

1 Kennenlernen der Entwicklungs-Umgebung

1.1 Aufgabe

- Installieren Sie die Umgebung wie in den Folien und im [README.md](#) beschrieben.
- Probieren Sie die Shortcuts aus.
- Laden Sie den cmake-Cache neu und bauen Sie das Target: `example_app`.
- Lassen Sie die Applikation nun über das «Run»-Menü oder den Button in der Toolbar laufen.
- Setzen Sie in `src/main.cpp` einen Break-Point neben der Zeilennummerierung und lassen die Applikation im Debug-Modus laufen. Versuchen Sie die Navigation im Debug-Modus.

2 Erstellen eines eigenen Projektes

2.1 Aufgabe

- Kopieren Sie das `example` Projekt und setzen Sie einen neuen Namen für die Kopie.
- Erneuern Sie den cmake-Cache und bauen Sie das neue Projekt.

2.2 Lösung

Ändern Sie im kopierten `CMakeLists.txt` die Zeile

```
1 set(project_name example)
```

zu

```
1 set(project_name primitive_typen)
```

3 Primitive Typen

3.1 Aufgabe

- Leeren Sie die `main()`-Funktion im kopierten Projekt.
- Schreiben Sie nun eine Applikation, welche immer 2 dezimale Integer Parameter erwartet und diese einliest.
- Nutzen Sie die Standard-Library um die Strings in Integer zu parsen.
- Die beiden Dezimal-Werte sollen dann nach `float` gecastet und dividiert werden.

- Stellen Sie sicher, dass der zweite Parameter nie 0 oder negativ ist.
- Geben Sie das Resultat auf der Konsole aus.
- In allen Fehler-Fällen brechen Sie ab und geben eine Fehlermeldung aus.

3.2 Lösung

main.cpp

```
1  #include <cstdlib>
2
3  #include <string>
4  #include <iostream>
5
6  int main(int argc, char* argv[]) {
7      if (argc != 3) {
8          std::cout << "Expected 2 parameters. Got: " << argc - 1 << std::endl;
9          return EXIT_FAILURE;
10     }
11
12     const int param1 = std::stoi(argv[1]);
13     const int param2 = std::stoi(argv[2]);
14
15     if (param2 <= 0) {
16         std::cerr << "Expected second parameter to be bigger then zero. Got: " <<
17             param2 << std::endl;
18         return EXIT_FAILURE;
19     }
20
21     const auto result = static_cast<float>(param1) / static_cast<float>(param2);
22
23     std::cout << result << std::endl;
24
25     return EXIT_SUCCESS;
26 }
```