

Elias Kuthe
Roman Kalkreuth
Bernd Zey

Dortmund, den 18. Oktober 2019

Praktikum zur Vorlesung Einführung in die Programmierung WS 19/20

Blatt 1

Es können 10 Punkte erreicht werden.

Allgemeine Hinweise

1. Bitte lesen Sie vor der Bearbeitung **alle** Aufgaben sorgfältig durch! Dies erspart Ihnen unnötige Arbeit und somit auch Zeit!
2. Die einzigen Header, die Sie zur Bearbeitung der Aufgaben verwenden dürfen, sind `iostream` und solche, die laut Aufgabenstellung explizit erlaubt werden.
3. Lassen Sie sich fertiggestellte Aufgaben bitte möglichst **frühzeitig** testieren. In der letzten halben Stunde vor Schluss wird nur noch **eine** Aufgabe testiert!
4. Wir akzeptieren ein Testat nur, wenn die Lösung eigenständig auf Anhieb erklärt werden kann. Andernfalls müssen wir die entsprechende Teilaufgabe mit 0 Punkten bewerten.
5. Sie werden nur in dem Praktikumstermin testiert, zu dem Sie auch angemeldet sind. Ausnahmen müssen **vor Beginn des jeweiligen Termins** mit den Betreuern abgestimmt sein.
6. Schreiben Sie auf jede Seite ihren Namen und ihre Matrikelnummer!
7. Bei den folgenden Aufgaben muss nach dem Betriebssystem unterschieden werden. Im Retina-Pool (P1-01-108 A+B) ist **Windows** installiert und in den Informatik-Pools (Otto-Hahn-Straße 12) **Debian GNU/Linux**.

Aufgabe 1: Internetzugang (1 Punkt)

1. Besuchen Sie in einem Web-Browser (z.B. Firefox) die Homepage der Vorlesung. Die Adresse lautet:
`https://ls11-www.cs.tu-dortmund.de/teaching/ep1920vorlesung`
Fügen Sie diese Seite den Lesezeichen (Favoriten) hinzu, damit sie später schnell auf die Vorlesungsfolien und Aufgabenzettel zugreifen können.
2. Suchen Sie auf den Seiten zum Praktikum die Tutoren Ihrer Praktikumsgruppe und notieren Sie hier die Namen.

_____ (1)

Aufgabe 2: Kennenlernen der Programmierumgebung Eclipse (2 Punkte)

Eclipse ist eine Entwicklungsumgebung zum Entwickeln von Programmen in der Sprache C++ (und vielen weiteren Programmiersprachen). 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.

1. Über das K-Menü (Debian) oder das Startmenü (Windows):
Starten Sie **Eclipse 4.6 Neon** (Debian) bzw. **Eclipse 4.7 Oxygen** (Windows). Beim ersten Start fragt Eclipse nach einem Pfad zu dem *workspace*: In diesem Ordner werden alle Projekte gespeichert, welche im Laufe des Praktikums erstellt werden. Zudem legt Eclipse in diesem Ordner einige Einstellungen der Benutzeroberfläche ab (z.B. welche Projekte aktuell geöffnet sind oder wie die einzelnen Unterfenster von Eclipse angeordnet wurden). Unter Debian können sie den vorgeschlagenen Pfad beibehalten. Unter Windows **müssen** Sie diesen auf `R:\workspace` ändern (Achtung: Falls Sie dies nicht tun, gehen alle gespeicherten Projekte beim Abmelden verloren!). Setzen Sie außerdem das Häkchen, welches besagt, dass Eclipse standardmäßig diesen Pfad verwenden soll und klicken Sie auf „OK“. Sie sehen nun einen Willkommen-Tab, welcher durch Klicken auf das `x` geschlossen werden kann.
2. Nehmen Sie über den Menüeintrag „Window → Preferences“ die folgenden Einstellungen vor:
 - General → Workspace → Save automatically before build: YES
 - General → Editors → Text Editors → Spelling → Enable spell checking: NO
 - C/C++ → Editor → Save Actions → Format source code: YES
 - Run/Debug → Console → Limit console output: NO
 - *Nur unter Windows:* C/C++ → New C/C++ Project Wizard → „MinGW GCC“ auswählen, dann Klick auf „Make toolchain(s) preferred“
3. Legen Sie über das Menü (File → New → Project) ein neues C++-Projekt an. Im daraufhin erscheinenden Dialog wählen Sie „C++ Project“ und klicken dann auf „Next“. Geben Sie dem Projekt nun den Namen **Aufgabe_01** und wählen Sie den Projekttyp „Empty Project“ mit der Toolchain „Linux GCC“ (Debian) oder „MinGW GCC“ (Windows). Die Toolchain sollte dabei durch die Konfiguration im letzten Schritt bereits korrekt gesetzt sein. Den nächsten Dialog mit dem Titel „Select Configurations“ können Sie überspringen („Next“ klicken). Anschließend klicken Sie auf „Finish“. Nach dem Erstellen des Projekts fragt Eclipse, ob Sie die C++-Perspektive öffnen wollen. Wählen Sie „Remember my decision“ und klicken Sie auf „Yes“.
4. Sie sehen nun das Hauptfenster von Eclipse wie in Abbildung 1. Es ist unterteilt in mehrere Bereiche:
 - A** Hier werden alle erstellten Projekte und die dazugehörigen Dateien angezeigt („Project 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 gewechselt werden. Für Sie sind zunächst nur zwei „Views“ von Bedeutung:
 - Problems:** Zeigt syntaktische Fehler des Programms an.
 - Console:** Zeigt die Programmausgaben an.

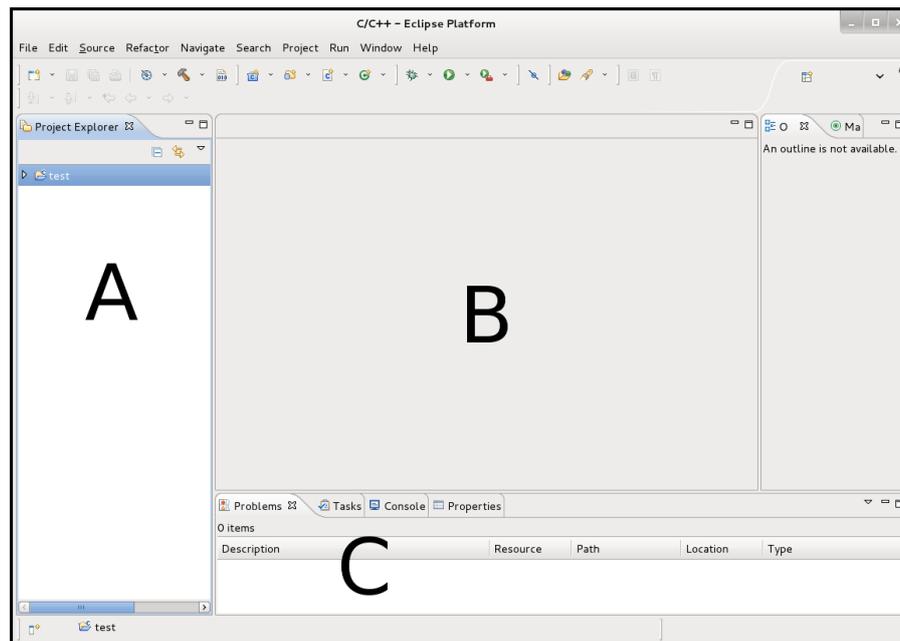


Abbildung 1: Hauptfenster von Eclipse

Aufgabe 3: Erstellen eines ersten Programms mit Eclipse (2 Punkte)

1. Erstellen Sie einen neuen Ordner `src` (Abkürzung für Sources) in dem Projekt `Aufgabe_01`: klicken Sie mit der rechten Maustaste auf das Projekt und wählen dann „New → Folder“. In diesem Ordner werden **immer** die Quelldateien abgelegt; diese Konvention soll bei zukünftigen Projekten beachtet werden.

Erstellen Sie nun eine neue Quelldatei mit dem Namen `zeitumrechnung.cpp` in diesem Ordner (Achtung: Dateierendung `.cpp` beachten). Dazu klicken sie mit der rechten Maustaste auf den Ordner `src` 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!

Farbe	Bedeutung
violett	
blau	
grün	

4. Bitte speichern Sie dieses Projekt („File → Save All“). Tragen Sie den Pfad des gespeicherten Projekts hier leserlich ein. Den Pfadnamen („Location“) erhalten Sie durch Klicken auf den Projektnamen (mit der rechten Maustaste) und Auswählen von „Properties“ (Projekteigenschaften).

_____ (2)

Aufgabe 4: Kompilieren und Ausführen eines Programms mit Eclipse (2 Punkte)

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

1. Kompilieren Sie das Programm, indem Sie im Menü „Project“ den Unterpunkt „Build All“ auswählen.
2. Wenn ein Fehler angezeigt wird (in der Problems-View oder durch rotes Unterschlängeln der entsprechenden Stelle), korrigieren Sie Ihren Programmtext und kompilieren ihn erneut.
3. Führen Sie das Programm aus (grüner Pfeil oben in der Mitte, oder mit „Run“ über das Menü) und 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.
4. 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
172169		

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

_____ (2)

Aufgabe 5: Fehlermeldungen und Warnungen (3 Punkte)

Erstellen Sie für jede der fünf folgenden Teilaufgaben a) – e) ein eigenes Projekt mit den Namen `Aufgabe_01_5a`, `Aufgabe_01_5b`, `Aufgabe_01_5c`, `Aufgabe_01_5d` und `Aufgabe_01_5e`.

Legen Sie in jedem der Projekte jeweils einen Ordner `src` und 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-View und der Console-View!).

- 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;“.

_____ (3)