

Praktikum Informatik Installationsanleitung

September 22, 2025

Contents

Vorwort	3
1 Windows	4
1.1 Installation von Java	4
1.2 Installation von Eclipse und MinGW	4
1.3 Installation und Konfiguration von Boost	7
2 Linux (Ubuntu, Version 22.04)	8
2.1 Installation von Java	8
2.2 Installation von Eclipse	8
2.3 Installation und Konfiguration von Boost	9
3 MacOS	10
3.1 Installation Xcode Command Line Tools	10
3.2 Installation der Abhängigkeiten	10
3.3 Installation und Konfiguration Eclipse	10
3.4 Installation und Konfiguration von Boost	11
3.5 Debugger Konfiguration	12

Vorwort

Beachten Sie bitte die in in dieser Anleitung angegebenen Befehle oder Angaben nicht aus diesem Dokument zu kopieren und einzufügen. Dabei kommt es zu Fehlern, da Steuerzeichen aus dem Dokument mit kopiert werden und dadurch der eigentliche Befehl nicht mehr funktioniert. Nehmen Sie sich bitte die Zeit die Befehle selbständig einzugeben.

Sollten Sie bereits über eine Installation von Eclipse verfügen, sollten Sie diese Anleitung dennoch komplett durcharbeiten um zu überprüfen, ob die in diesem Praktikum benötigten Versionen der Software installiert sind. Dies ist für eine reibungslose Durchführung des Praktikums unerlässlich und erspart Ihnen viele Probleme.

1 Windows

1.1 Installation von Java

- a. Testen Sie, ob Java bereits auf Ihrem System in der Version 20.0.2 installiert ist. Dazu öffnen Sie eine Eingabeaufforderung (Startmenü → suchen nach cmd.exe → ausführen) und geben Sie den Befehl "java -version" ein. Sollten Sie eine Fehlermeldung erhalten, müssen Sie mit Punkt *b* fortfahren. Erscheint eine Ausgabe mit der Angabe der Java-Version 20.0.2, ist Java in der richtigen Version bereits auf Ihrem System installiert und es bedarf keiner weiteren Schritte. Sollte Ihnen eine Version kleiner als 20.0.2 angezeigt werden, sollten Sie die Version deinstallieren und mit Punkt *b* fortfahren.
- b. Laden Sie die "jdk-20.0.2.zip"-Datei aus Moodle für Windows herunter und entpacken Sie diese nach "C:\jdk-20.0.2"
- c. Passen Sie die Systemvariablen Ihres Systems an. Dazu suchen Sie im Startmenü von Windows nach "Systemumgebungsvariablen bearbeiten" und führen das Tool aus. In dem Fenster Systemeigenschaften klicken Sie auf Umgebungsvariablen (s. Figure 1.1).
- d. In dem neu geöffneten Fenster finden Sie im unteren Bereich den Abschnitt Systemvariablen. Hier wählen Sie nun "Path" aus und klicken auf "Bearbeiten...".
- e. Nun erstellen Sie mit dem Klicken auf Neu einen neuen Eintrag und geben dort als Text "C:\jdk-20.0.2\bin" ein und bestätigen die Änderungen mit dem Klick auf "OK". Wählen Sie nun mit der Maus den neuen Eintrag aus und klicken Sie auf der rechten Seite so lange auf die Schaltfläche Nach oben bis der neue Eintrag ganz oben erscheint. Bestätigen Sie mit "OK" und Schließen Sie alle anderen Fenster.
- f. Nun können Sie wie unter Punkt *a* beschrieben erneut testen, ob Java auf Ihrem System installiert ist. Beachten Sie dabei, dass Sie dazu die Eingabeaufforderung neu öffnen müssen.

1.2 Installation von Eclipse und MinGW

- a. Laden Sie die ZIP-Datei "mingw64.zip" aus Moodle herunter und entpacken Sie diese nach "C:\mingw64".

1 Windows

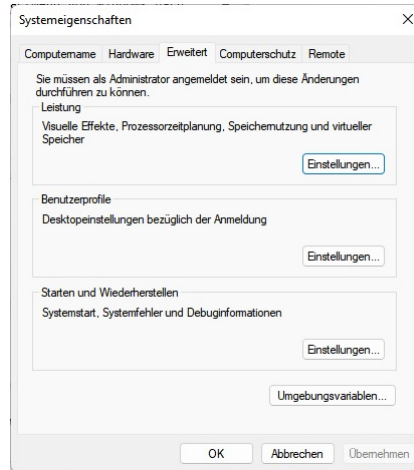


Figure 1.1: Systemeigenschaften

- b. Fügen Sie wie unter 1.1.b – 1.1.e erläutert der "Path-Variable" den neuen Wert "C:\mingw64\bin" hinzu.
- c. Als nächsten laden Sie den Eclipse Installer herunter (<https://www.eclipse.org/downloads/download.php?file=/oomph/epp/2025-09/R/eclipse-inst-jre-win64.exe>) und führen die heruntergeladene Datei aus.
- d. Bei der Auswahl der zu installierenden Elemente wählen Sie Eclipse "IDE for C/C++ Developers" aus (s. Figure 1.2).
- e. Im nächsten Dialogfenster wählen Sie den Installationspfad (dieser kann unverändert bleiben) aus und bestätigen die Installation (Auswahl der Java VM bleibt unverändert, s. Figure 1.3).
- f. Nun ist Eclipse auf Ihrem System in dem ausgewählten Pfad installiert und kann gestartet werden. Dazu klicken Sie in dem Installer auf Launch oder führen später die "eclipse.exe-Datei" in dem Installationspfad aus.
- g. Beim Start von Eclipse werden Sie nach dem Ort für den Workspace gefragt. In dem Workspace speichert Eclipse alle Projekte und die dazugehörigen Metadaten ab. Wählen Sie bitte einen entsprechenden Pfad auf Ihrem System aus und bestätigen Sie die Auswahl mit Launch (s. Figure 1.4).

1 Windows

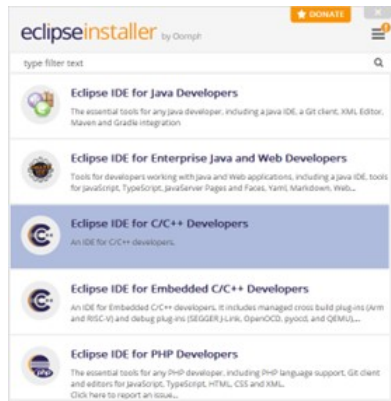


Figure 1.2: Eclipse Installer

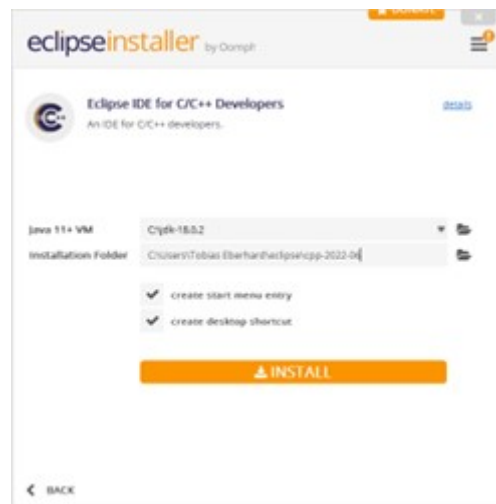


Figure 1.3: Eclipse Installer 2

- h. Testen Sie nun die Installation von Eclipse und MinGW-w64, indem Sie ein neues Projekt in Eclipse anlegen. Dazu wählen Sie "File → New → C/C++ Project → C++ Managed Build" aus und klicken Sie auf "Next". Geben Sie nun einen Projektnamen (z.B. "Aufgabenblock_1") ein, wählen Sie als Project type "Hello World C++ Project" und als Toolchain "MinGW GCC" aus. Legen Sie das Projekt mit "Finish" an. Nun haben Sie Ihr erstes Projekt mit einem kleinen Beispielprogramm angelegt. Damit Sie dieses kleine Test-Programm ausführen können, müssen Sie es zuerst kompilieren. Dazu wählen Sie "Project → Build Project" aus. Nach dem der Build-Prozess abgeschlossen ist, können Sie das Projekt über "Run → Run As → 2 Local C/C++ Application" ausführen. Nun erscheint in der Konsolen-Ausgabe "!!!Hello Word!!!".

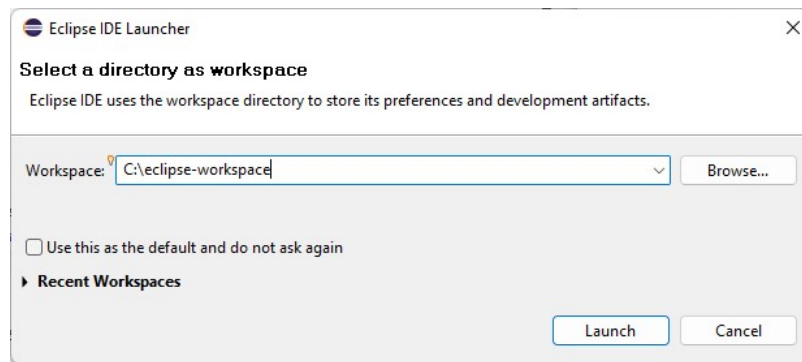


Figure 1.4: Eclipse IDE Launcher

1.3 Installation und Konfiguration von Boost

- a. Laden Sie die ZIP-Datei "boost.zip" aus Moodle herunter und entpacken Sie diese nach "C:\boost".
- b. Öffnen Sie nun Eclipse und wählen das entsprechende Projekt aus. Beachten Sie, dass Sie die Konfiguration von Boost für sämtliche im Praktikum erzeugten (Teil-) Projekte durchführen müssen, welche die grafische Ausgabe verwenden (ab Aufgabeblock 2).
- c. Wählen Sie "Project → Properties → C/C++ Build → Settings → Tool Settings → GCC C++ Compiler → Includes" aus. Fügen Sie auf der rechten Seite unter Include paths (-I) den Eintrag "C:\boost" hinzu.
- d. Wählen Sie nun auf der linken Seite der Settings den Punkt "MinGW C++ Linker → Libraries" aus. Hier fügen Sie auf der rechten Seite unter "Libraries (-l)" den Eintrag "ws2_32" hinzu. Im unteren Bereich "Library search path (L)" erstellen Sie den Eintrag "C:\boost\libs". Bestätigen Sie diese Änderungen mit dem Klick auf Apply and Close. Bestätigen Sie die aufkommende Abfrage mit "yes".

2 Linux (Ubuntu, Version 22.04)

2.1 Installation von Java

- a. Testen Sie, ob Java bereits auf Ihrem System installiert ist. Dazu öffnen Sie ein Terminal und geben Sie den Befehl `"java -version"` ein. Sollten Sie eine Fehlermeldung erhalten, müssen Sie mit Punkt *b* fortfahren. Erscheint eine Ausgabe mit der Angabe einer Java-Version, ist Java bereits auf Ihrem System installiert und es bedarf keiner weiteren Schritte.
- b. Installieren Sie Java mit dem Befehl `"sudo apt install openjdk-18-jre"`.
- c. Nun können Sie wie unter Punkt *a* beschrieben erneut testen, ob Java auf Ihrem System installiert ist.

2.2 Installation von Eclipse

- a. Installieren Sie mit dem Befehl `"sudo apt install build-essential"` alle benötigten Systemtools für Ubuntu.
- b. Downloaden und entpacken Sie den Eclipse Installer (<https://www.eclipse.org/downloads/download.php?file=/oomph/epp/2025-09/R/eclipse-inst-jre-linux64.tar.gz>).
- c. Führen Sie den entpackten Installer mit dem Befehl `"./eclipse-inst"` aus und wählen Sie Eclipse IDE for C/C++ Developers aus.
- d. Im nächsten Dialogfenster wählen Sie den Installationspfad (dieser kann unverändert bleiben) aus und bestätigen die Installation (Auswahl der Java 11+ VM bleibt unverändert).
- e. Nun ist Eclipse auf Ihrem System in dem ausgewählten Pfad installiert und kann gestartet werden. Dazu klicken Sie in dem Installer auf Launch.
- f. Beim Start von Eclipse werden Sie nach dem Ort für den Workspace gefragt. In dem Workspace speichert Eclipse alle Projekte und die dazugehörigen Metadaten ab. Wählen Sie bitte einen entsprechenden Pfad auf Ihrem System aus und bestätigen Sie die Auswahl mit "Launch".

- g. Testen Sie nun die Installation von Eclipse, indem Sie ein neues Projekt in Eclipse anlegen. Dazu wählen Sie "File → New → C/C++ Project → C++ Managed Build" aus und klicken Sie auf "Next". Geben Sie nun einen Projektnamen (z.B. "Aufgabenblock_1") ein, wählen Sie als Project type "Hello World C++ Project" und als Toolchain "Linux GCC" aus. Legen Sie das Projekt mit "Finish" an. Nun haben Sie Ihr erstes Projekt mit einem kleinen Beispielprogramm angelegt. Damit Sie dieses kleine Test-Programm ausführen können, müssen Sie es zuerst kompilieren. Dazu wählen Sie "Project → Build Project" aus. Nach dem der Build-Prozess abgeschlossen ist, können Sie das Projekt über "Run → Run As → 2 Local C/C++ Application" ausführen. Nun erscheint in der Konsolen-Ausgabe "!!!Hello Wordl!!!".

2.3 Installation und Konfiguration von Boost

- a. Führen Sie in einem Terminal den Befehl "sudo apt install libboost-all-dev".

3 MacOS

Beachten Sie, dass bei den Installationsschritten ***IHRUSERNAME*** mit Ihrem Benutzernamen auf Ihrem Mac ersetzt werden muss.

3.1 Installation Xcode Command Line Tools

- a. Öffnen Sie ein Terminal und führen Sie den Befehl "sudo xcode-select --install" aus.

3.2 Installation der Abhängigkeiten

- a. Um Eclipse und den Debugger auf Ihrem Mac zu verwenden, benötigt es ein paar Voraussetzungen. Dazu haben wir eine Skript-Datei mit dem Namen "lldb-mi.sh" in Moodle hinterlegt. Sie finden die Datei unter den Vorgabedateien im Ordner "Mac". Diese müssen Sie auf ihrem Mac in den User-Ordner (/Users/*IHRUSERNAME*/) herunterladen. Öffnen Sie nun ein Terminal und machen die Datei mit dem Befehl "sudo chmod +x /Users/*IHRUSERNAME*/lldb-mi.sh" ausführbar. Danach können Sie die Datei mit "/Users/*IHRUSERNAME*/lldb-mi.sh" ausführen. Folgen Sie den Anweisungen im Terminal. Die Batch-Datei installiert und kompiliert unter anderem den benötigten Debugger in das Verzeichnis "/Users/*IHRUSERNAME*/lldb-mi/". **Bei der Installation kann es dazu kommen, dass Sie mehrfach nach Ihrem Passwort gefragt werden.**

3.3 Installation und Konfiguration Eclipse

- a. Laden Sie den Eclipse Installer für Ihrem Mac herunter und führen Sie das Installationspaket aus:
 - a. für den Mac mit **m*-Chip** laden Sie bitte folgenden Installer herunter:
<https://www.eclipse.org/downloads/download.php?file=/oomph/epp/2025-09/R/eclipse-inst-jre-mac-aarch64.dmg>
 - b. für alle anderen Macs (mit Intel-Chip) nutzen Sie folgenden Installer:
<https://www.eclipse.org/downloads/download.php?file=/oomph/epp/2025-09/R/eclipse-inst-jre-mac64.dmg>
- b. Starten Sie den Eclipse Installer aus dem Paket und wählen Sie "Eclipse IDE for C/C++ Developers" aus.

- c. Wählen Sie in der nächsten Abfrage einen geeigneten Installationspfad (der Pfad kann unverändert bleiben) aus und bestätigen Sie die Installation (Auswahl der Java 11+ VM bleibt unverändert).
- d. Nach erfolgreicher Installation können Sie den Installer mit "Launch" beenden und Eclipse starten.
- e. Beim Start von Eclipse werden Sie nach dem Ort für den Workspace gefragt. In dem Workspace speichert Eclipse alle Projekte und die dazugehörigen Metadaten ab. Wählen Sie bitte einen entsprechenden Pfad auf Ihrem System aus und bestätigen Sie die Auswahl mit "Launch".
- f. Testen Sie nun die Installation von Eclipse, indem Sie ein neues Projekt in Eclipse anlegen. Dazu wählen Sie "File → New → Project → C/C++ Project → C++ Managed Build" aus und klicken Sie auf "Next". Geben Sie nun einen Projektnamen (z.B. "Aufgabenblock_1") ein, wählen Sie als Project type "Hello World C++ Project" und als Toolchain "MacOSX GCC" aus. Legen Sie das Projekt mit "Finish" an. Nun haben Sie Ihr erstes Projekt mit einem kleinen Beispielprogramm angelegt. Damit Sie dieses kleine Test-Programm ausführen können, müssen Sie es zuerst kompilieren. Dazu wählen Sie "Project → Build Project" aus. Nach dem der Build-Prozess abgeschlossen ist, können Sie das Projekt über "Run → Run As → 2 Local C/C++ Application" ausführen. Nun erscheint in der Konsolen-Ausgabe "!!!Hello Word!!!".
- g. Um evtl. falsche Fehlermeldungen von Eclipse auszublenden, öffnen Sie das Menü "Project → Properties → C/C++ General → Code Analysis → Use project settings" und unchecken Sie "Syntax and Semantic Errors". Beachten Sie, dass sie diese Einstellungen für jedes (Teil-)Projekt in Eclipse wiederholen müssen.

3.4 Installation und Konfiguration von Boost

- a. Laden Sie die ZIP-Datei "boost.zip" aus Moodle herunter und entpacken Sie diese nach /Users/*IHRUSERNAME*/.
- b. Öffnen Sie nun Eclipse und wählen das entsprechende Projekt aus. Beachten Sie, dass Sie die Konfiguration von Boost für sämtliche im Praktikum erzeugten (Teil-) Projekte durchführen müssen, welche die grafische Ausgabe verwenden (ab Aufgabeblock 2).
- c. Wählen Sie "Project → Properties → C/C++ Build → Settings → Tool Settings → GCC C++ Compiler → Includes" aus. Fügen Sie auf der rechten Seite unter "Include paths (-I)" den Eintrag "/Users/*IHRUSERNAME*/boost" hinzu.
- d. Wählen Sie nun auf der linken Seite den Settings den Punkt "MacOS X C++ Linker → Libraries" aus. Hier fügen Sie auf der rechten Seite unter "Library search

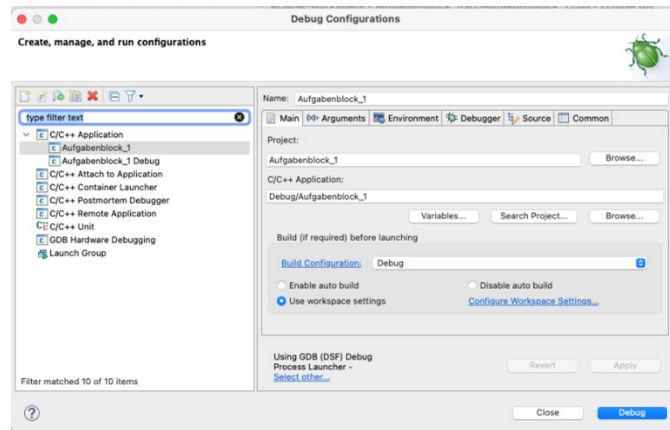


Figure 3.1: Eclipse Debug Configuration

path (L)" den Eintrag `/Users/*IHRUSERNAME*/boost/libs` hinzu. Bestätigen Sie diese Änderungen mit dem Klick auf `"Apply and Close"`. Bestätigen Sie die aufkommende Abfrage mit `"yes"`.

3.5 Debugger Konfiguration

- a. Öffnen Sie Eclipse und öffnen Sie über das Kontextmenü `"Help → Install New Software"` den Paket-Manager von Eclipse. Hier wählen Sie bei `"Work with"` `"CDT ..."` aus. Markieren Sie in der Auswahl unter `"CDT Optional Features"` das Paket `"C/C++ LLDB Debugger Integration (experimental)"` aus und bestätigen die Installation mit `"Next"`, `"Finish"` und `"Restart Now"`.
- b. Nachdem Eclipse neu gestartet wurde, gehen Sie in das Menü `"Run → Debug Configurations..."`. In dem neu geöffneten Fenster wählen Sie, wie in Figure 3.1 zu sehen, `"Select other..."`. Dort wählen Sie den Haken bei `"Use configuration specific settings"` und wählen im unteren Bereich `"LLDB-MI Debug Process Launcher"` aus. Bestätigen Sie die Auswahl mit `"OK"`. (s. Figure 3.1) Wählen Sie nun den Menüreiter `"Debugger"` aus und klicken neben der `"LLDB command"-Auswahl` auf `"Browse..."`. (s. Figure 3.2) Hier wählen Sie nun die Datei `"lldb-mi"` aus dem Ordner `/Users/*IHRUSERNAME*/lldb-mi` aus und bestätigen die Auswahl mit `"Open"`. Nun Bestätigen Sie noch mit `"Apply"` und können dann das Debugging mit `"Debug"` starten. Beim Starten des Debuggers kann es zu der Abfrage von Benutzername und Passwort Ihres Macs kommen. **Beachten Sie, dass sie diese Einstellungen für jedes (Teil-)Projekt in Eclipse wiederholen müssen, damit das Debugging verwendet werden kann.**

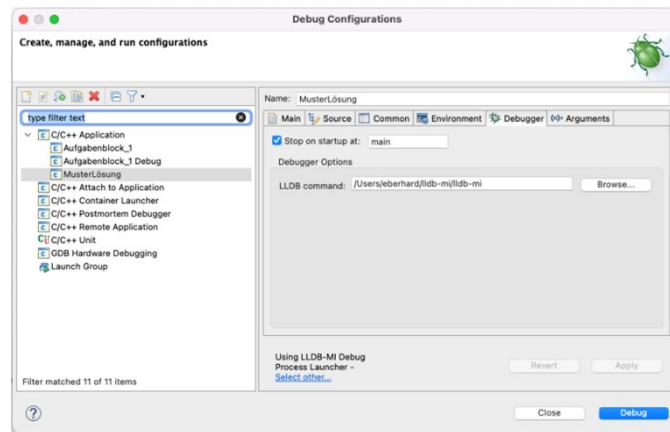


Figure 3.2: Eclipse Debug Configuration 2