgabe 3 (Programmcodeaufteilung). Um größere Softwareprojekte übersichtlich zu gestalten und um die Kompilierzeit zu verringern, wird der Programmcode üblicherweise in Headerdateien, Quelldateien und Demonstrationsbeispiele aufgeteilt. Die Headerdateien enthalten (unter anderem) die Funktionsdeklarationen und die Quelldateien enthalten (unter anderem) die Funktionsimplementierungen.

Auf der Kurswebseite finden Sie ein Dateiarhiv, das ein unfertiges Softwareprojekt enthält. Das Projekt hat die folgende Struktur.

```

+-- include
|   |
|   +-- meine_funktionen.hpp    // Header
|
+-- src
|   |
|   +-- meine_funktionen.cpp    // Quellen
|
+-- examples
|   |
|   +-- main.cpp                // Demo

```

Im Ordner `./include/` befinden sich die Headerdateien, im Ordner `./src/` die Quelldateien und im Ordner `./examples/` befindet sich ein Beispielpogramm, welches die bereit gestellten Funktionen demonstrieren soll. Um das Projekt zu kompilieren, soll (im Stammverzeichnis des Projekts) folgender Compileraufruf verwendet werden.

```
clang++ -std=c++17 -I./include src/meine_funktionen.cpp examples/main.cpp -o testprogramm
```

Zur Aufgabe: Ergänzen Sie die Headerdatei um aussagekräftige Kommentare, die Quelldatei um (nachvollziehbare, wartbare und gut kommentierte) Funktionsimplementierungen und demonstrieren Sie die Verwendung der Funktionen in der `main.cpp`.

Aufgabe 4 (Fremde Bibliotheken nutzen). Fremde Bibliotheken korrekt einzubinden ist (je nach verwendetem Betriebssystem und Popularität der Bibliothek) unterschiedlich zeitaufwendig. In dieser Aufgabe wollen wir die `{fmt}`-Library verwenden, aber ihre Einrichtung überspringen. Dazu nutzen wir eine, durch diesen Link konfigurierte, (Klick \rightsquigarrow) Godbolt-Instanz (\Leftarrow [link](#)).

Zur Aufgabe: Bringen Sie das Vorlesungsbeispiel zur `{fmt}`-Library auf der besagten Godbolt-Instanz zum Laufen.

³<https://en.cppreference.com/w/cpp/container/vector>

⁴https://en.cppreference.com/w/cpp/string/basic_string