

Programmiertechniken 1

(automatisierte) Tests

Qualitätssicherung

- Auch bei Software muss eine Qualitätssicherung auf verschiedenen Ebenen durchgeführt werden.
- Überprüfung der Funktion von komplexer Software wird immer schwieriger.
- Minimal: Test gegen eine Spezifikation.

Testen

- **Testen** ist **systematischer, automatisierter** Vergleich eines **realen** Systemverhaltens mit einem **erwarteten** Systemverhalten.
- Ablauf und Erwartungen werden in **Testspezifikation** definiert.
- Durch Testen kann keine vollständige Überprüfung der Funktion festgestellt werden!
Zielsetzung ist das Finden von Abweichungen.
- Automatisierung ermöglicht schnelle Wiederholbarkeit (→ **Regression Testing**).
- Entwicklung gegen Tests: **Test Driven Development**

Testspezifikation - Beispiel

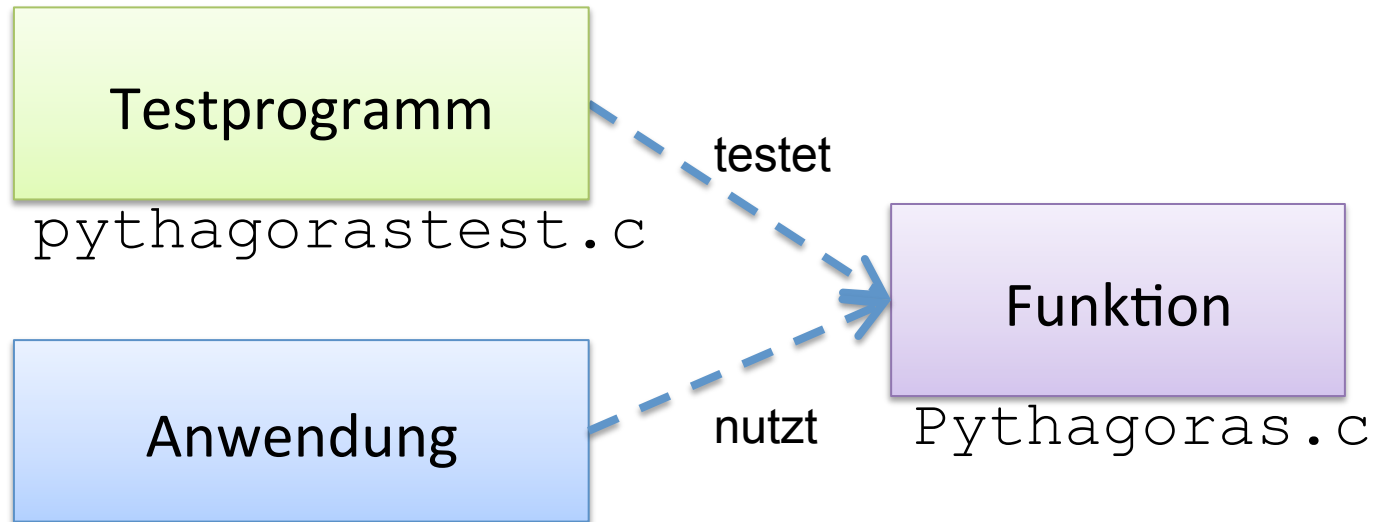
Funktion: double pythagoras(double a, double b)

Source File: pythagoras.c

Anforderungen: <Referenz auf Aufgabenbeschreibung>

Test Nr.	Argumente	Erwartung
1	a=4, b=4	5
2	...	

Testprogramm



Testprogramm trägt den gleichen Namen wie Source-Datei ergänzt um Postfix „test“.

Testprogramm - Testausführung

Testprogramm führt die Funktion mit den Argumenten der Spezifikation aus und vergleicht mit der Erwartung. Das Testprogramm soll nur Fehlschläge melden.

In `ti_expectations.h` sind Macros für den Vergleich bereitgestellt.

Bespiel:

`EXPECT_TRUE()` erwartet, dass der Ausdruck in der Klammer als Ergebnis wahr liefert.

```
EXPECT_TRUE(2.0 == sqrt(4));  
EXPECT_TRUE(2.0 < foo());
```

Hinweis: Bei Virtual-C muss die Konstante links stehen!