

**Projekt:**

**Kniffel**

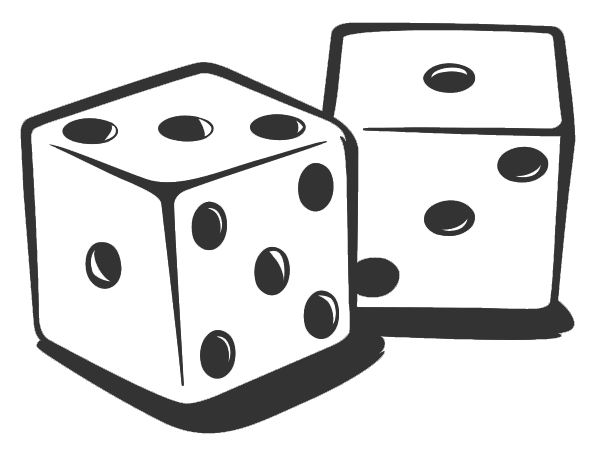
**Internationaler Frauenstudiengang für Informatik, B.A.**

**der**

**Hochschule Bremen**

**Modul: C++**

**Sommersemester 2014**

****

|  |  |
| --- | --- |
| Themensteller: | Prof. Dr. Ulrich Breymann |
| Abgabetermin: | 25.Juni 2014 |
| Vorgelegt von: | Ute Albersmeier  Irina Fress  Frieda Hildebrand  Larisa Staroverova |

Inhaltsverzeichnis

[1. Beschreibung der zu lösenden Aufgabe 3](#_Toc390072199)

[2. Architekturbeschreibung 4](#_Toc390072200)

[2. 1. UML-Diagramm 4](#_Toc390072201)

[2. 2. Ablaufdiagramm 5](#_Toc390072202)

[2. 3. Details der Lösung 7](#_Toc390072203)

[3. Technische Voraussetzungen 8](#_Toc390072204)

[3.1. GoogleTest (Google C++ Testing Framework) unter Eclipse anbinden 8](#_Toc390072205)

[4. Anmerkung 14](#_Toc390072206)

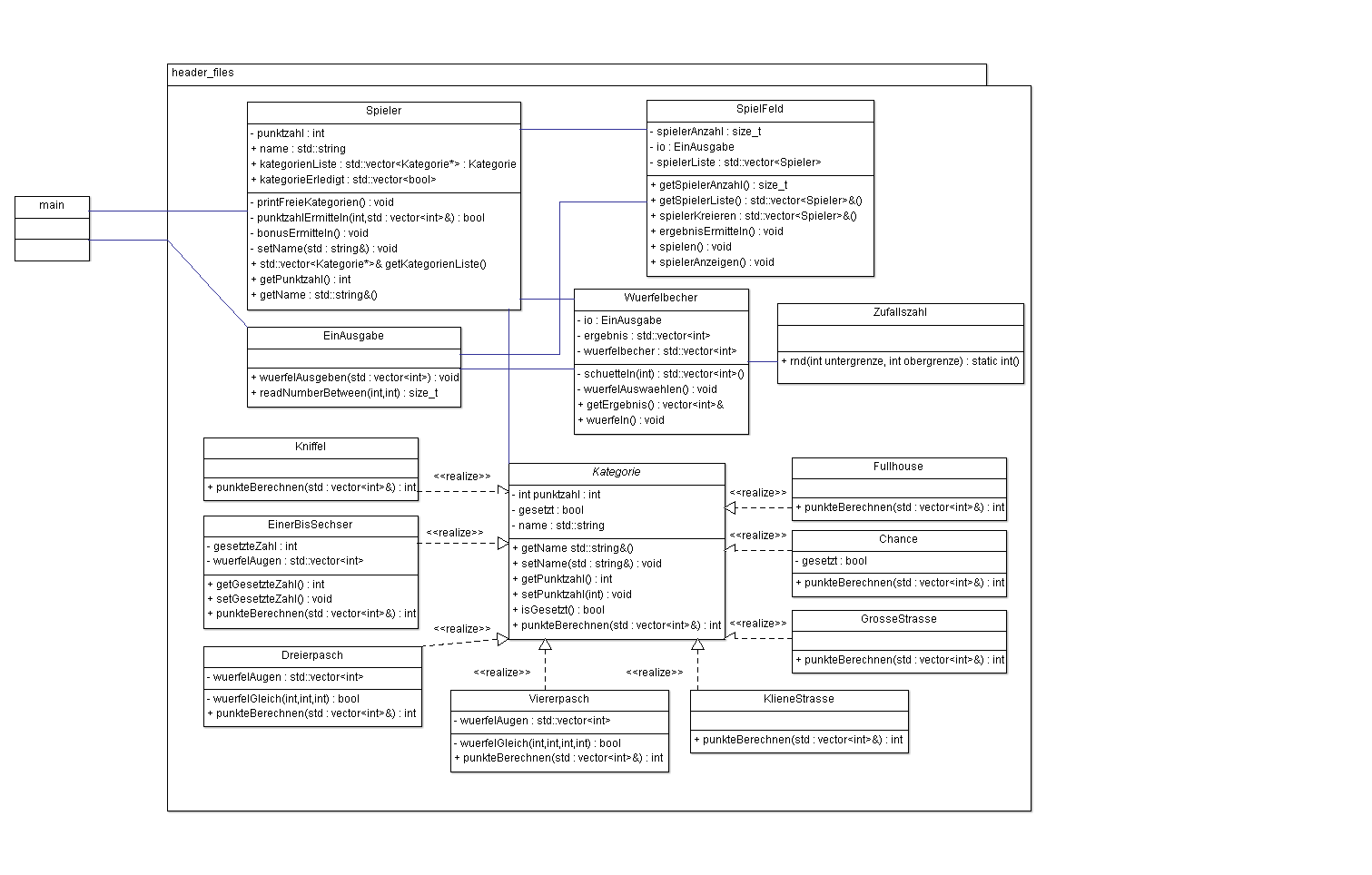
[5. Downloads 17](#_Toc390072207)

# Beschreibung der zu lösenden Aufgabe

* Programm starten
* Programm fragt nach Anzahl der Spieler (die minimale Spieleranzahl ist eins, die maximale Spieleranzahl ist fünf)
* Programm bittet Spieler den Namen einzugeben
* Programm dauer 13 Runden (hier beginnt das eigentliche Spiel):
* Jede Runde wird für jeden Spieler wiederholt:
  + - Zuerst wird eine nummerierte Kategorienliste ausgegeben, damit der Spieler eine Übersicht hat, welche Kategorien belegt werden müssen
    - Insgesamt gibt es 5 Würfel, die in jeder Runde maximal 3 mal geworfen werden können
    - Bei dem ersten Wurf werden alle 5 Würfel geworfen und ausgegeben
    - Danach wird der Spieler gefragt, ob er nochmal würfeln möchte:
      * + Falls „Ja“, wird der Spieler gefragt, wie viele Würfel geworfen werden sollen und welche zurück in den Becher kommen
        + Falls „Nein“, wird der Spieler gefragt, welche Kategorie belegt werden soll
    - Punkte werden im Hintergrund berechnet und in die Kategorienliste eingetragen
    - Die ausgewählte Kategorie wird für die nächsten Runden blockiert
    - Dann ist der nächste Spieler dran
* Im Laufe des Spiels werden die Punkte für jeden Spieler berechnet.
* Es wird außerdem ermittelt, ob ein Spieler Bonuspunkte erhält
* Eventuelle Bonuspunkte werden dazu addiert
* Nachdem alle 13 Runden vorbei sind, werden die Platzierungen ermittelt
* Die Platzierungen mit der jeweiligen Punktezahl werden untereinander absteigend ausgegeben
* Programm endet

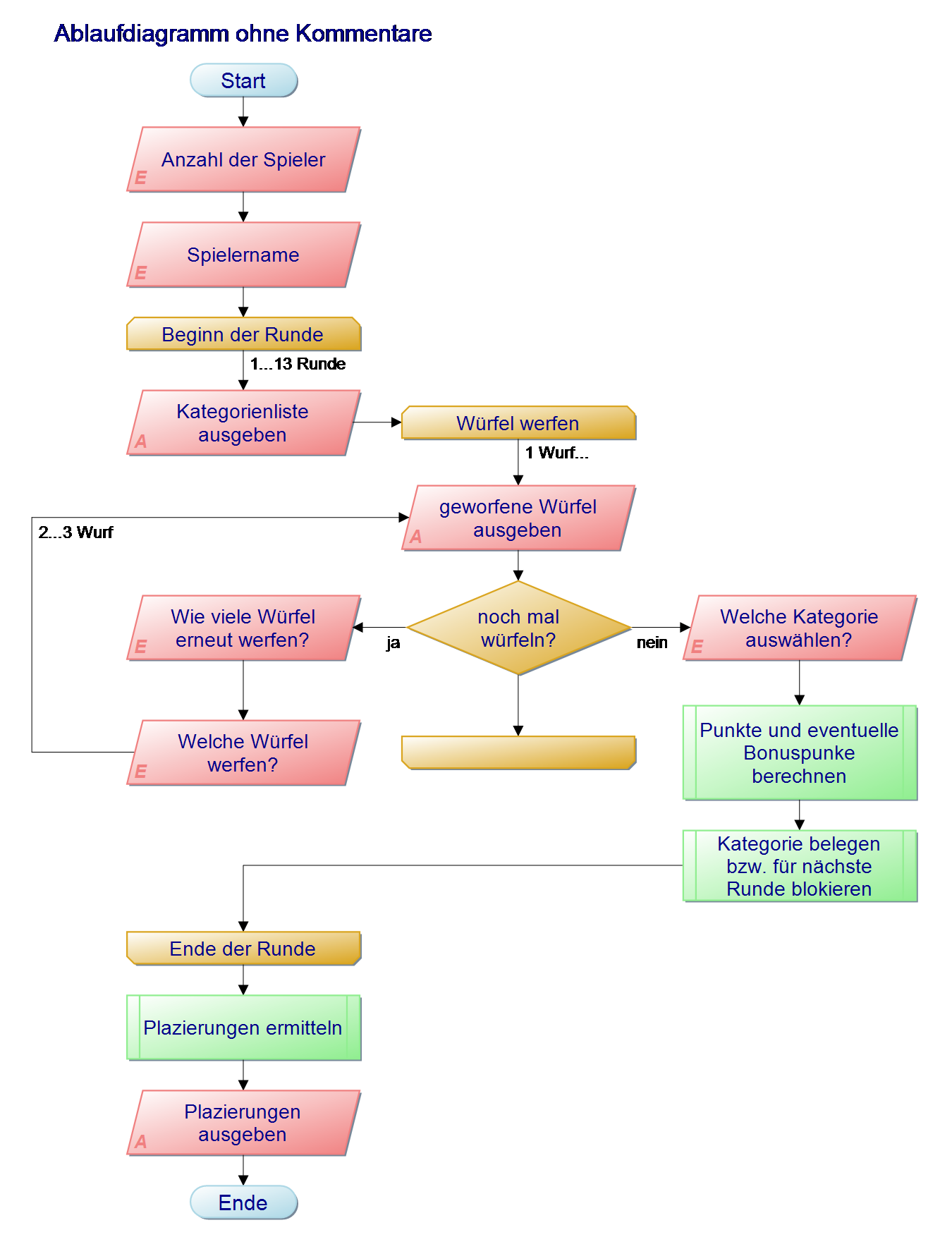
## Architekturbeschreibung

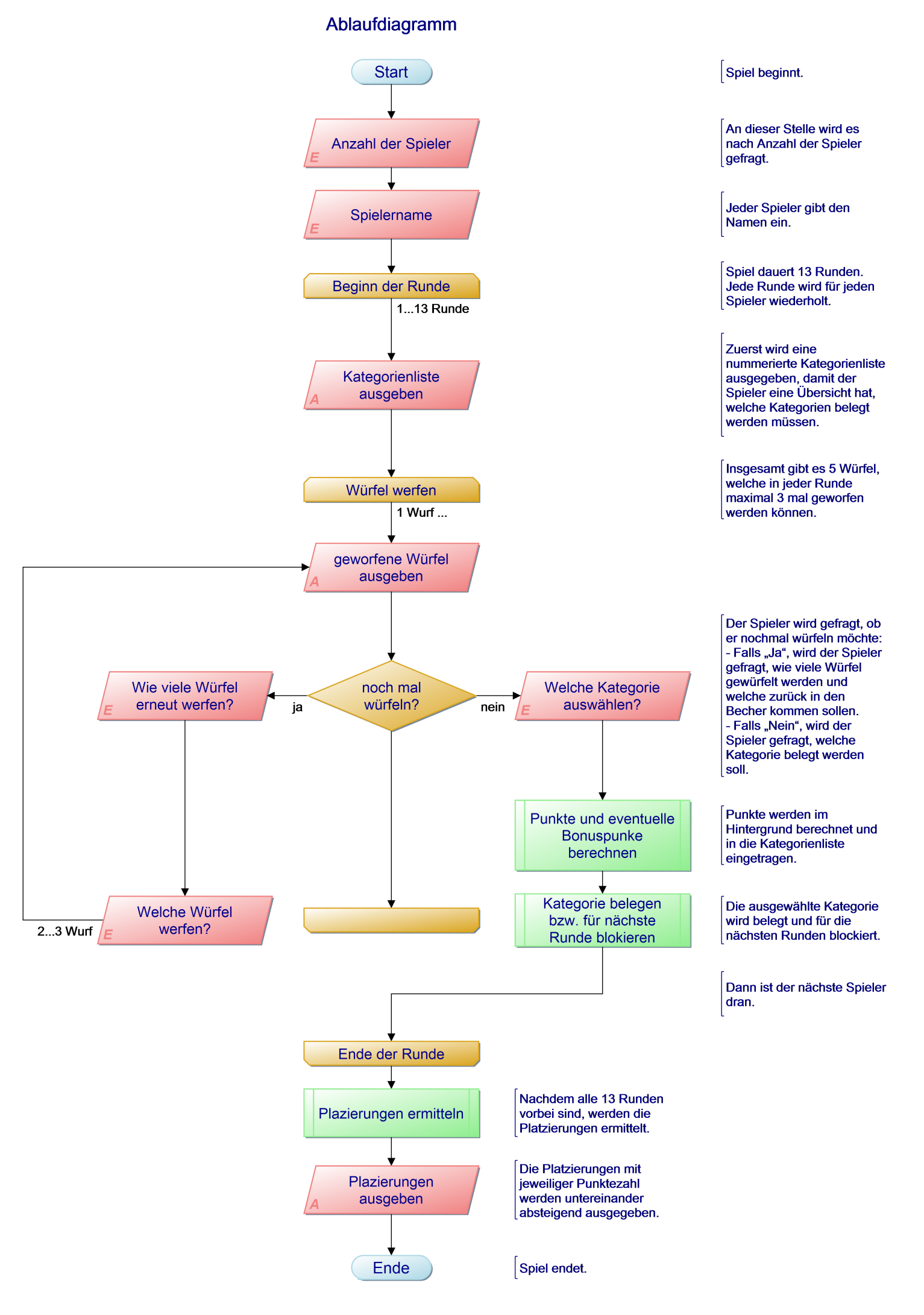
## 2. 1. UML-Diagramm



**Abbildung 1: UML-Diagramm**

## 2. 2. Ablaufdiagramm

****



**Abbildung 2: Ablaufdiagramm**

## 2. 3. Details der Lösung

|  |  |
| --- | --- |
| C:\Users\Larisa\Dropbox\Screenshots\Screenshot 2014-06-09 06.58.06.png  **Abbildung 3: Details der Lösung unter Eclipse** | **Abbildung 4: Details der Lösung unter NetBeans** |

## Technische Voraussetzungen

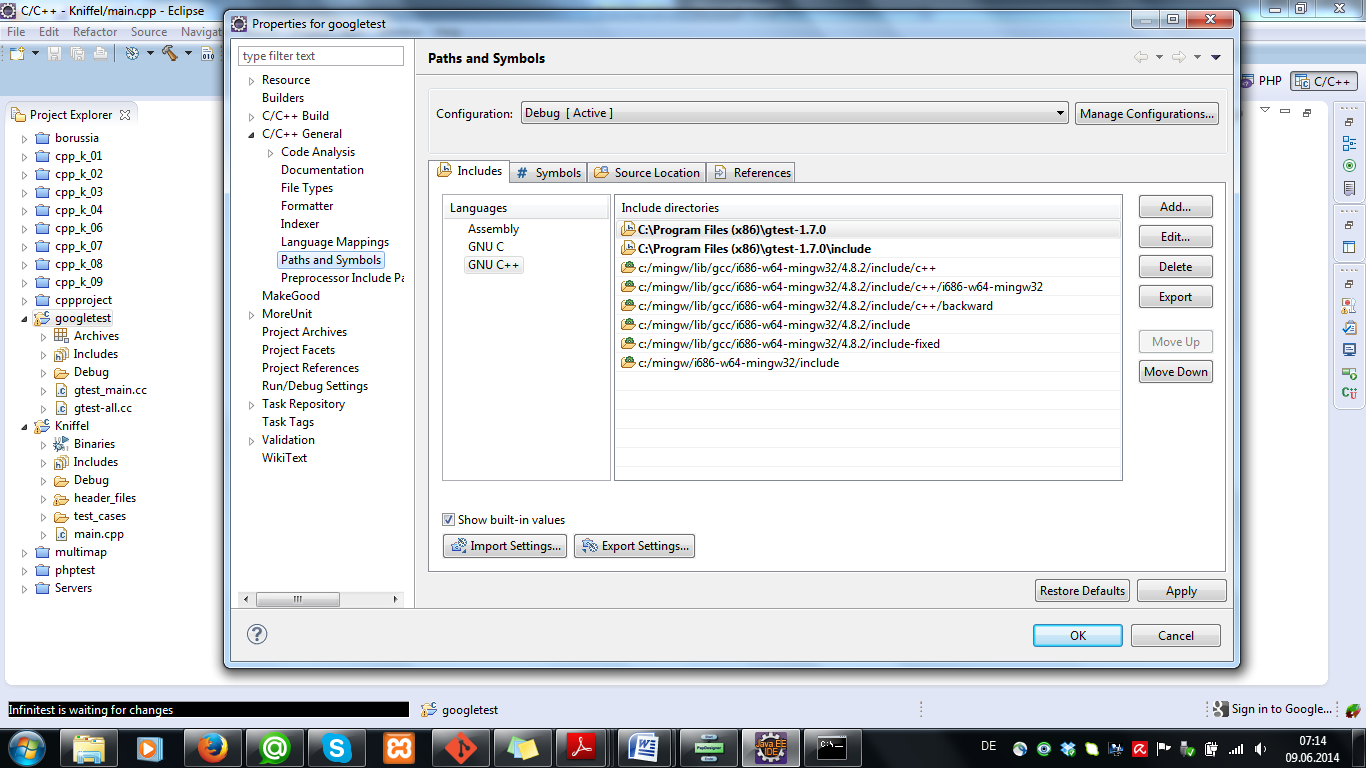
Für die Ausführung und Benutzung des Programms sind folgende Voraussetzungen zu empfehlen:

* Microsoft Windows, MacOS X, Linux Betriebssystem;
* Eclipse (Juno) IDE for C/C++ Developers;
* NetBeans IDE 8.0;
* GoogleTest (Google C++ Testing Framework);

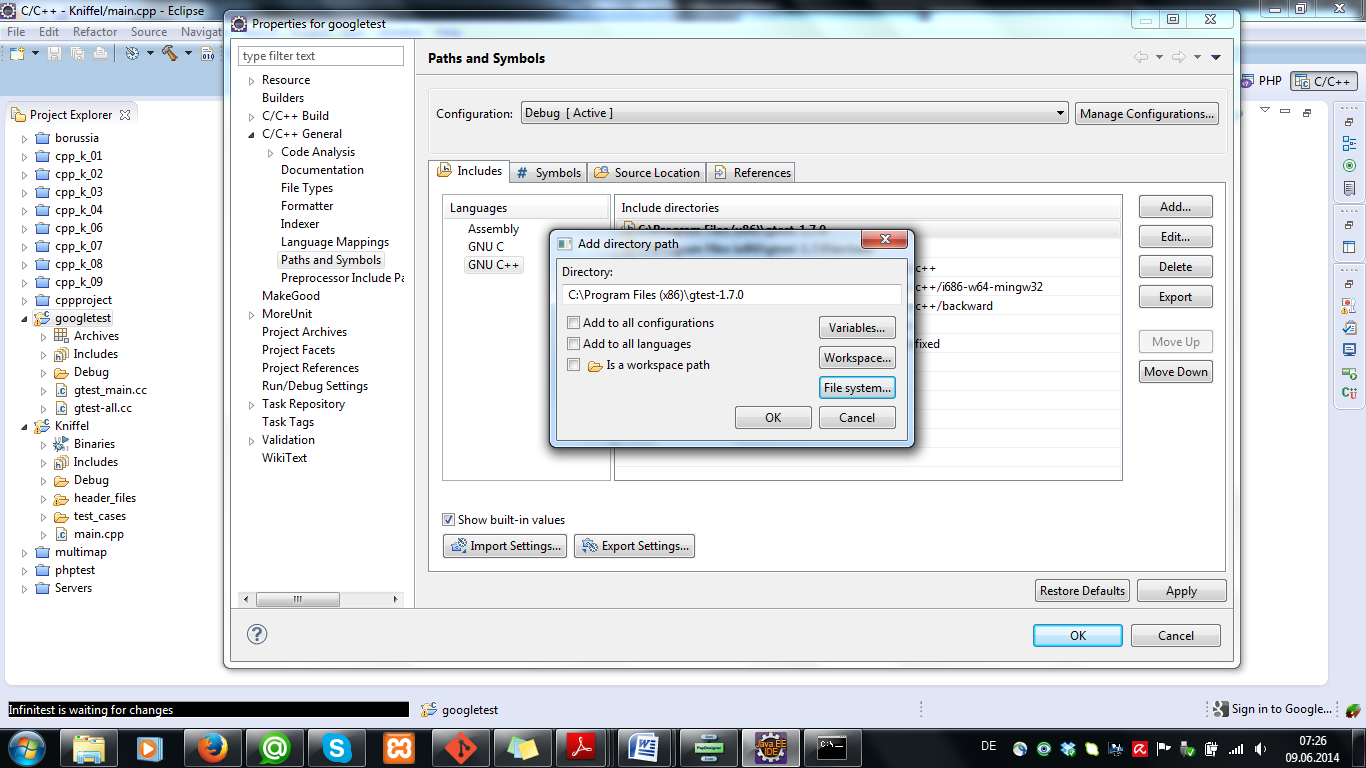
## GoogleTest (Google C++ Testing Framework) unter Eclipse anbinden

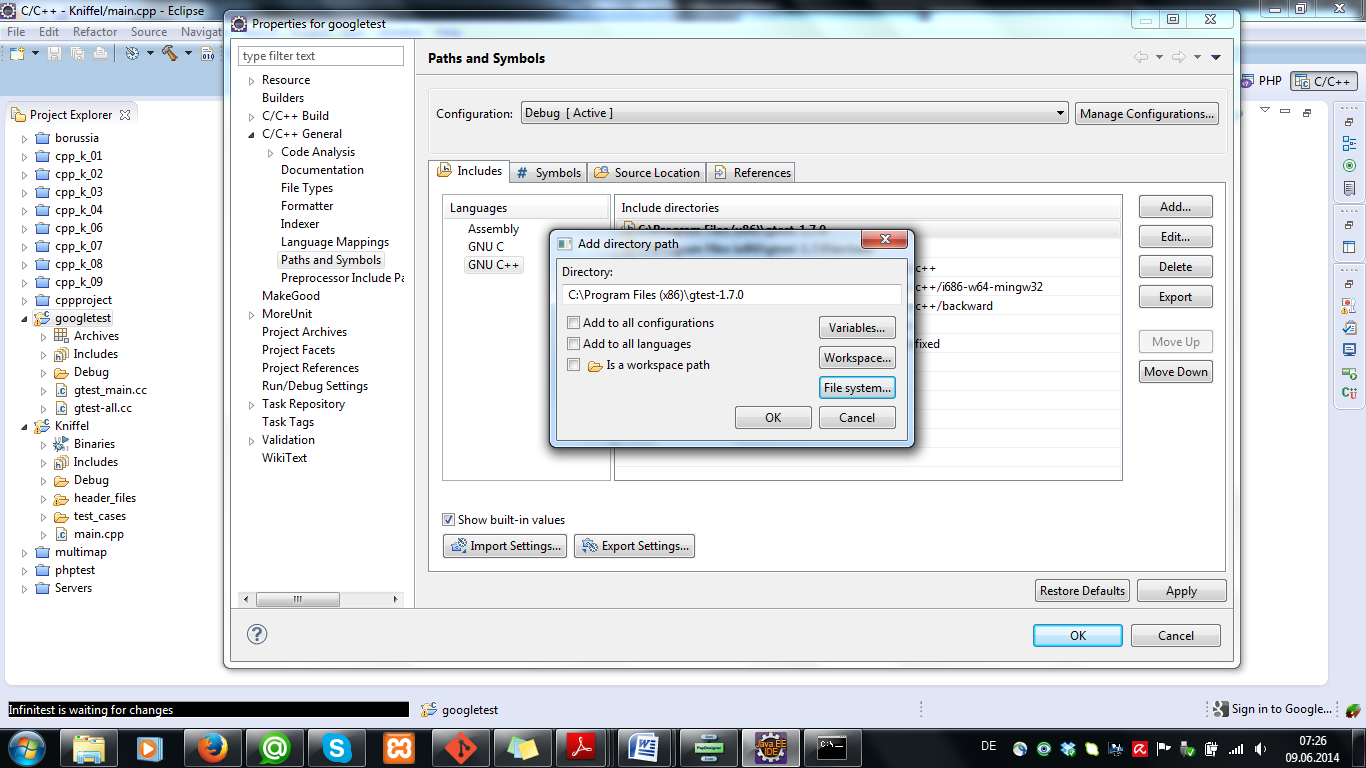
Um GoogleTest (Google C++ Testing Framework) unter Eclipse anzubinden, laden Sie zuerst eine gtest-1.7.0.zip Datei herunter und extrahieren Sie diese Datei im Verzeichnis Ihrer Wahl.

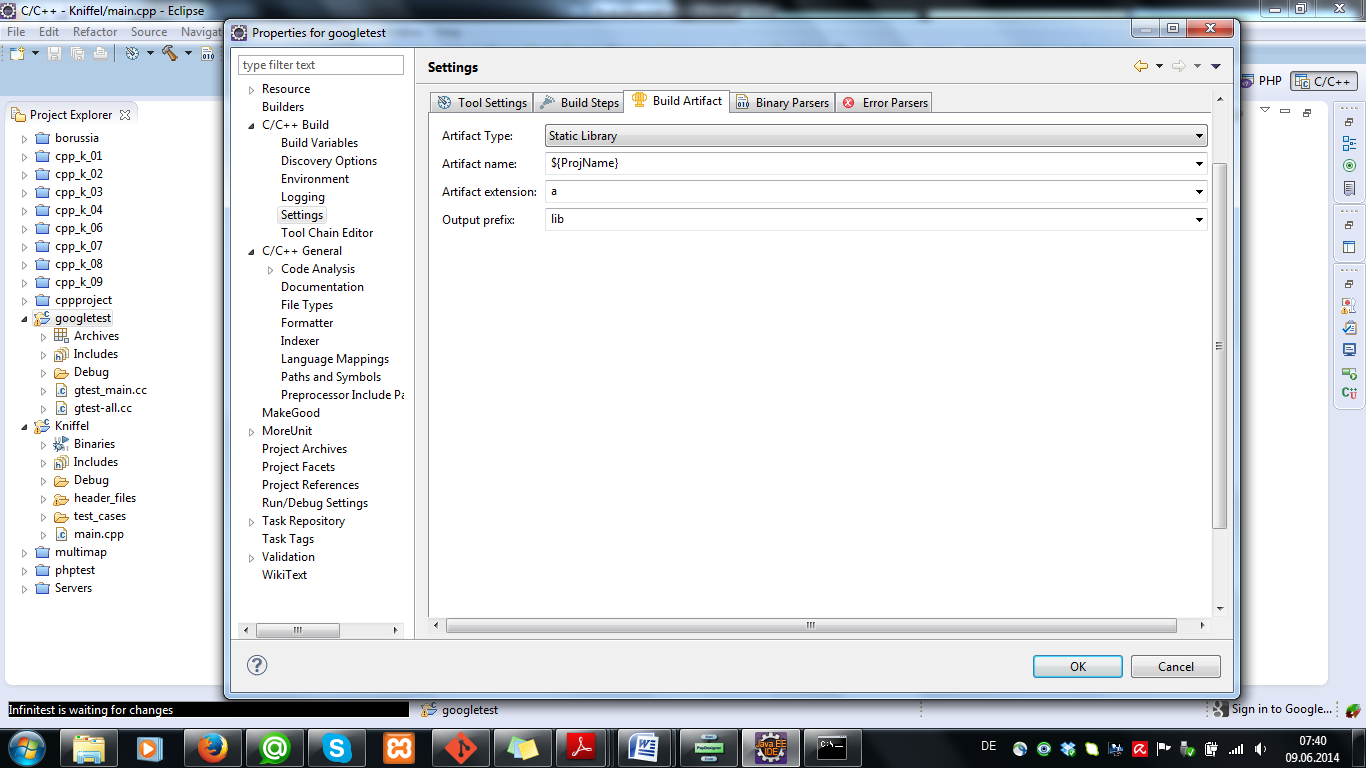
Im nächsten Schritt müssen Sie ein Projekt „**googletest**“ in Eclipse anlegen. Dieses Projekt wird später als eine statische Bibliothek dienen, welche ans „**Kniffel**“ Projekt angebunden wird.



Klicken Sie mit der rechten Maustaste auf das Projekt „**googletest**“. Wählen Sie „**Properties**“ aus. Nachdem Sie „**Properties**“ ausgewählt haben, erscheint das „**Properties for googletest**“ Fenster. Wählen Sie in diesem Fenster **C/C++ General 🡪 Paths and Symbols**. Danach klicken Sie auf den „**Includes**“ Tab und wählen Sie unter Languages „**GNU C++**“ aus. Auf der rechten Seite finden Sie den Button „**Add**“, klicken Sie darauf.

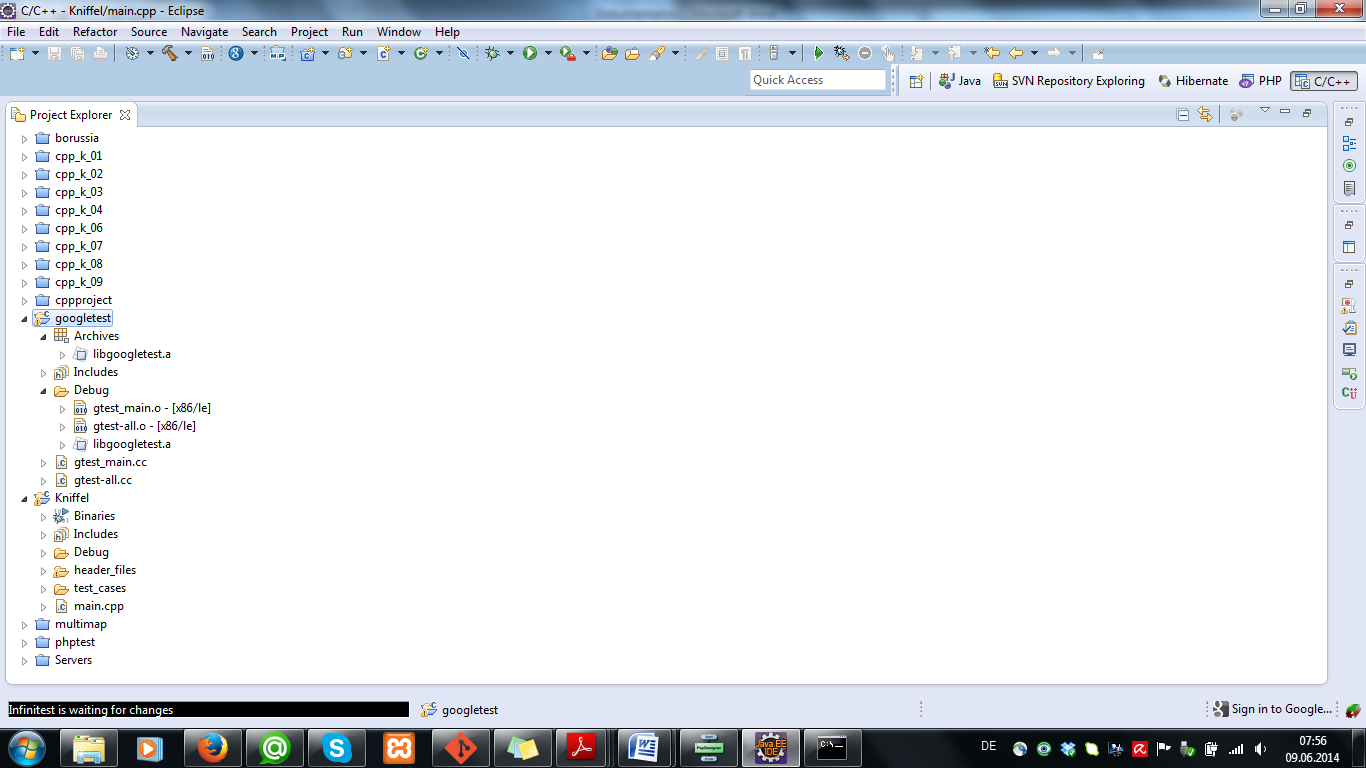
Es erscheint ein „**Add directory path**“ Fenster. Klicken Sie auf den Button „**File system**“. Geben Sie den genauen Pfad zum Verzeichnis an, wo Sie **gtest-1.7.0** extrahiert haben. Nachdem Sie das **gtest-1.7.0** Verzeichnis in Ihrem Dateisystem gefunden haben, bestätigen Sie mit dem Klick auf „**Ok**“.

Im nächsten Schritt inkludieren Sie **gtest-1.7.0\include**. Klicken Sie nochmal auf den Button „**Add**“, dann auf „**File system**“. Geben Sie den genauen Pfad zum Verzeichnis **gtest-1.7.0\include** an und bestätigen Sie mit dem Klick auf „**Ok**“.

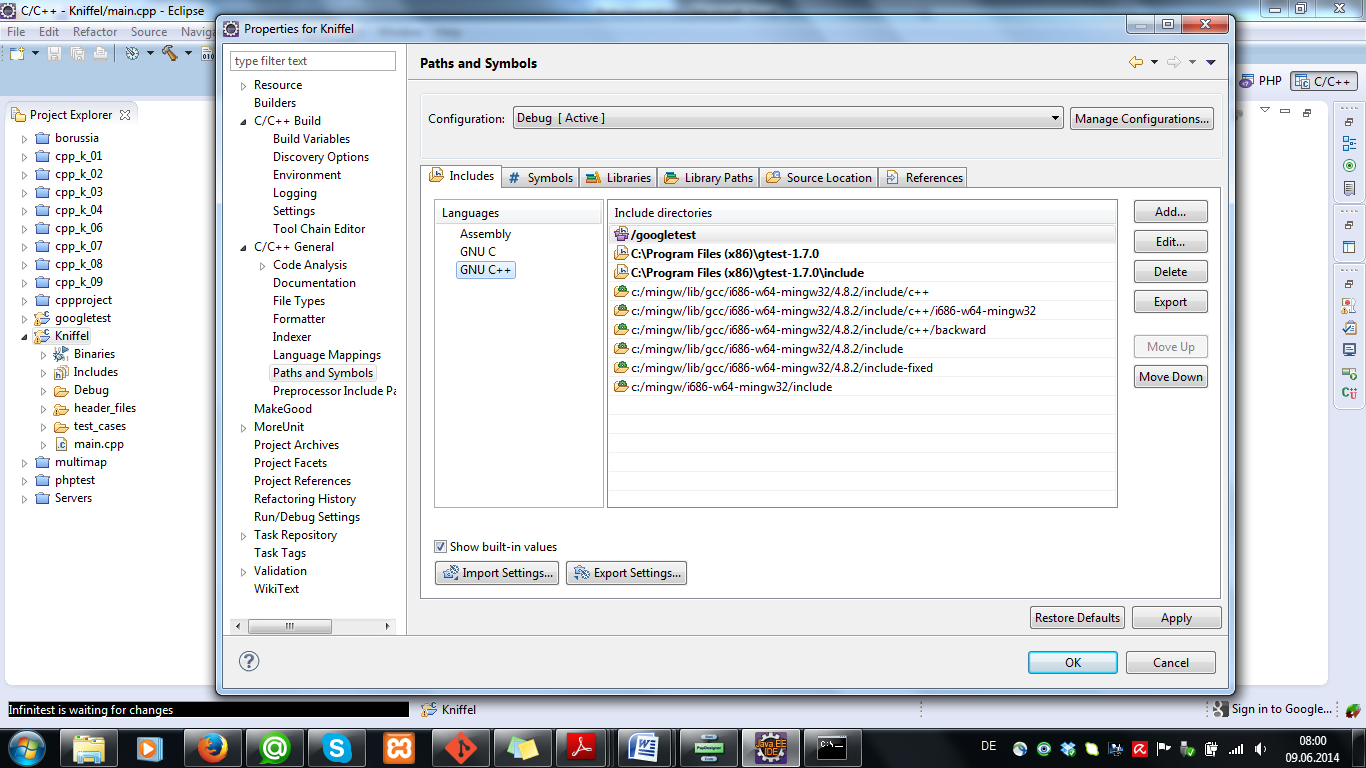


Nachdem Sie **gtest-1.7.0** und **gtest-1.7.0\include** inkludiert haben, wählen Sie auf der linken Seite **C/C++ Build 🡪 Settings** aus. Auf der rechten Seite suchen Sie nach dem Tab „**Build Artifact**“. Klicken Sie darauf. Unter „**Artifact Type**“ wählen Sie aus der Dropdown - Liste „**Static Library**“. Bestätigen Sie mit dem Klick auf „**Ok**“.

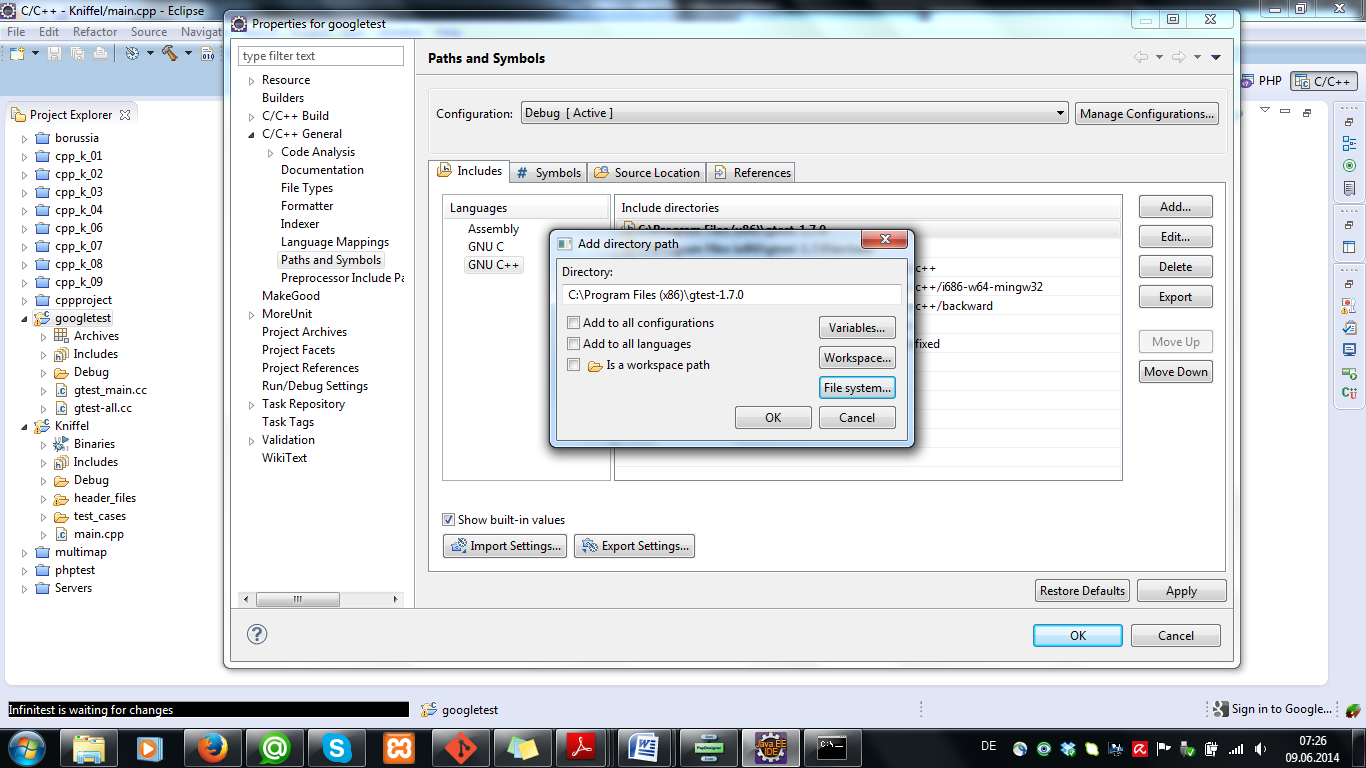
Nachdem Sie alle Schritte abgearbeitet haben, müssen Sie das Projekt „**googletest**“ erneut kompilieren („**Clean and Build**“).

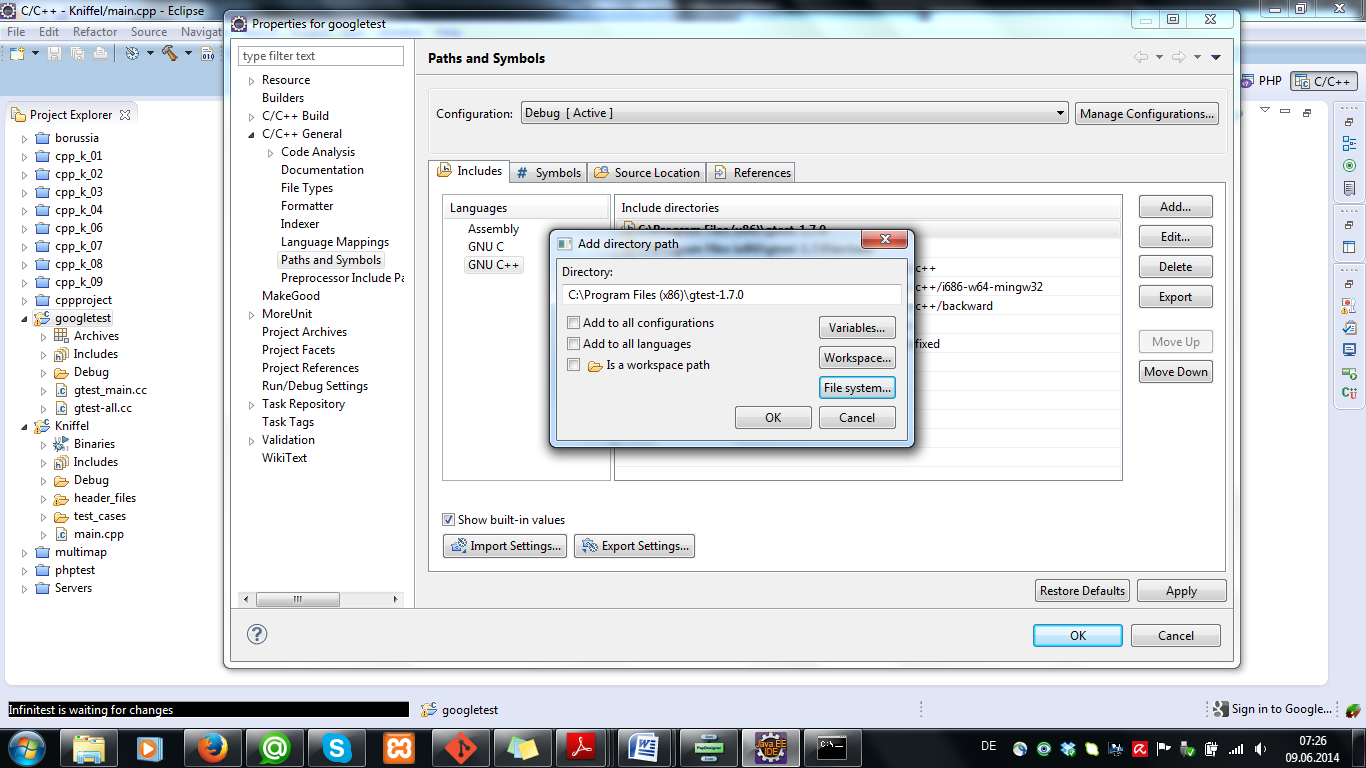


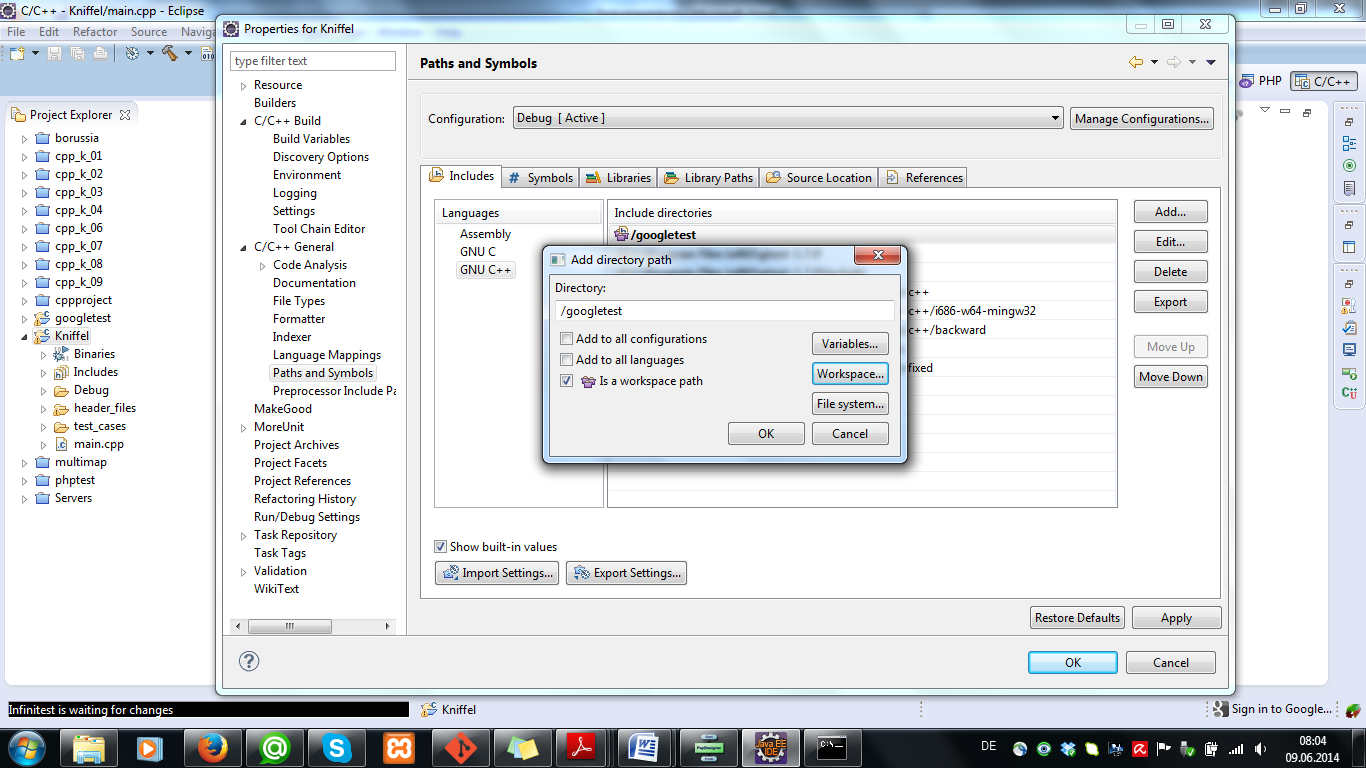
Unter „**Debug**“ können Sie **libgoogletest.a** finden. Jetzt müssen Sie diese statische Bibliothek ans „**Kniffel**“ Projekt anbinden. Um dies machen zu können, befolgen Sie die vorliegenden Anweisungen:



Klicken Sie mit der rechten Maustaste auf das Projekt „**Kniffel**“. Wählen Sie „**Properties**“ aus. Nachdem Sie „**Properties**“ ausgewählt haben, erscheint das „**Properties for Kniffel**“ Fenster. Wählen Sie in diesem Fenster **C/C++ General 🡪 Paths and Symbols** aus. Danach klicken Sie auf den „**Includes**“ Tab und wählen Sie unter Languages „**GNU C++**“ aus. Auf der rechten Seite finden Sie den Button „**Add**“, klicken Sie darauf.

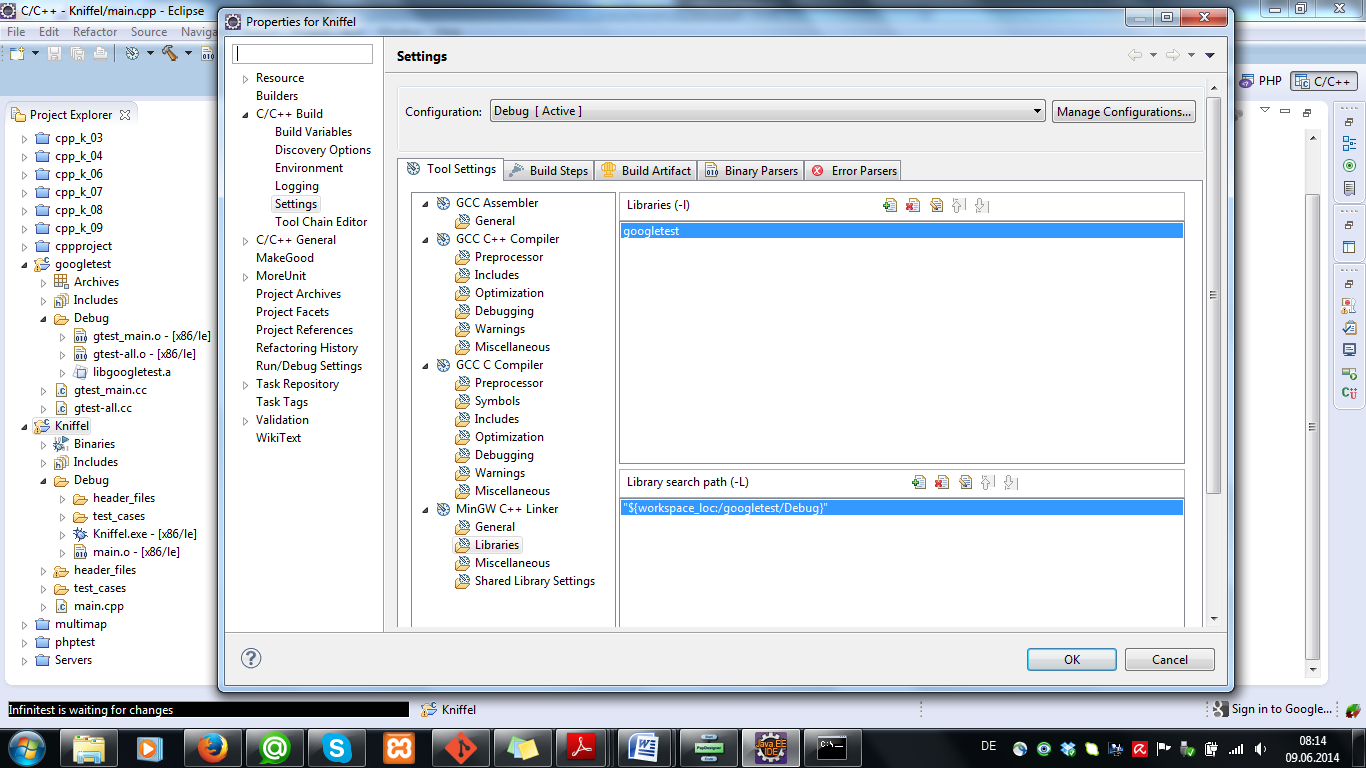
Dann klicken Sie auf den Button „**File system**“ und geben den genauen Pfad zum Verzeichnis an, wo Sie **gtest-1.7.0** extrahiert haben. Nachdem Sie das **gtest-1.7.0** Verzeichnis in Ihrem Dateisystem gefunden haben, bestätigen Sie mit dem Klick auf „**Ok**“.

Im nächsten Schritt inkludieren Sie **gtest-1.7.0\include**. Klicken Sie nochmal auf den Button „**Add**“, dann auf „**File system**“. Geben Sie den genauen Pfad zum Verzeichnis **gtest-1.7.0\include** und bestätigen Sie mit dem Klick auf „**Ok**“.

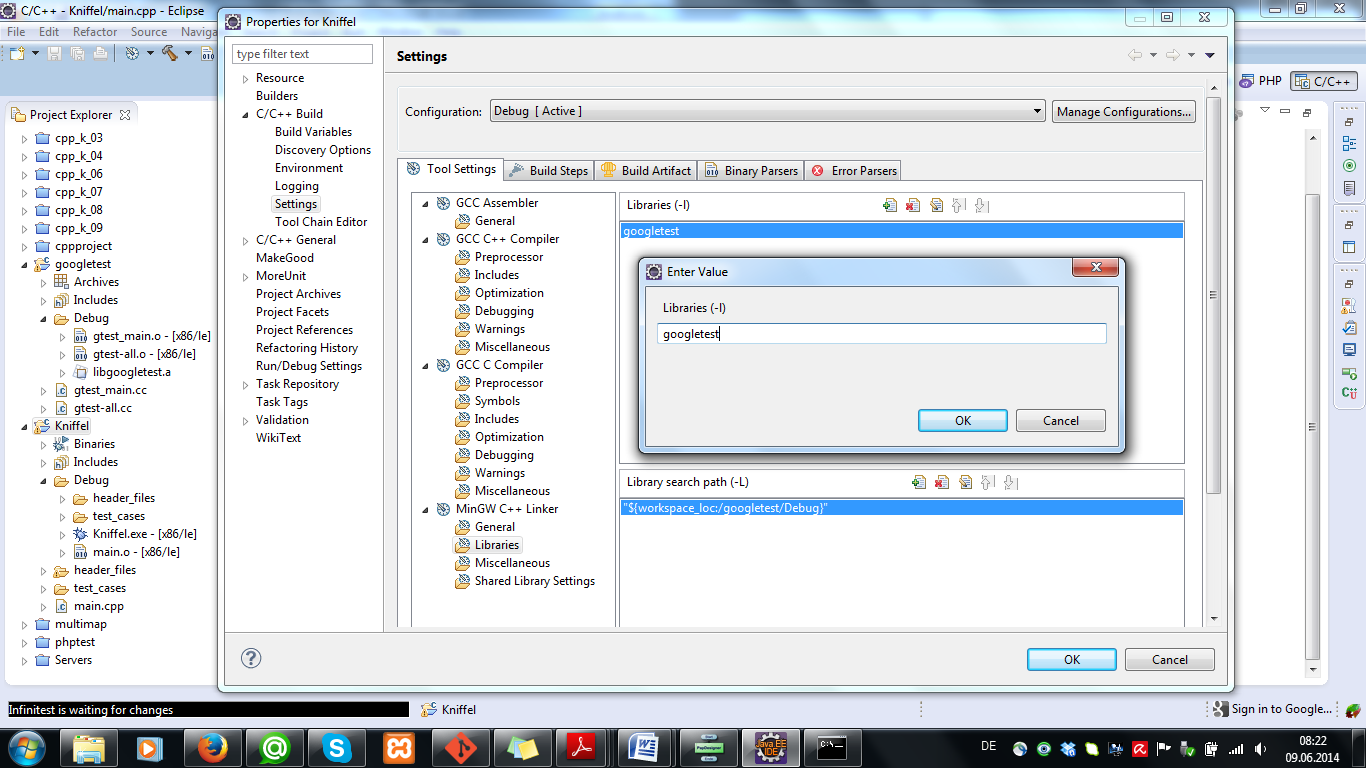


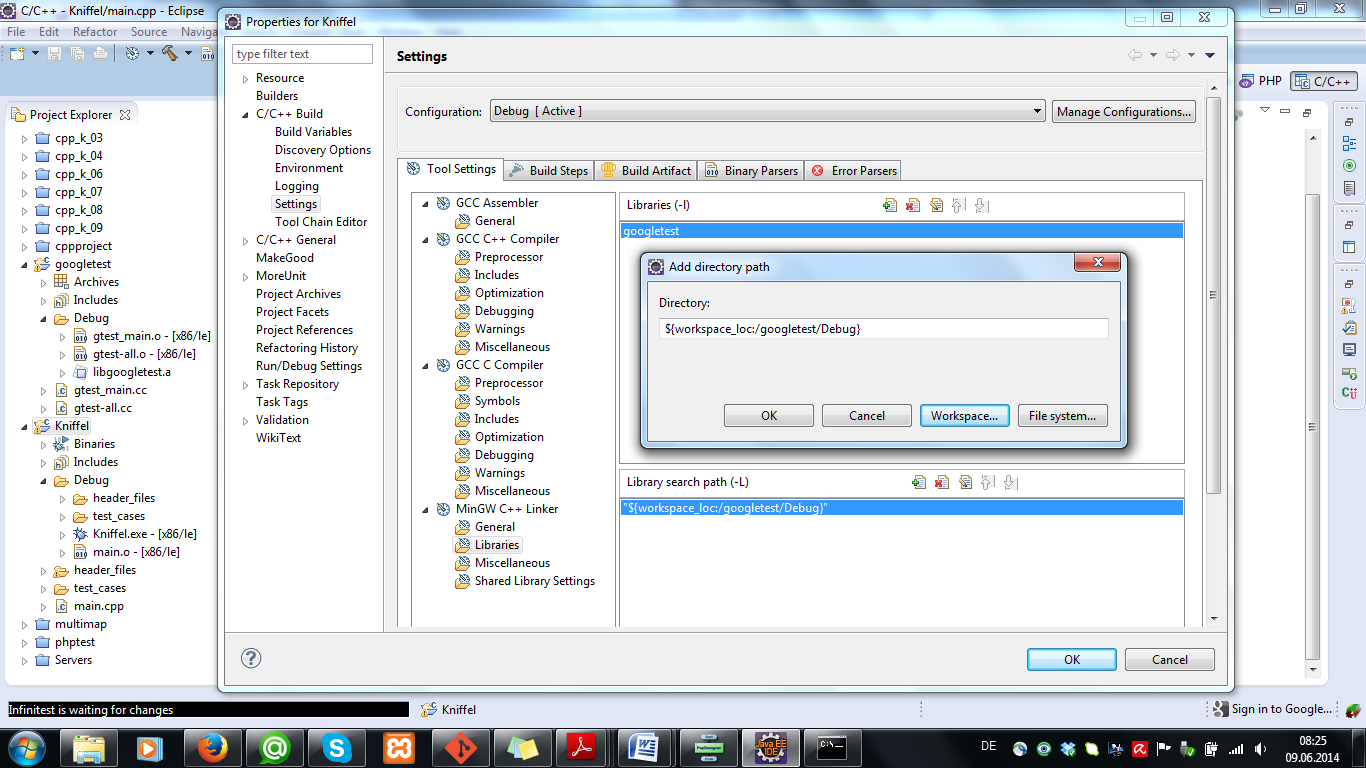
Jetzt inkludieren Sie die statische Bibliothek „**googletest**“. Klicken Sie nochmal auf den Button „**Add**“, dann auf „**Workspace**“. Wählen Sie „**googletest**“ Projekt aus und bestätigen Sie mit dem Klick auf „**Ok**“.

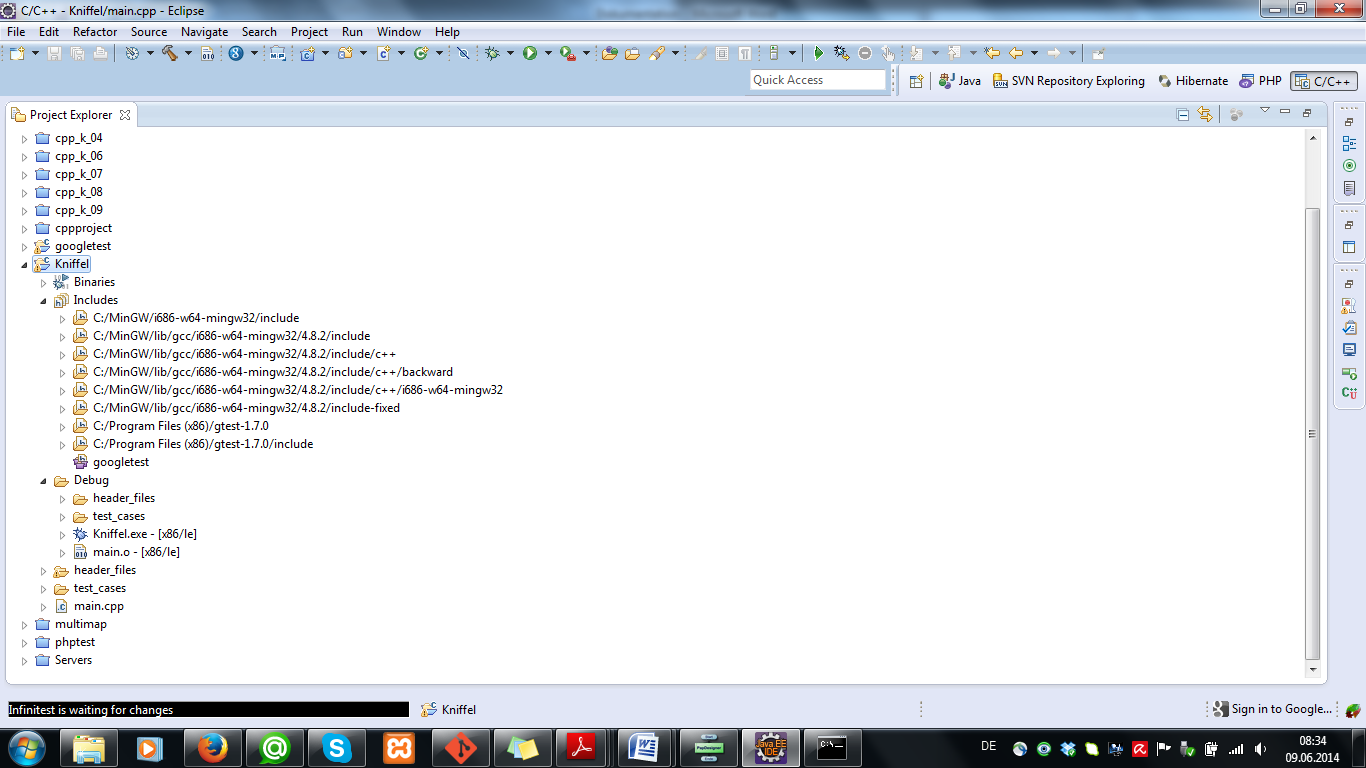
Dann klicken Sie auf „**Apply**“ und nochmal auf „**Ok**“

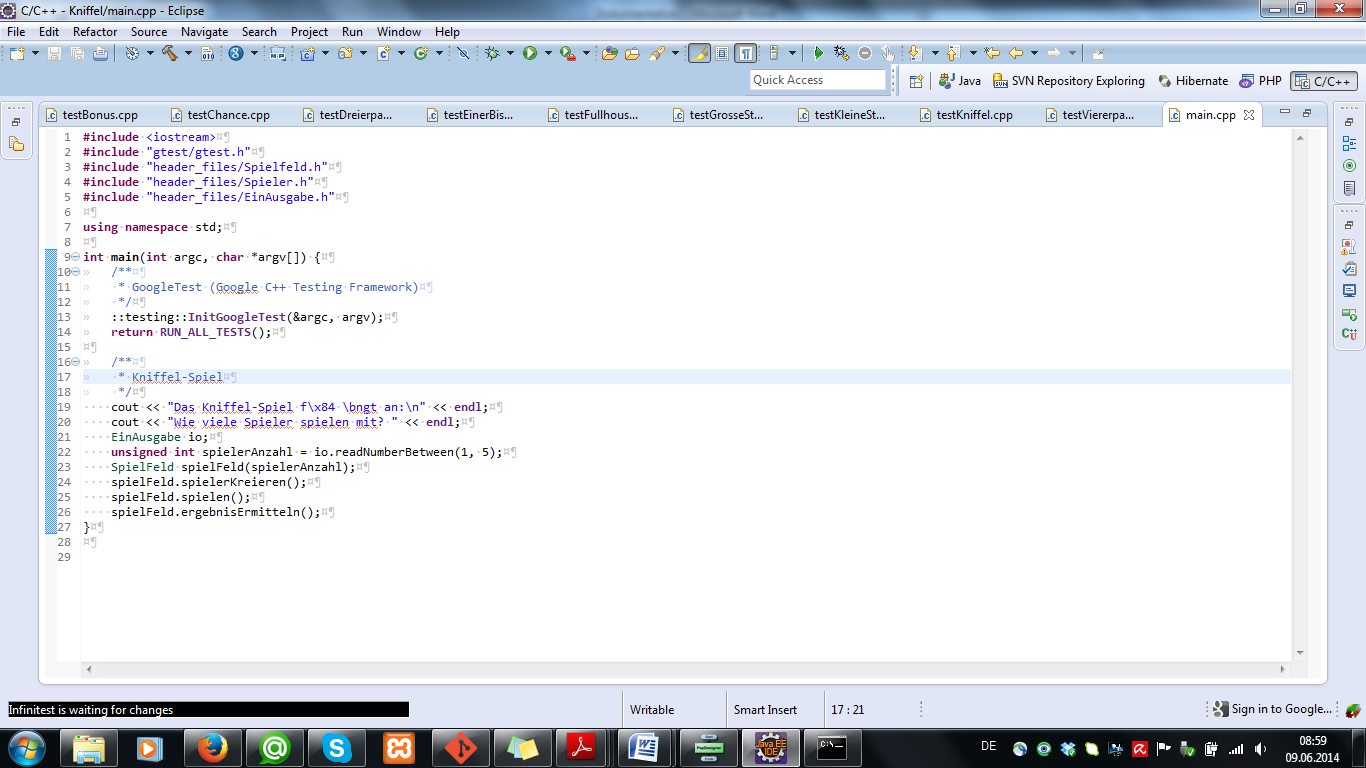


Nachdem Sie **gtest-1.7.0**, **gtest-1.7.0\include** und „**googletest**“ Projekt inkludiert haben, müssen Sie nun den Linker darauf hinweisen, dass er eine statische Bibliothek anbinden soll. Auf der linken Seite klicken Sie auf **C/C++ Build 🡪 Settings**. Danach klicken Sie auf den Tab „**Tool Settings**“. Wählen Sie **MinGW C++ Linker 🡪 Libraries** aus. Im Fenster „**Libraries (-l)**“ klicken Sie auf das Plus-Symbol.

Es erscheint ein „**Enter** **Value**“ Fenster. Geben Sie „**googletest**“ und bestätigen Sie mit dem Klick auf „**Ok**“.

Im unteren Bereich des „**Properties for Kniffel**“ Fenster finden Sie „**Library search path (-L)**“. Klicken Sie auf das Plus-Symbol. Es erscheint ein „**Add directory path**“ Fenster. Klicken Sie auf „**Workspace…**“ und geben Sie den genauen Path zur statischen Bibliothek ein, welche im **googletest/Debug** Verzeichnis zu finden ist. Nachdem Sie alle diese Schritte getan haben, müssen Sie das „**Kniffel**“ Projekt nochmal kompilieren „**Clean and Build**“.

Überprüfen Sie, ob Ihr „**Kniffel**“ Projekt ungefähr genauso aussieht. Im Verzeichnis „**Includes**“ müssen **gtest-1.7.0**, **gtest-1.7.0/inlude** und statische Bibliothek **googletest** zu finden sein. Falls dies der Falls ist, haben Sie erfolgreich GoogleTest an Eclipse angebunden.



Der letzte Hinweis: vergessen Sie nicht in main.cpp „**gtest/gtest.h**“ inkludieren. Um alle Tests zum Laufen zu bringen, geben Sie folgendes ein:

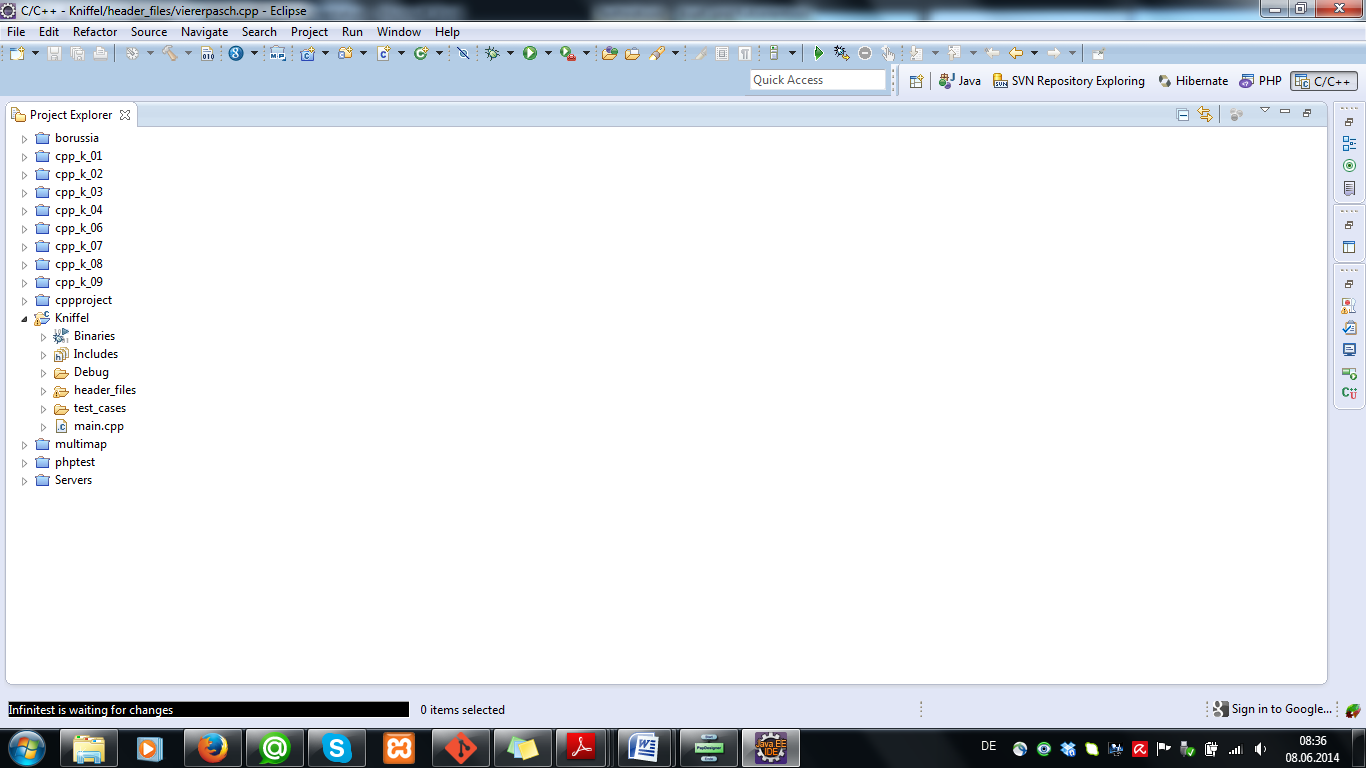
::testing::**InitGoogleTest**(&argc, argv);

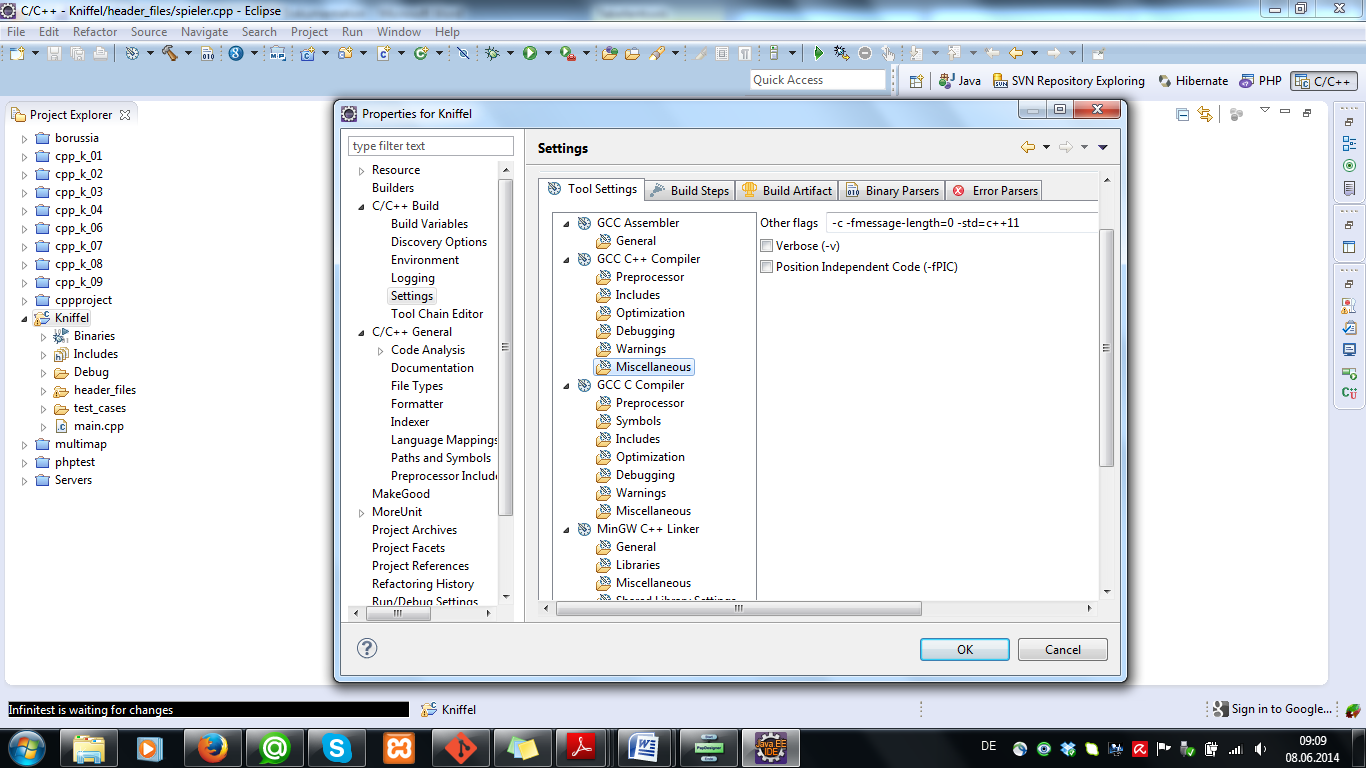
**return** **RUN\_ALL\_TESTS**();

In der Konsole können Sie sehen, welche Tests erfolgreich waren und welche fehlgeschlagen sind.

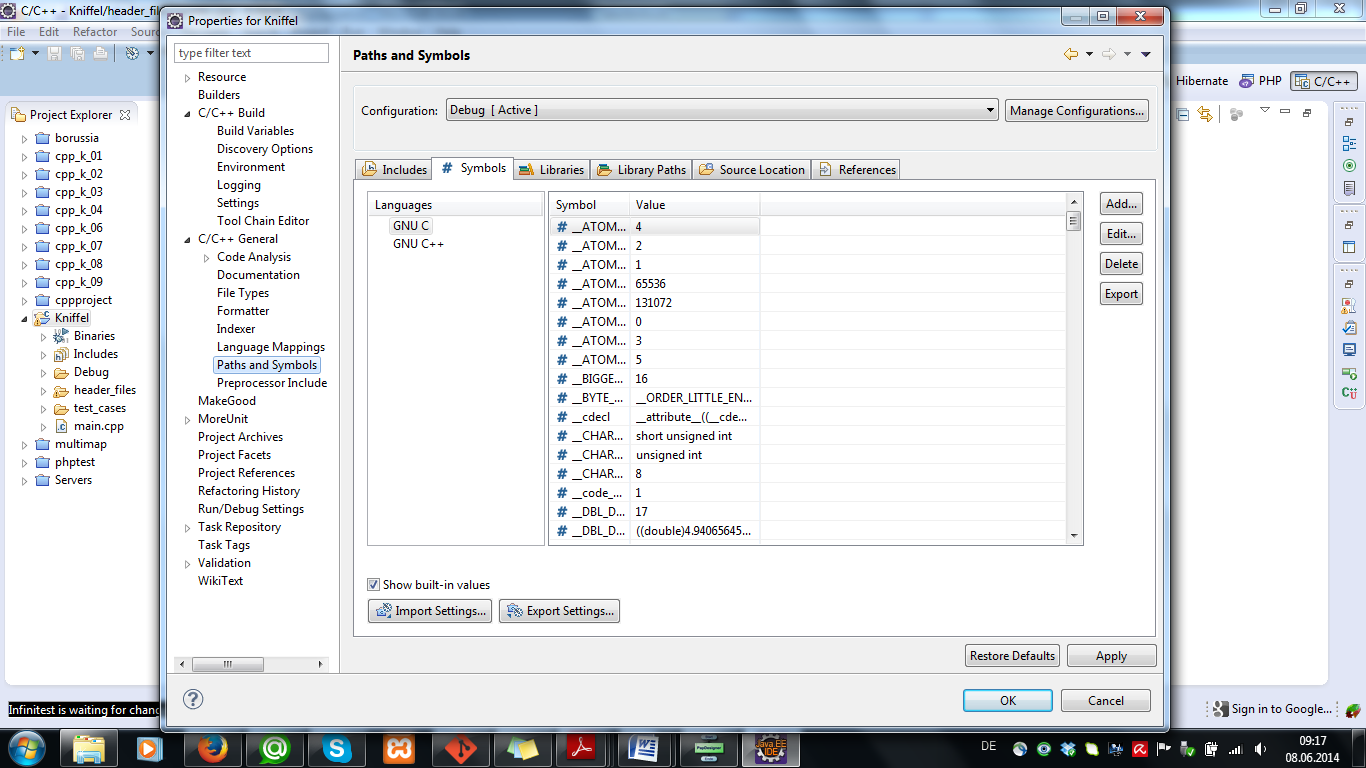
## Anmerkung

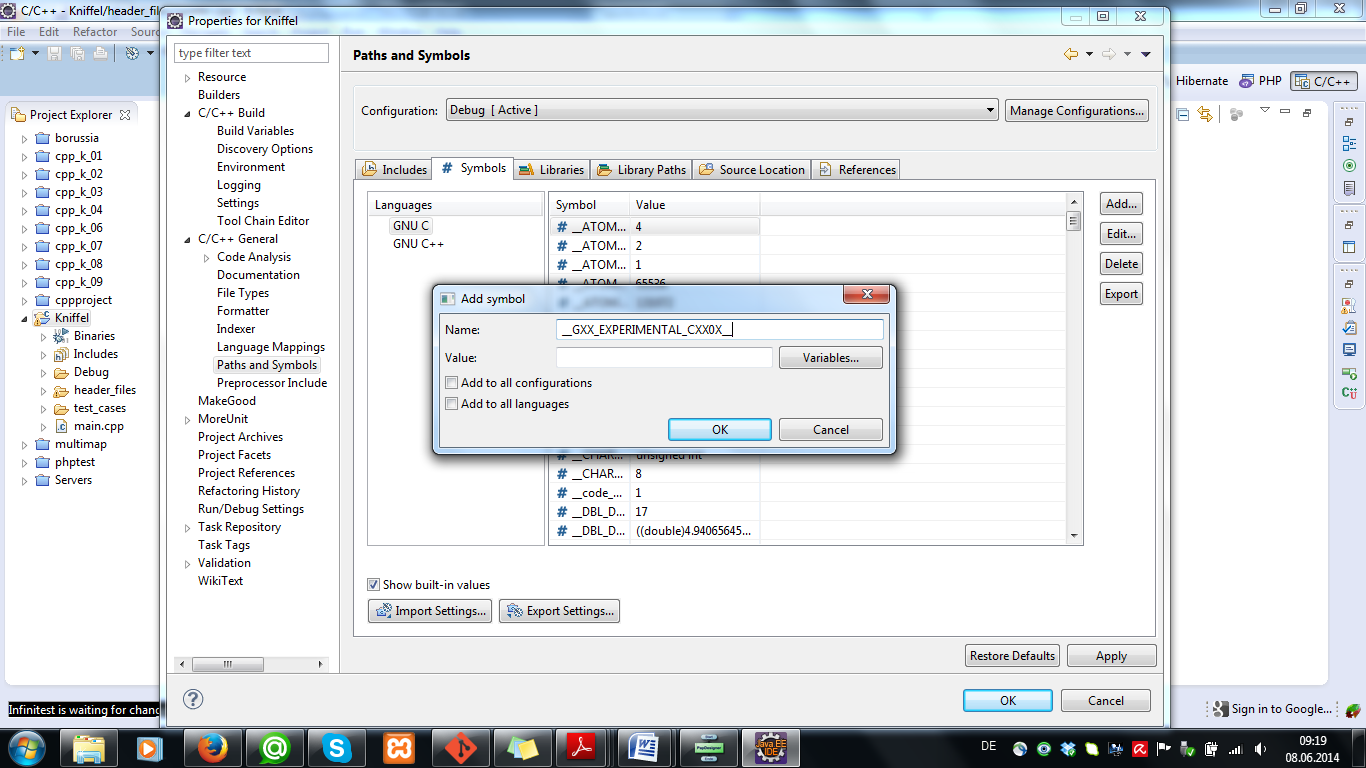
In Ecpilse tritt folgende Fehlermeldung „Symbole „unique\_ptr“ could not be resolved“ auf. Um diesen Fehler zu beheben, folgen Sie den Anweisungen:

Wenn Sie mit der rechten Maustaste auf das Projekt „**Kniffel**“ klicken, wählen Sie „**Properties**“ aus. Nachdem Sie „**Properties**“ ausgewählt haben, erscheint das „**Properties for Kniffel**“ Fenster.

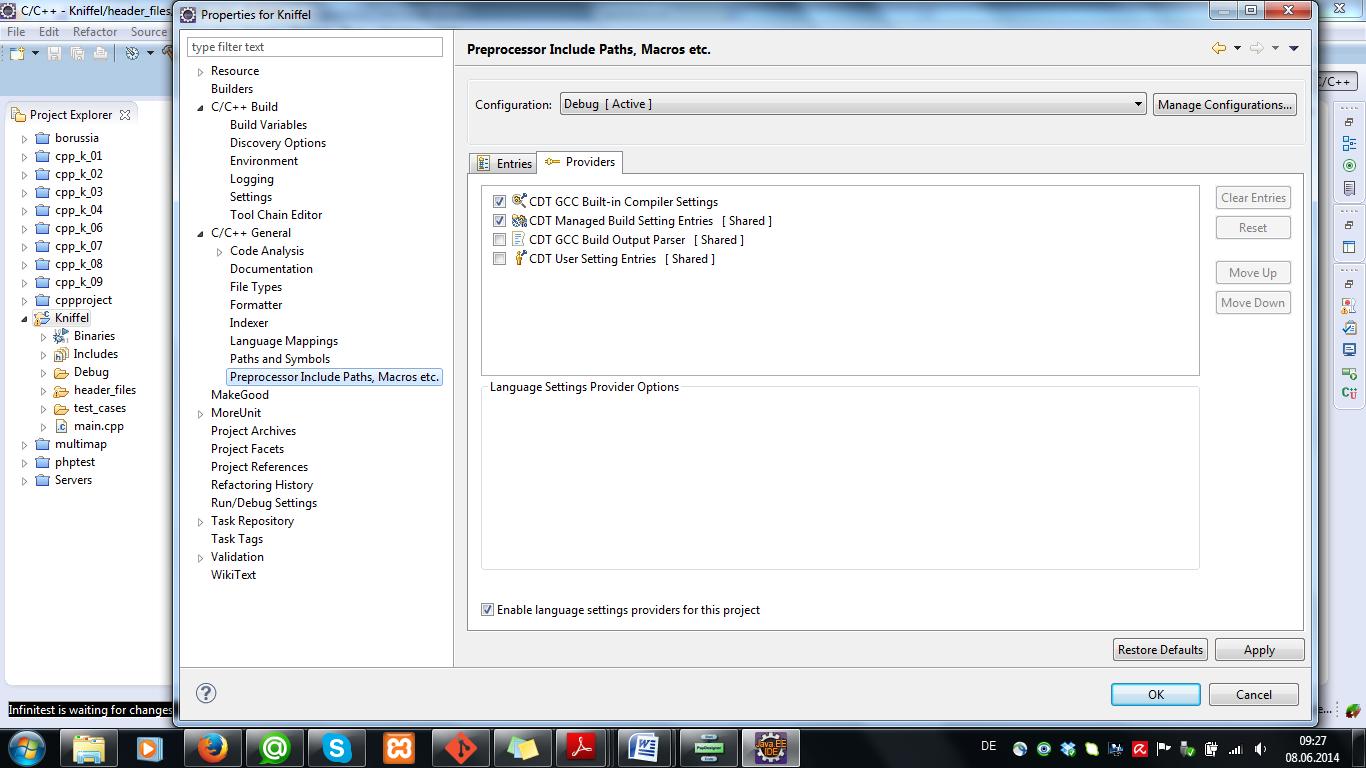


Wählen Sie im folgenden Fenster **C/C++ Build 🡪 Settings 🡪 Tool Settings 🡪 GCC C++ Compiler 🡪 Miscellaneous** 🡪 **Other Flags** aus. Überprüfen Sie, dass am Ende der Zeile **„-std=c++11**“ steht.



Falls die Fehlermeldung immer noch auftritt, klicken Sie mit der rechten Maustaste auf das Projekt. Dann gehen Sie unter **C/C++ General 🡪 Paths and Symbols 🡪 Symbols 🡪 GNU C++**

Auf der rechten Seite finden Sie den Button „**Add**“, klicken Sie darauf und fügen Sie unter „**Name**“ “**\_\_GXX\_EXPERIMENTAL\_CXX0X\_\_**” hinzu. Das Feld „**Value**“ lassen Sie leer. Überprüfen Sie, dass Sie zwei Unterstriche am Anfang und am Ende eigegeben haben.



Danach gehen Sie unter **C/C++ General 🡪Preprocessor Include Path Macros etc. 🡪 Providers**. Wählen Sie “**CDT Managed Build Setting Entries**” aus. Nachdem Sie auf “**Apply**” geklickt haben, gehen Sie unter “**Entries**” Tab. Unter **GNU C++** können Sie “**CDT Managed Build Setting Entries**” sehen. Überprüfen Sie, ob die gerade definierte Variable “**\_\_GXX\_EXPERIMENTAL\_CXX0X\_\_**” vorhanden ist. Nach der Überprüfung klicken Sie auf “**Apply**”. Nachdem alle Schritte erfolgreich abgearbeitet sind, muss das Projekt erneut kompiliert werden „**Clean and Build Project**“.

## Downloads

1. <https://netbeans.org/downloads/> - NetBeans IDE 8.0
2. <http://www.eclipse.org/downloads/packages/eclipse-ide-cc-developers/junor> - Eclipse (Juno) IDE for C/C++ Developers
3. <https://code.google.com/p/googletest/downloads/list> - GoogleTest (Google C++ Testing Framework)