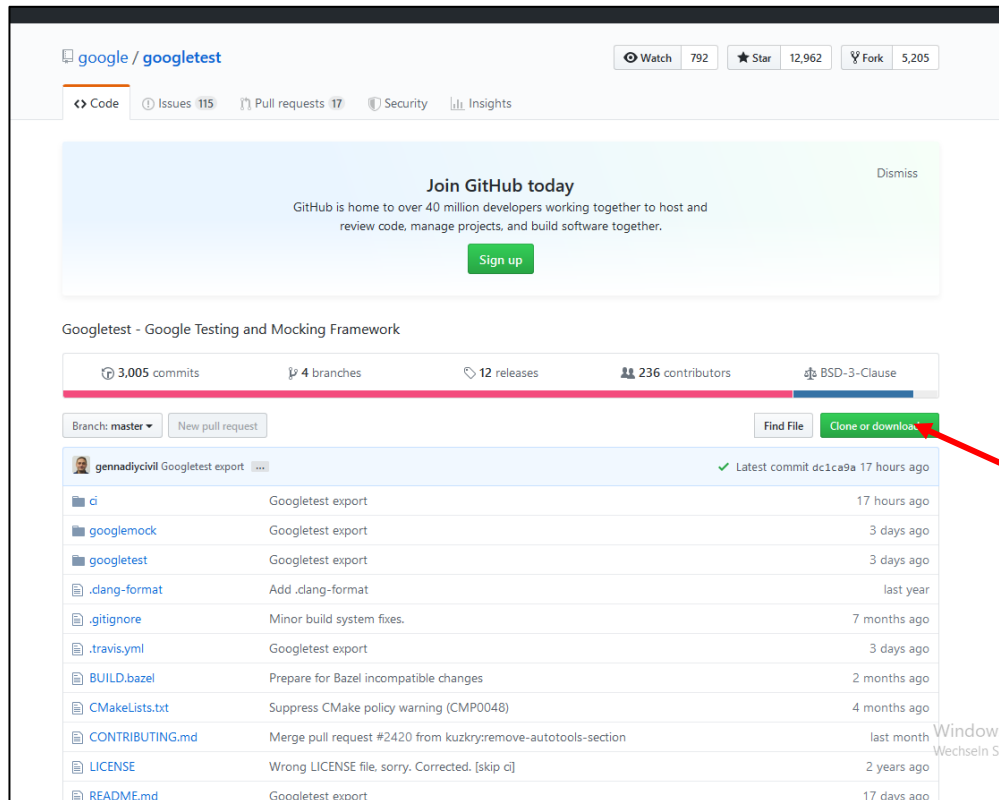


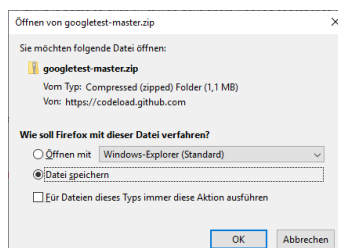
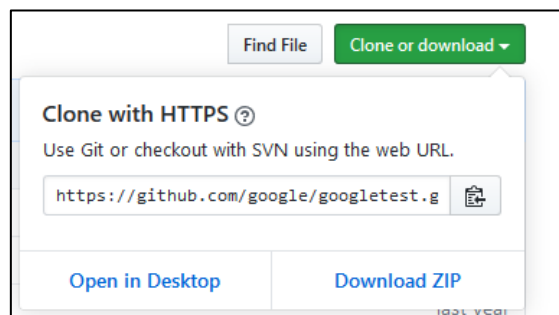
NetBeans und MinGW müssen installiert sein. Es muss eine MinGW Version sein, in der im include Ordner u.a. die Dateien crtDBG.h und crtdefs.h liegen.

## 1. GoogleTest herunterladen:

<https://code.google.com/p/googletest/>



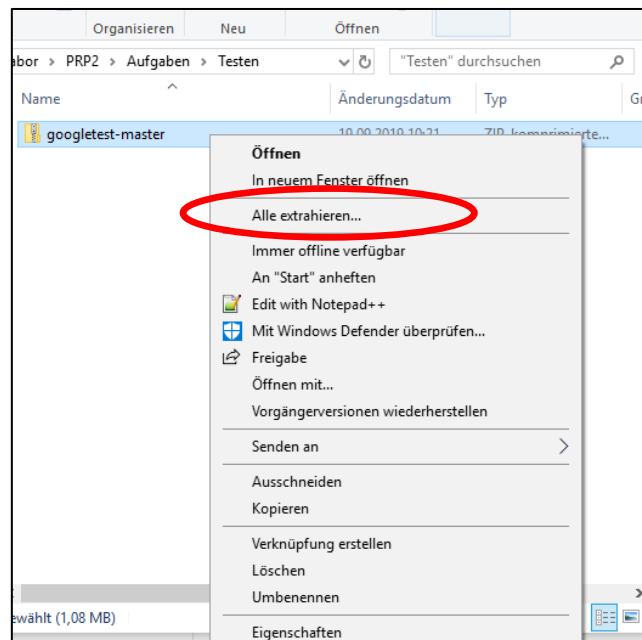
Clone or download.



Datei speichern.

Silke Behn, Zhen Ru Dai (Stand 07.10.2019 )

Das ZIP File extrahieren -> rechtsklick auf die Datei und dann „Alle extrahieren“.

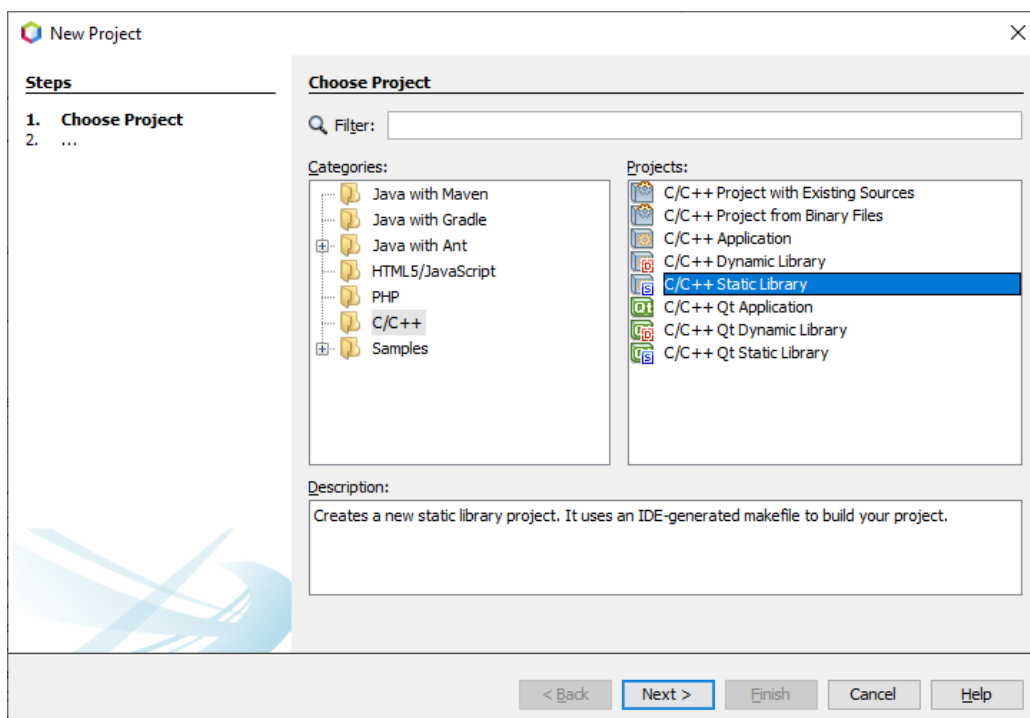


## 2. GoogleTest in eine Static Library compilieren

Nach dem Extrahieren haben wir einen Ordner <googletest-master>.

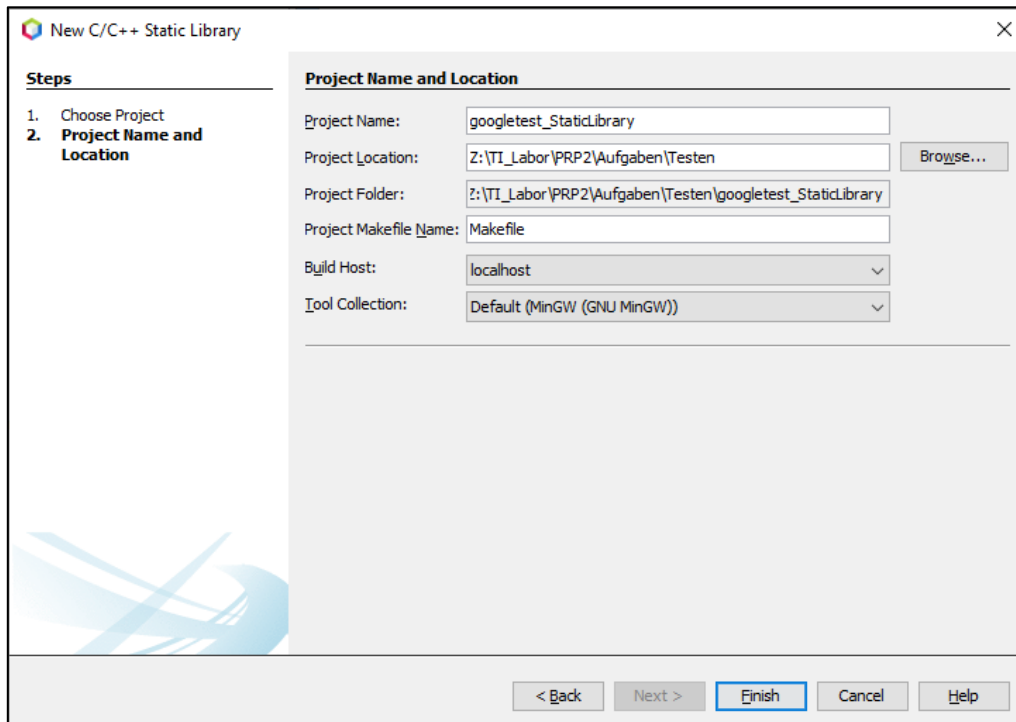
NetBeans starten.

File -> NewProject -> C/C++ - C/C++ Static Library.



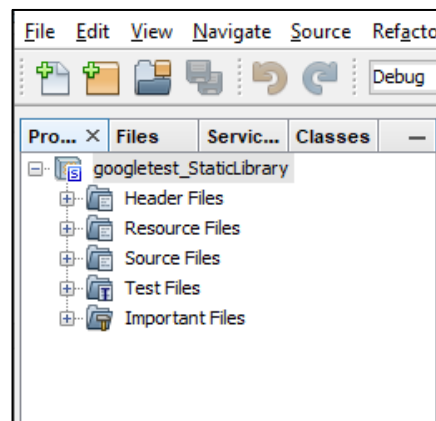
ProjectName: z.B. googletest\_StaticLibrary.

Project Folder: z.B. Z:\TI\_Labor\PRP2\Aufgaben\Testen.



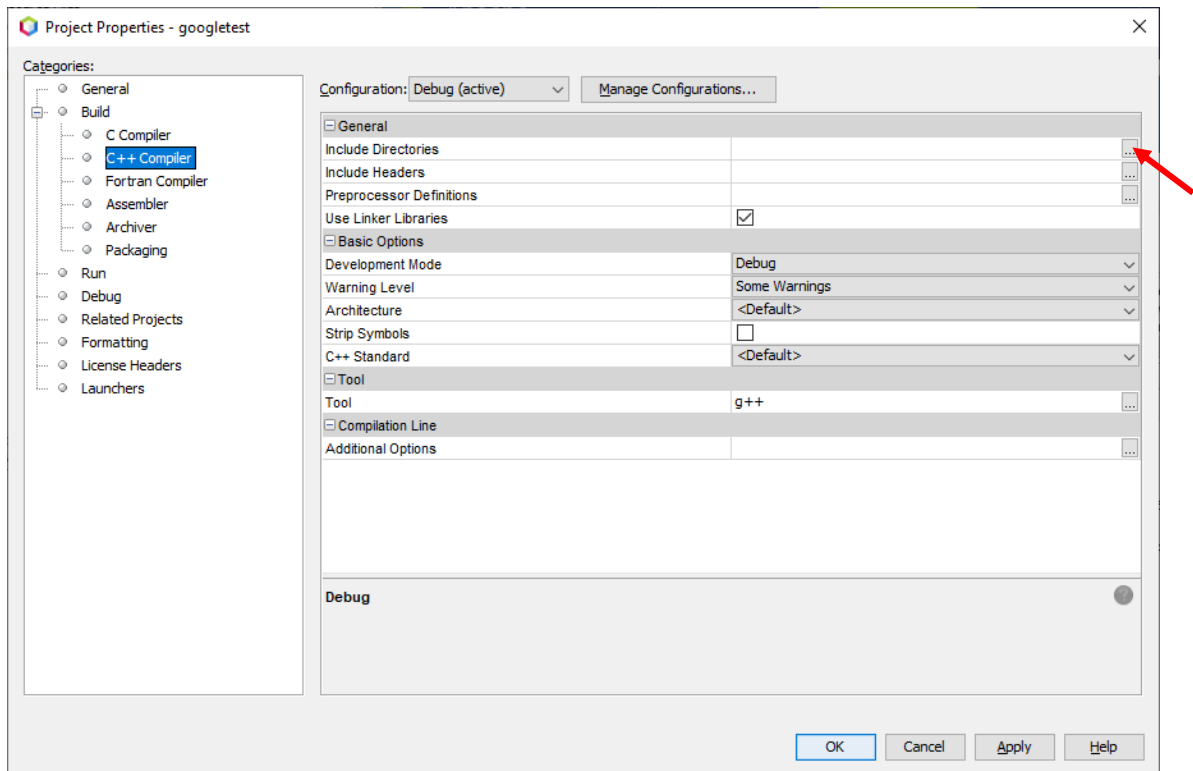
Finish.

Das Static Library Projekt mit dem Namen googletest\_StaticLibrary wird erstellt.

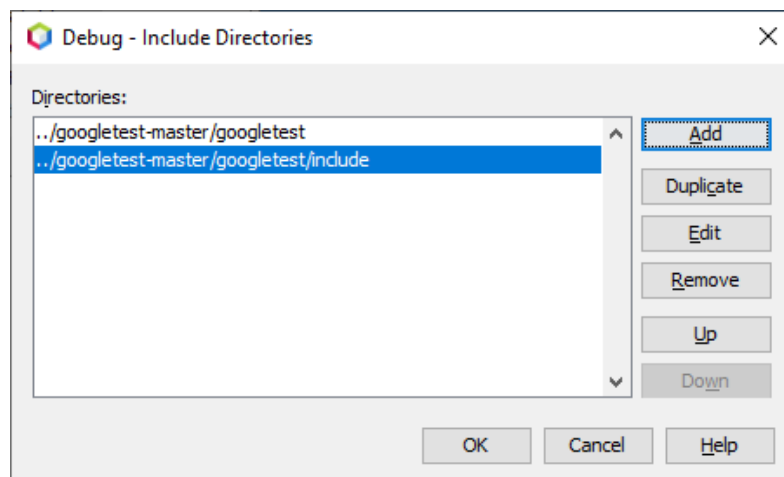


Rechtsklick auf googletest\_StaticLibrary => Properties

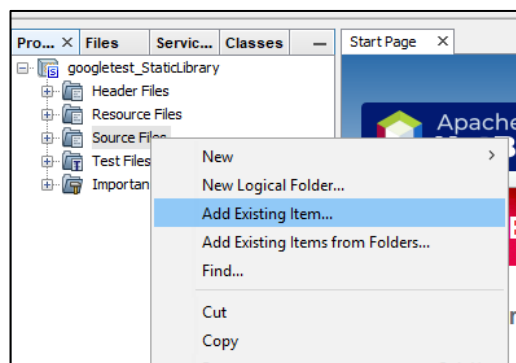
Build => C++ Compiler => Include Directories



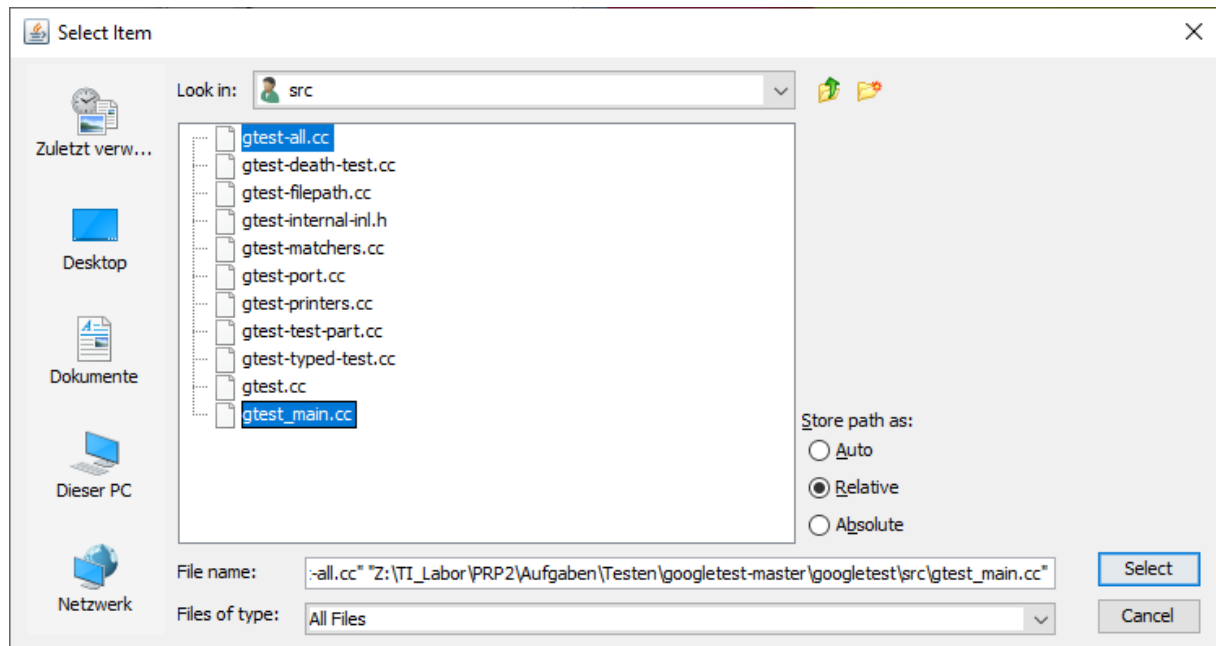
Hier müssen die Ordner „googletest-master/googletest“ und „googletest-master/googletest/include“ eingetragen werden.



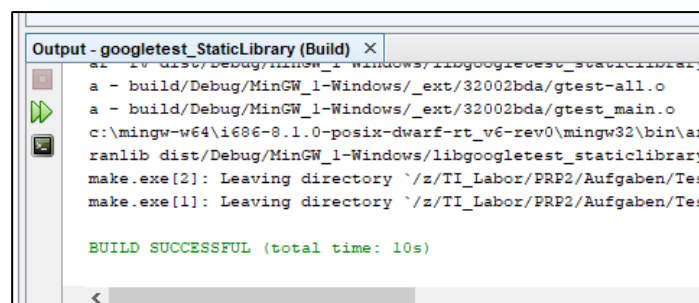
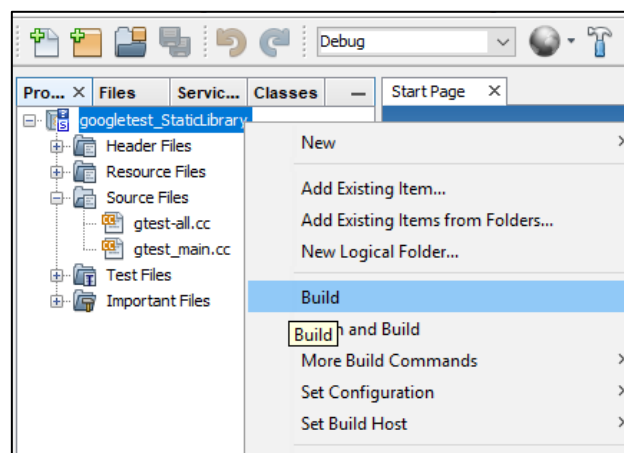
Nun müssen zwei Files in das Projekt aufgenommen werden.  
Rechtsklick auf Source Files => Add Existing Item ...



In googletest-master / googletest/src die Dateien: gtest-all.cc und gtest\_main.cc auswählen



Nun kann die Library compiliert werden: Rechtsklick auf das Projekt => Build.

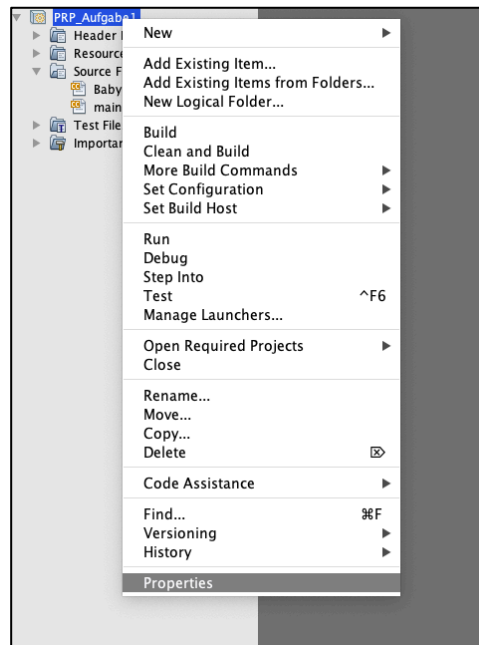


Im Output Fenster sollte nun „BUILD SUCCESSFUL“ stehen.

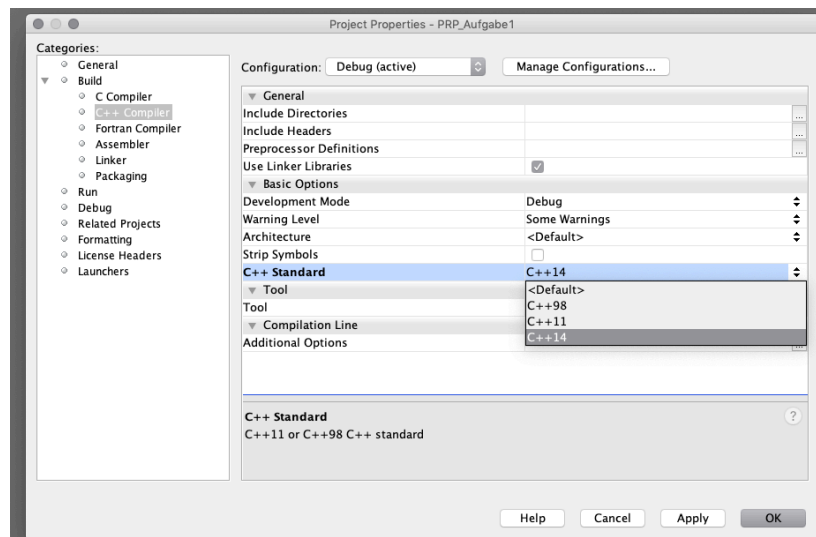
### 3. Einbinden von Google Test Library im Projekt

Für die Aufgaben wird jeweils ein C++ Application Projekt erstellt. Nun sollen die Aufgaben mit Hilfe der Library getestet werden.

Stellen Sie erst den Compiler auf Standard 14 ein. Rechtsklick auf Projekt -> Properties.

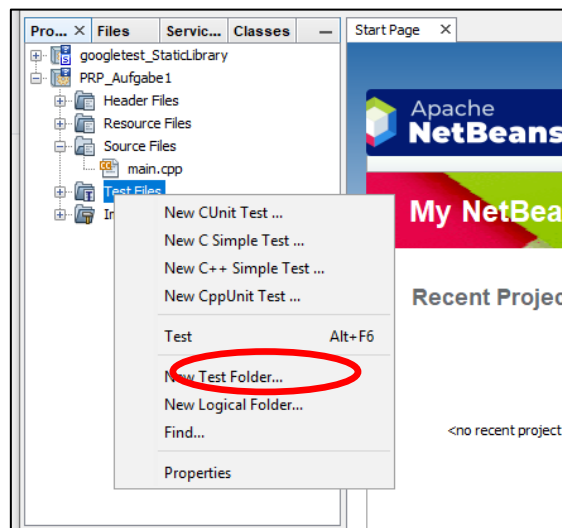


Wähle den C++14 Standard aus.

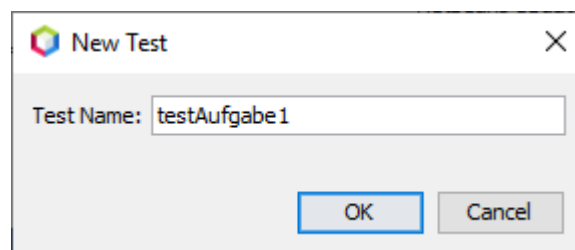


OK drücken.

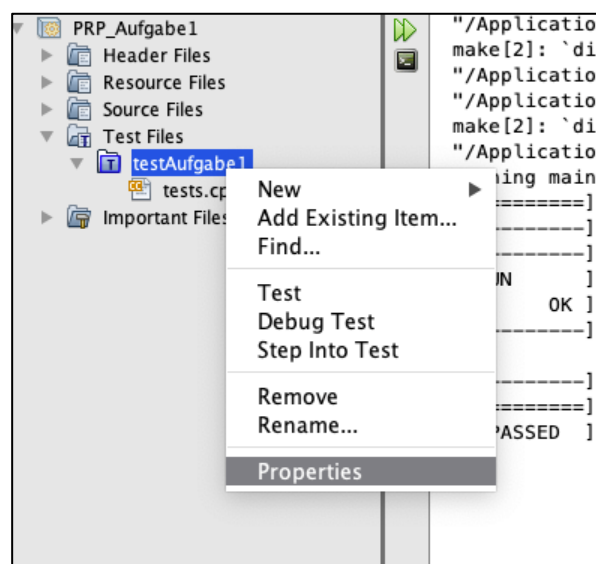
Im Projekt ein Rechtsklick auf Test Files.  
New Test Folder.

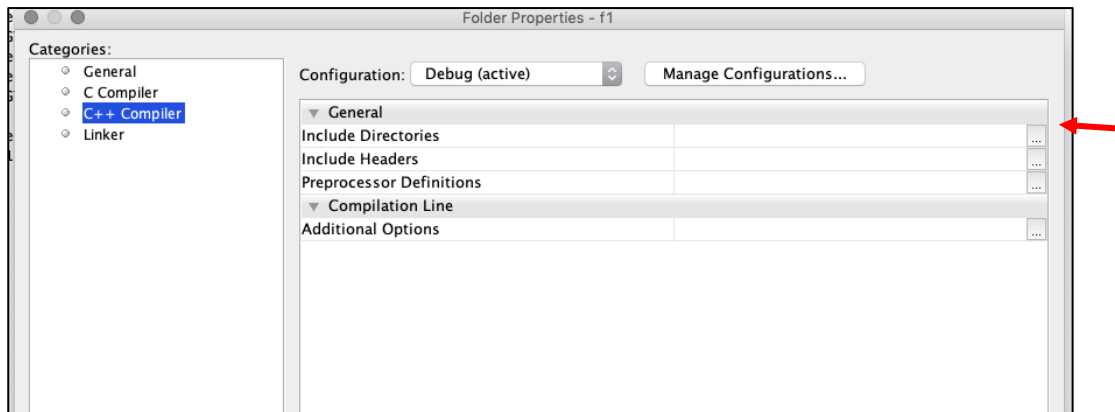


Einen Namen geben.

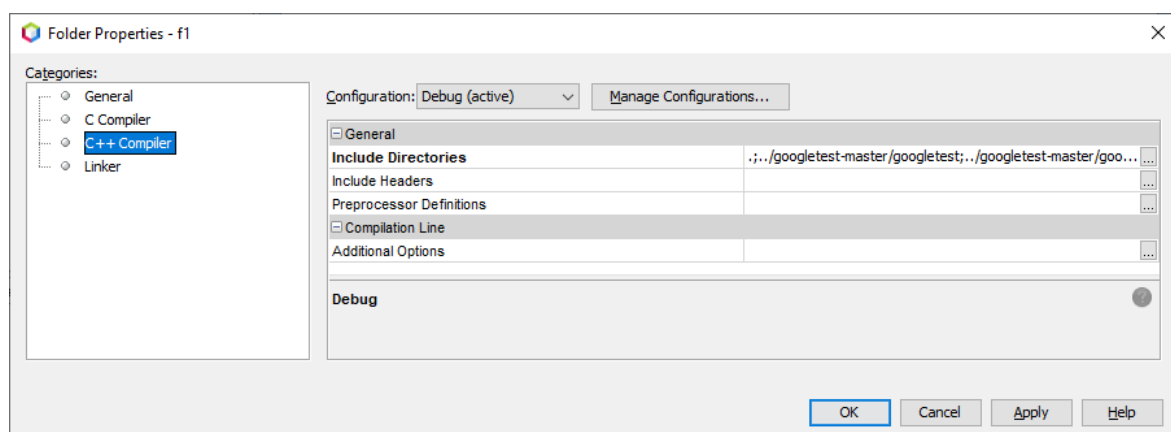
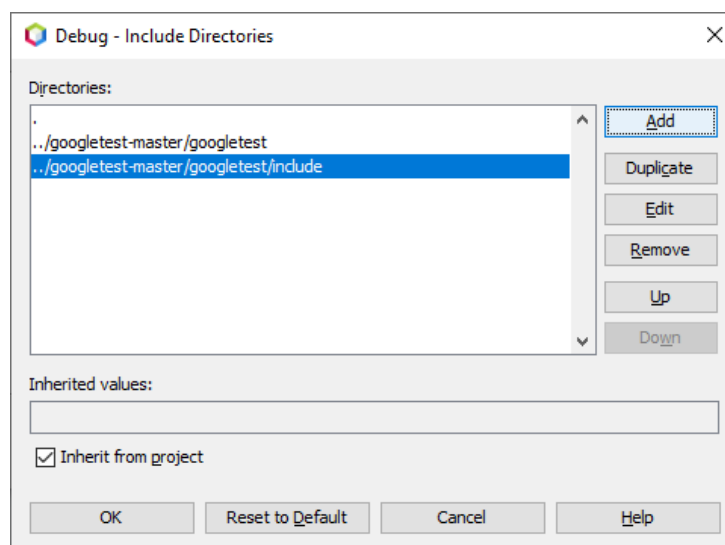


Rechtsklick auf testAufgabe1 und dort Properties wählen.



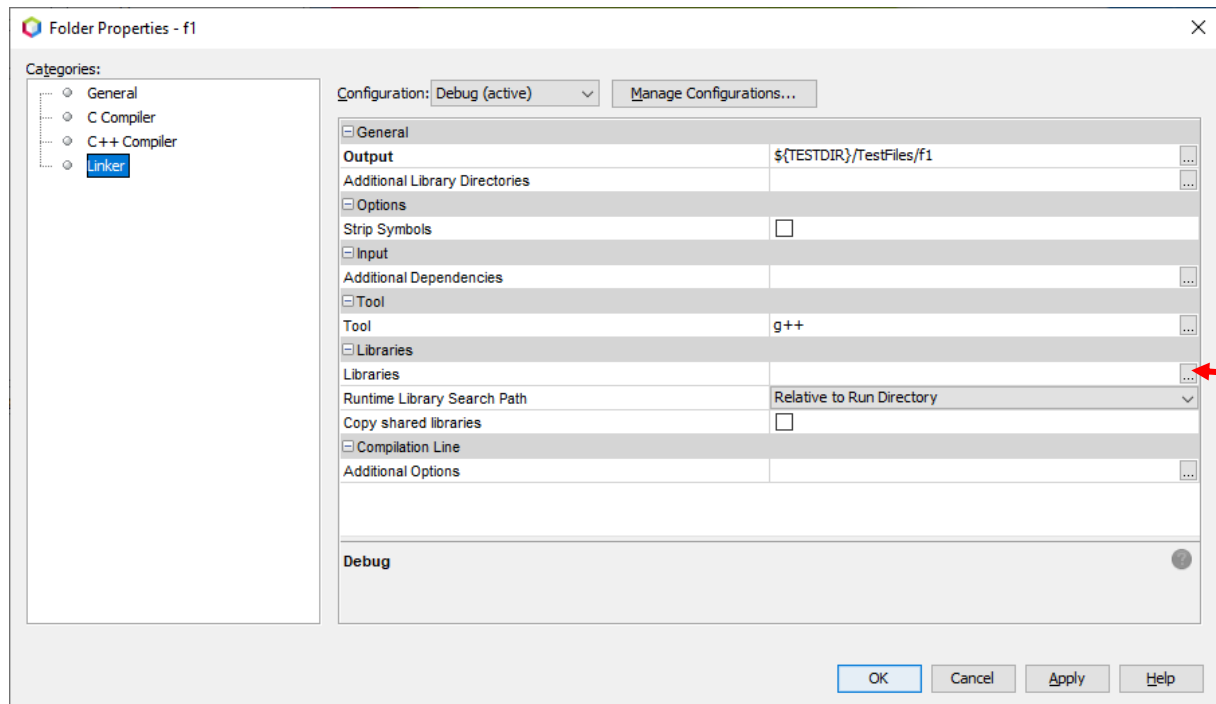


Bei C++ Compiler wieder die beiden Pfade zu googletest und googletest/include eintragen:

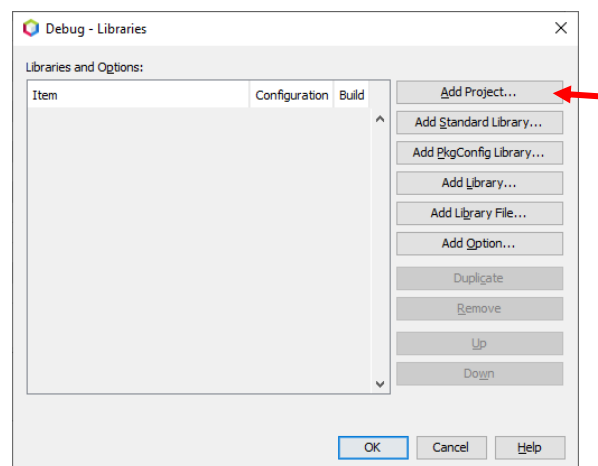


Nun muss die googletest Library in das Projekt eingebunden werden:  
Im Properties Fenster vom testAufgabe1 Ordner (wie oben) => in Linker => Libraries.

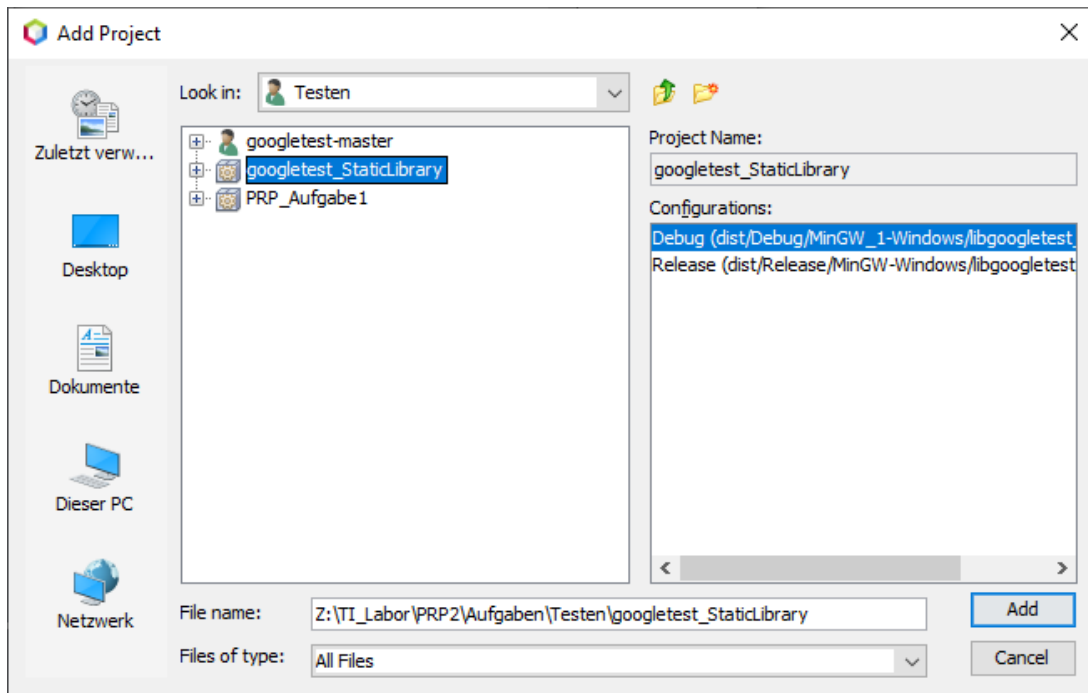




Add Project.

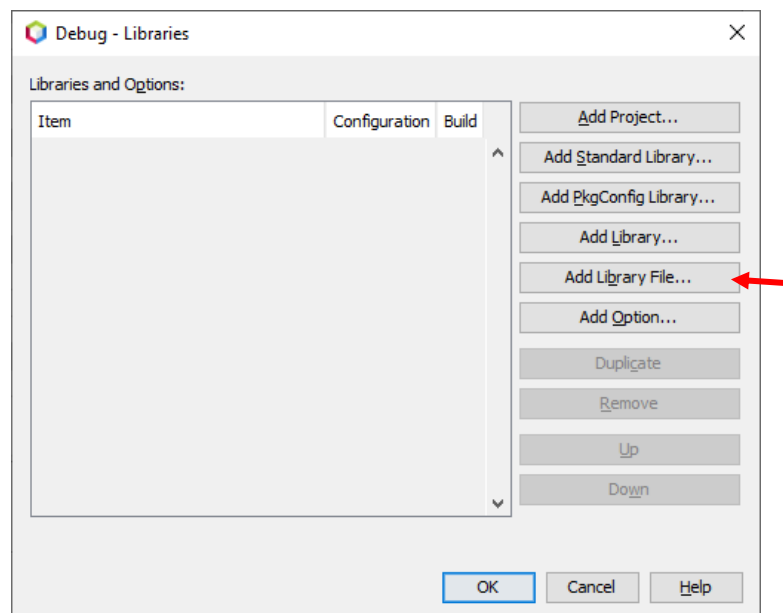


Das static Library Projekt Auswählen-> googletest\_StaticLibrary.

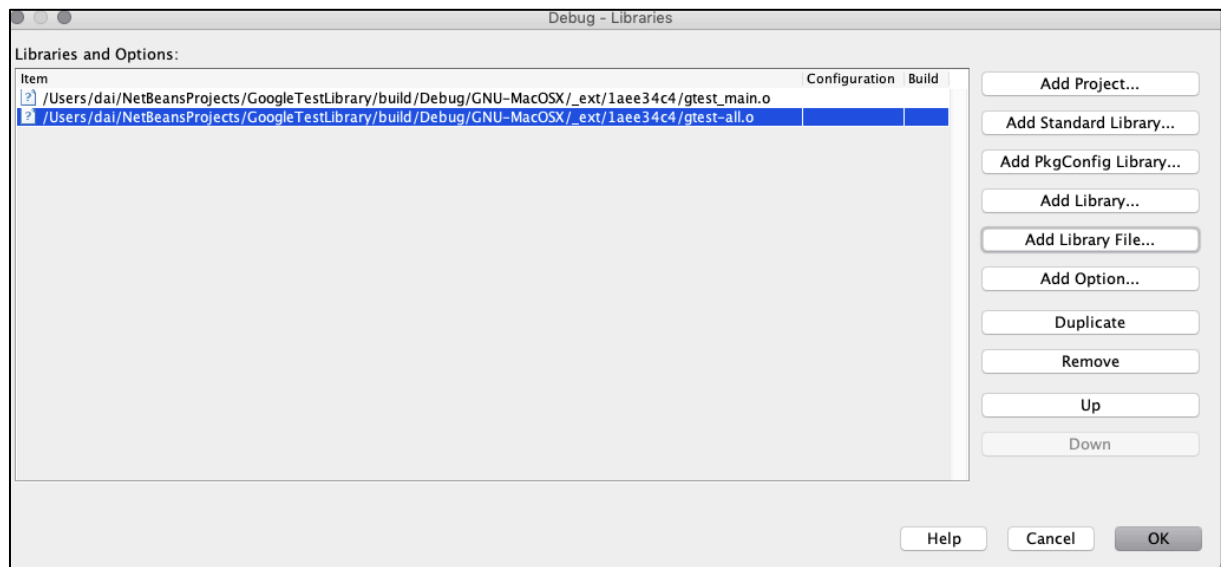


Wenn es Schwierigkeiten beim Projekt hinzufügen geben sollte (z.B. als MAC Benutzer), dann machen Sie folgendes:

Im Properties Fenster vom testAufgabe1 Ordner (wie oben) => in Linker => Libraries => Add Library File.

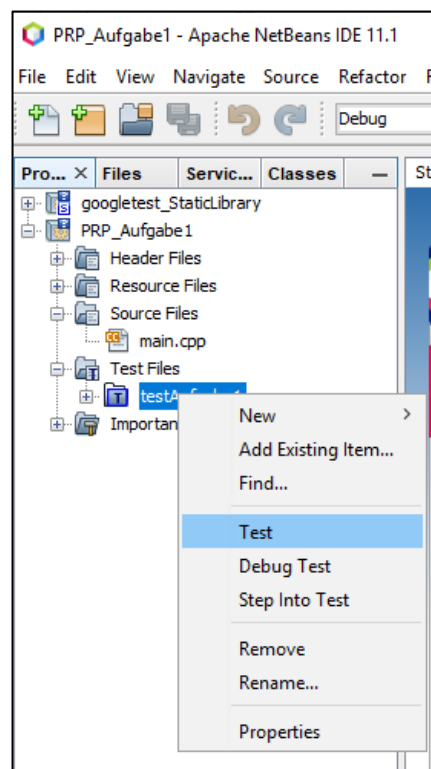


In GoogleTest\_StaticLibrary/build/Debug/.../gtest\_main.o und ../gtest\_all.o auswählen.

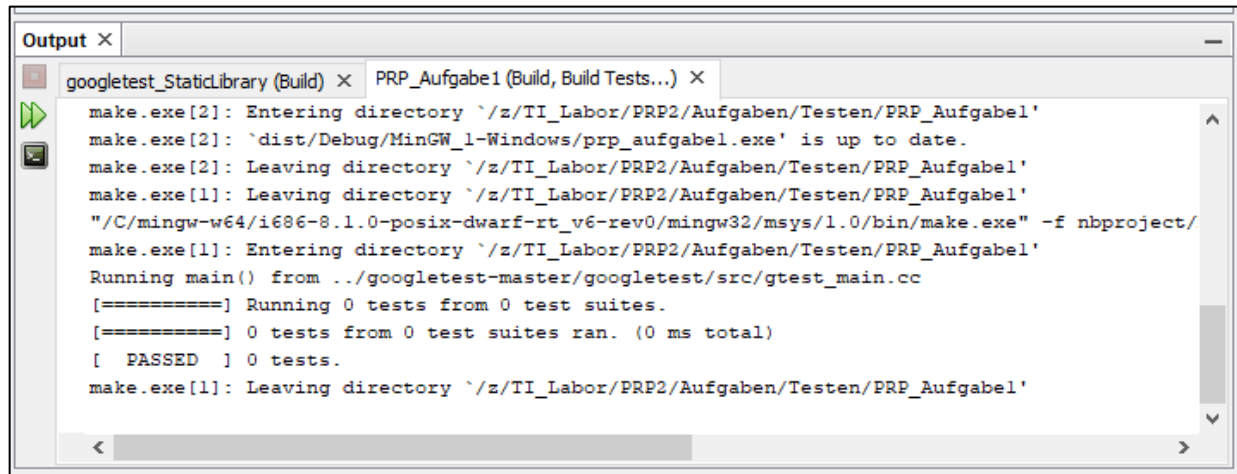


Jetzt sollte es auch für euch klappen.

Nun kann das Projekt getestet werden, auch wenn noch keine Tests geschrieben wurden:  
Rechtsklick auf testAufgabe1 => Test.



Im Output Fenster steht dann erstmal das folgende Ergebnis:

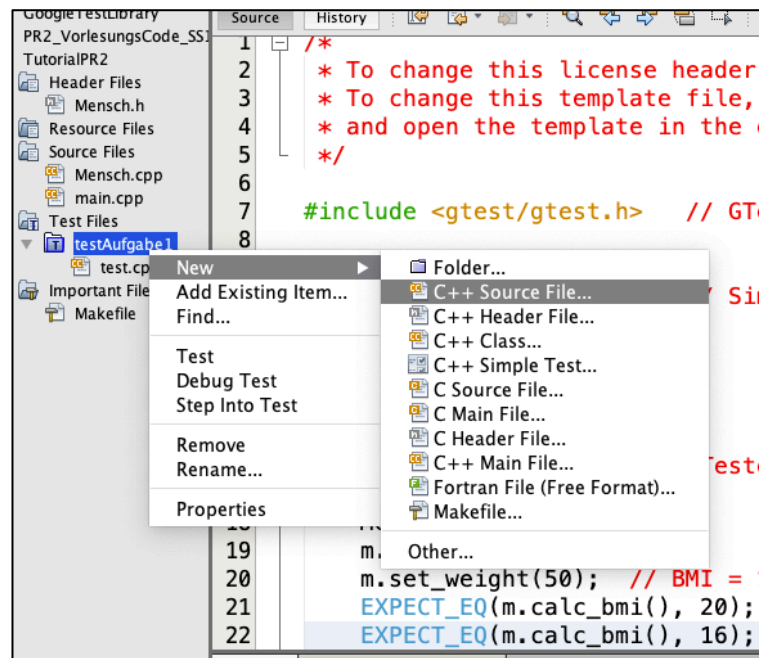


```
Output X
googletest_StaticLibrary (Build) X PRP_Aufgabe1 (Build, Build Tests...) X
make.exe[2]: Entering directory `./z/TI_Labor/PRP2/Aufgaben/Testen/PRP_Aufgabe1'
make.exe[2]: `dist/Debug/MinGW_1-Windows/prp_aufgabe1.exe' is up to date.
make.exe[2]: Leaving directory `./z/TI_Labor/PRP2/Aufgaben/Testen/PRP_Aufgabe1'
make.exe[1]: Leaving directory `./z/TI_Labor/PRP2/Aufgaben/Testen/PRP_Aufgabe1'
"/C:/mingw-w64/i686-8.1.0-posix-dwarf-rt_v6-rev0/mingw32/msys/1.0/bin/make.exe" -f nbproject/
make.exe[1]: Entering directory `./z/TI_Labor/PRP2/Aufgaben/Testen/PRP_Aufgabe1'
Running main() from ../googletest-master/googletest/src/gtest_main.cc
[=====] Running 0 tests from 0 test suites.
[=====] 0 tests from 0 test suites ran. (0 ms total)
[ PASSED ] 0 tests.
make.exe[1]: Leaving directory `./z/TI_Labor/PRP2/Aufgaben/Testen/PRP_Aufgabe1'
```

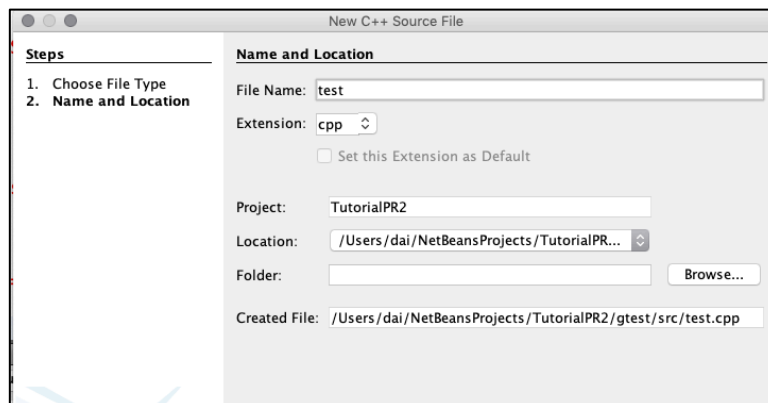
Es wurden 0 Tests durchgeführt und es gab kein Problem.

#### 4. Testen von C++ Programmen

Test Datei erstellen, in dem die Tests implementiert werden soll. Rechtsklick auf testAufgabe1 -> New -> C++ Source File.



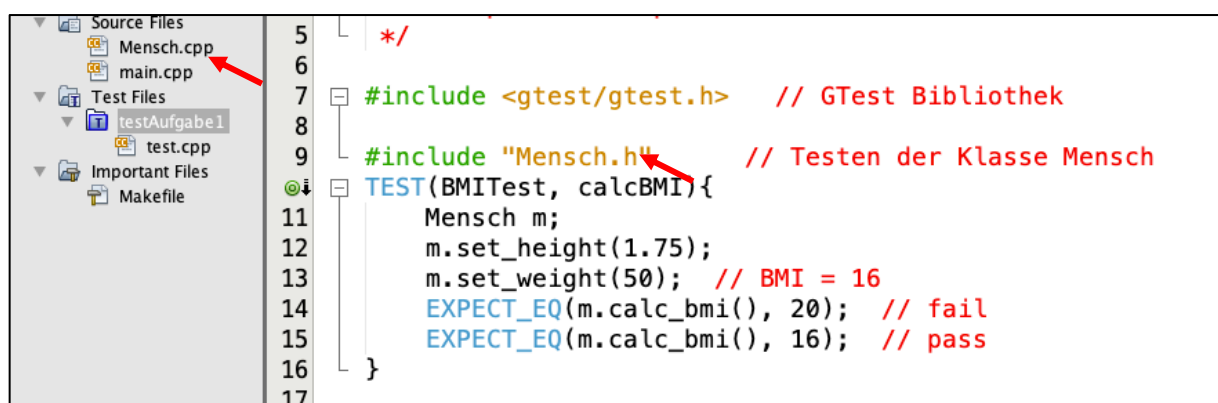
Einen Namen geben, z.B. test.cpp.



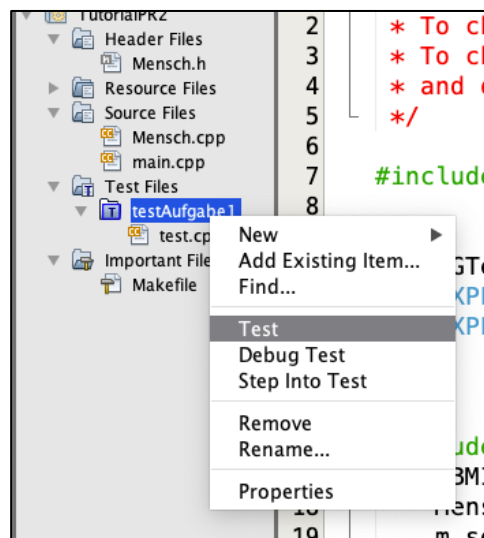
In der test.cpp Datei muss die GoogleTest Library eingebunden werden. Anschließend können die Google Tests geschrieben werden.



Wenn Sie eine bestimmte Klasse (z.B. Mensch) testen wollen, müssen Sie diese Klasse erstmal inkludieren.



Starten Sie den Test. Rechtsklicken auf den test folder und auf Test klicken:



Folgendes Ergebnis sollte dargestellt werden:

```
Running main() from /Users/dai/googletest-master/googletest/src/gtest_main.cc
[=====] Running 1 test from 1 test suite.
[-----] Global test environment set-up.
[-----] 1 test from BMITest
[ RUN      ] BMITest.calcBMI
/Users/dai/NetBeansProjects/TutorialPR2/gtest/src/test.cpp:14: Failure
Expected equality of these values:
  m.calc_bmi()
    Which is: 16
  20
[  FAILED  ] BMITest.calcBMI (0 ms)
[-----] 1 test from BMITest (0 ms total)

[-----] Global test environment tear-down
[=====] 1 test from 1 test suite ran. (0 ms total)
[  PASSED  ] 0 tests.
[  FAILED  ] 1 test, listed below:
[  FAILED  ] BMITest.calcBMI

1 FAILED TEST
```

Weitere Informationen über Google Test Implementierungen:

<https://github.com/google/googletest/blob/master/googletest/docs/primer.md>