

Praktikum zur Vorlesung Einführung in die Programmierung WS 25/26

Blatt 0

Es können 0 Punkte erreicht werden.

Allgemeine Hinweise

1. Dieses freiwillige Blatt dient dazu, Sie mit der Entwicklungsumgebung vertraut zu machen. Dementsprechend wird es nicht testiert.
2. Die Bearbeitungszeit dieses Blattes reicht bis zum 24.10.2025. Die reguläre Bearbeitungszeit der restlichen Praktikumsblätter beträgt je eine Woche.

Aufgabe 1: Internetzugang (0 Punkte)

1. Besuchen Sie in einem Webbrowser (z.B. Firefox) den Moodle-Raum des Praktikums. Die Adresse lautet:
<https://moodle.tu-dortmund.de/course/view.php?id=54266>
Fügen Sie die Seite den Lesezeichen (Favoriten) hinzu, damit Sie später schnell auf die Aufgabenzettel zugreifen können.
2. Suchen Sie im Moodle-Raum die Namen der Tutoren Ihrer Praktikumsgruppe und notieren Sie diese (beispielsweise in einer Textdatei – Sie müssen dieses Blatt nicht ausdrucken). Notieren Sie außerdem Ihren Namen und Ihre Matrikelnummer.

_____ (0)

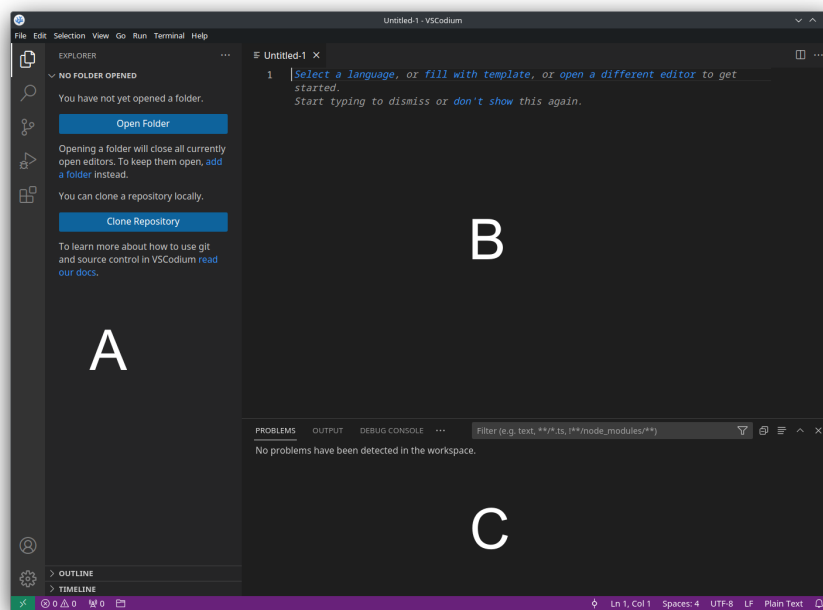


Abbildung 1: Hauptfenster von VSCodium

Aufgabe 2: Kennenlernen der Programmierumgebung (0 Punkte)

Diese Aufgabe behandelt die Vorgehensweise für die vorbereiteten Linux-Rechner des IRB-Pools im OH12 und im Retina-Pool. Wir empfehlen in beiden Fällen die Nutzung von Visual Studio Code.

Eine Auswahl an Anleitungen zur Installation von Visual Studio Code auf Ihrem Heimrechner finden sich im Moodle-Raum unter *Anleitungen*.

Entwicklungsumgebungen wie Visual Studio Code helfen Programme in C++ (und vielen weiteren Programmiersprachen) zu schreiben und auszuführen. Wir benötigen zu Beginn nur einen sehr kleinen Teil dieser Möglichkeiten. Daher werden Sie im Folgenden schrittweise an diese Funktionen herangeführt. Gehen Sie hier mit Bedacht vor, da nur leicht abweichende Einstellungen später zu Problemen führen können.

IRB-Pools mit Linux

In den Pools der IRB ist **VSCode** (und die Variante VSCodium) vorinstalliert, jedoch sind einige zusätzliche Schritte zur Einrichtung einer C++ Umgebung notwendig. Führen Sie folgende Schritte aus:

1. Beim Einloggen in die Poolrechner wählen sie durch das menü unten links ihre gewünschte Desktopoberfläche aus. Wir empfehlen „KDE Plasma“.
Bei der Anmeldung tauchen ggf. ein Fenster eines „GPG-Assistenten“ auf, das sie wegklicken können.
2. Über das K-Menü (Startmenü von Debian GNU/Linux):
Geben Sie in der Suchleiste „code“ ein und starten Sie das gefundene Programm. Einige Sekunden später öffnet sich das Fenster von VSCode. Als ersten Schritt wählen Sie Ihren *workspace*. Navigieren Sie hierzu über File → Open Folder. Klicken Sie auf „Persönlicher Ordner“. Erstellen Sie hier den Ordner **workspace** und wählen Sie diesen Ordner aus.

3. VSCode benötigt einige Erweiterungen, um die Arbeit mit C++ angenehm zu gestalten. Folgen sie der Anleitung ab dem Abschnitt „Einrichtung und Benutzung von Visual Studio Code“

Retina-Pools mit Windows

1. Über das Windows-Startmenü:
Starten Sie „Visual Studio Code for C++“. Als ersten Schritt wählen Sie Ihren *workspace*. Im Retina-Pool sollten Sie den Ordner `R:\workspace` erstellen, da dieser Ordner von jedem Poolrechner aus erreichbar ist. Navigieren Sie hierzu über File → Open Folder. Klicken Sie auf „Dieser PC“ und dann auf den Datenträger mit Ihrem Kürzel (sm...). Erstellen Sie hier den Ordner **workspace** und wählen Sie diesen Ordner aus.
2. VSCode benötigt einige Erweiterungen, um die Arbeit mit C++ angenehm zu gestalten. Folgen sie der Anleitung ab dem Abschnitt „Einrichtung und Benutzung von Visual Studio Code“

Eigenes Gerät

Wenn sie ihr eigenes Gerät benutzen möchten, können sie der Anleitung für Windows oder MacOS folgen. Sollten sie privat Linux benutzen, gehen wir davon aus, dass sie VSCode eigenständig installieren können und anschließend der Anleitung ab dem Abschnitt „Einrichtung und Benutzung von Visual Studio Code“ folgen.

Überblick über VSCode

1. Sie sehen nun das Hauptfenster von Visual Studio Code. Es ist in mehrere Bereiche (Abbildung 1) unterteilt:
 - A** Hier werden alle erstellten Ordner und die dazugehörigen Dateien angezeigt („Explorer“).
 - B** In diesem Bereich werden geöffnete Dateien angezeigt.
 - C** In diesem Bereich werden mehrere „Views“ angezeigt. Zwischen den „Views“ kann durch Klicken auf den entsprechenden Tab, oder die Menüleiste am oberen Fensterrand gewechselt werden. Für Sie sind zunächst nur zwei „Views“ von Bedeutung:
 - Problems:** Zeigt syntaktische Fehler des Programms an.
 - Terminal:** Zeigt die Programmausgaben an und ermöglicht eine Ausführung von Kommandos.
2. Legen Sie mithilfe des Explorer Tabs unter **workspace** einen neuen Ordner **Praktikum_00** an (Rechtsklick → „New Folder...“).

_____ (0)

Aufgabe 3: Erstellen eines ersten Programms (0 Punkte)

1. Erstellen Sie einen neuen Unterordner **Aufgabe_3** im Ordner **Praktikum_00**. Die Quelldateien jeder Aufgabe werden **immer** in einem eigenen Ordner abgelegt; diese Konvention soll bei zukünftigen Projekten beachtet werden. Der Hintergrund ist, dass wir bei der Einrichtung der Entwicklungsumgebung dafür sorgen, dass alle Quelldateien in einem Ordner gemeinsam übersetzt werden. Eine Trennung der Aufgaben ist erforderlich, um Konflikte bei der Übersetzung zu vermeiden.

Erstellen Sie nun eine neue Quelldatei mit dem Namen **zeitumrechnung.cpp** in diesem Ordner (Achtung: Dateieindung **.cpp** beachten). Dazu klicken sie mit der rechten Maustaste auf den Ordner **Aufgabe_3** und wählen „New File...“.
2. Sie sollen nun in Bereich **B** (mit Titelzeile **zeitumrechnung.cpp**) das folgende Programm (möglichst fehlerfrei) eintippen, welches die Umwandlung von Sekunden in Tage, Stunden, Minuten und Sekunden implementiert.

```
1  #include <iostream>
2  using namespace std;
3
4  int main (){
5      int eingabe { 172169 }; // Eingabe in Sekunden
6      int sekunden { 0 };
7      int minuten { 0 };
8      int stunden { 0 };
9      int tage = { 0 };
10
11     cout << eingabe << " Sekunden entsprechen:" << endl;
12
13     tage = eingabe / 86400;
14     eingabe = eingabe - tage * 86400;
15     stunden = eingabe / 3600;
16     eingabe = eingabe - stunden * 3600;
17     minuten = eingabe / 60;
18     eingabe = eingabe - minuten * 60;
19     sekunden = eingabe;
20
21     cout << tage << " Tagen, "
22          << stunden << " Stunden, "
23          << minuten << " Minuten und "
24          << sekunden << " Sekunden.\n";
25     return 0;
26 }
```

HINWEIS: Für einige Sonderzeichen müssen Sie die „Alt Gr“-Taste verwenden. Öffnende, geschweifte Klammern „{“ werden z.B. eingegeben, indem Sie die „Alt Gr“-Taste gedrückt halten und dann zusätzlich die 7 drücken. Die schließende, geschweifte Klammer „}“ wird analog eingegeben – nur mit der Taste 0 statt der 7.

HINWEIS: Wenn Sie den Quellcode per Copy & Paste aus der PDF-Datei kopieren, dann enthält der Quellcode ungültige Zeichen, welche nicht sichtbar oder den regulären Zeichen recht ähnlich sind (Es gibt z.B. einen Unterschied zwischen einem Minus-Zeichen (–) und einem Spiegelstrich (-)). Diese Zeichen führen zu Problemen beim späteren Kompilieren und Ausführen des Programms. Schreiben Sie den Quellcode daher per Hand ab!

3. Bestimmen Sie, was die Farben im Programmtext (Quellcode) signalisieren sollen (Farben der Buchstaben, nicht die Zeilenmarkierungen oder rot unterschlängelte Worte). Füllen Sie dazu die folgende Tabelle aus. Unter Umständen treten mehr als die drei vorgegebenen Farben auf.

Farbe	Bedeutung
rot	
blau	
grün	

4. Bitte speichern Sie dieses Projekt („File → Save All“). Tragen Sie den Pfad des gespeicherten Projekts hier leserlich ein. Den Pfadnamen erhalten Sie in VSCode durch einen Rechtsklick auf die Quelldatei im Explorer (oder den Editor Tab) und dann auf „Copy Path“ – der Pfad ist dann in der Zwischenablage kann in ein Textdokument eingefügt werden.

(0)

Aufgabe 4: Kompilieren und Ausführen eines Programms (0 Punkte)

Der Rechner versteht die Sprache C++ nicht direkt. Ein C++-Programm muss daher zunächst in die Maschinsprache des Rechners übersetzt („kompiliert“) werden. Anschließend muss es noch mit Teilen verwendeter „Bibliotheken“ verknüpft („gelinkt“) werden.

1. Kompilieren Sie das Programm. Dies erreichen Sie folgendermaßen:

- Kopieren Sie den Pfad des **Ordners** der Aufgabe 3 in der „Explorer“-Ansicht (Rechtsklick auf den Ordner → „Copy Path“).
- Öffnen Sie in VSCode die „Terminal“-Ansicht.
- Navigieren Sie mit dem Terminal in den Zielordner mithilfe des Kommandos `cd <Zielpfad>`, wobei Sie den kopierten Pfad an Stelle von `<Zielpfad>` einfügen (Rechtsklick → Paste / Strg+Shift+V / CMD+V). Bestätigen Sie das Kommando mit der Eingabetaste.
- Führen Sie das Kommando zum Kompilieren aus: `g++ *.cpp -o <programmname>.exe`

HINWEIS: Unter MacOS wird „clang++“ an Stelle von „g++“ benutzt. Außerdem entfällt die Dateiendung „.exe“ unter MacOS und Linux immer. Platzhalter wie „<programmname>“ oder „<Zielpfad>“ müssen von ihnen durch sinnvolle Inhalte ersetzt werden.

- Wenn ein Fehler angezeigt wird (im „Terminal“, oder in der „Problems“-Ansicht, oder durch rotes Unterschlingeln der entsprechenden Stelle), korrigieren Sie Ihren Programmtext und kompilieren ihn erneut.
- Führen Sie das Programm aus, indem Sie es als Kommando aufrufen: `./<programmname>.exe`.
- Prüfen Sie, ob die Ausgabe des Programms Ihren Erwartungen entspricht (per Hand oder Taschenrechner berechnet oder per Proberechnung überprüft). Tragen Sie die Werte in die Tabelle ein.

Zugewiesener Wert für eingabe	Vom Programm ermittelte Werte für tage/stunden/minuten/sekunden	Von Hand berechnete Werte für tage/stunden/minuten/sekunden
172169		

- Ändern Sie den Wert, welcher der Variable **eingabe** zugewiesen ist in „272“. Kompilieren Sie das Programm neu, führen Sie es aus und prüfen Sie die Ausgabe erneut.

Zugewiesener Wert für eingabe	Vom Programm ermittelte Werte für tage/stunden/minuten/sekunden	Von Hand berechnete Werte für tage/stunden/minuten/sekunden
272		

HINWEIS: Das Kompilieren und Ausführen ihres Programms kann in VSCode durch Klicken des grünen Dreiecks oben rechts im Fenster erreicht werden. Diese Funktion führt die oben genannten Schritte automatisch aus. In dieser Aufgabe wurden die Schritte manuell ausgeführt, um diesen Hintergrund aufzuzeigen.

(0)

Aufgabe 5: Fehlermeldungen und Warnungen (0 Punkte)

Erstellen Sie für jede der fünf folgenden Teilaufgaben a) – e) eigene Projekte mit den Namen `Aufgabe_5a`, `Aufgabe_5b`, `Aufgabe_5c`, `Aufgabe_5d` und `Aufgabe_5e`.

Legen Sie in jedem der Projekte eine neue C++-Quell-Datei mit dem Namen `zeitumrechnung.cpp` an. Kopieren Sie den Quelltext aus Aufgabe 3 in jede Datei.

Führen Sie die in den Teilaufgaben a) - e) angegebenen Veränderungen am Quelltext durch und kompilieren Sie diesen. Achten Sie darauf, dass Sie die richtige Datei des entsprechenden Projekts editieren! Schreiben Sie eine kurze Interpretation der angezeigten Fehlermeldungen und Warnungen (in der „Problems“- und „Terminal“-Ansicht!).

- a) Ersetzen Sie `„int eingabe { 172169 };“`
durch `„int Eingabe { 172169 };“`.

- b) Ersetzen Sie `„eingabe = eingabe - stunden * 3600;“`
durch `„eingabe - stunden * 3600 = eingabe;“`.

- c) Ersetzen Sie `„tage = eingabe / 86400;“`
durch `„tage = eingabe // 86400;“`.

- d) Ersetzen Sie `„eingabe = eingabe - minuten * 60;“`
durch `„ eingabe == eingabe - minuten * 60;“`.

- e) Ersetzen Sie `„minuten = eingabe / 60;“`
durch `„minuten = eingabe / 0;“`.
