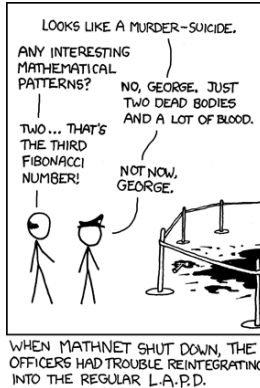


Einführung in die Programmiertechnik II

Übung 1



Johannes Wolf

Übungsbetrieb

- Übungstermine voraussichtlich alle 2 Wochen Dienstags
- Bearbeitungszeit in der Regel zwei Wochen
- Abgabe zu zweit möglich und empfohlen

Übungsleiter

- Johannes Wolf
H-2.26 (johannes.wolf@hpi.de)

Weitere Ansprechpartner

- Heiko Thiel
H-2.14 (heiko.thiel@hpi.de)
- Sören Discher
H-2.14 (soeren.discher@hpi.de)

- Leonardo da Pisa (* um 1170 in Pisa; † nach 1240 ebenda)
- *figlio di Bonaccio* – Sohn des Bonaccio
- 1202: *Liber abbaci* – Buch der Rechenkunst
 - 15 Kapitel
 - „Von der Kenntnis der neun Zahlzeichen der Inder, und wie mit ihnen jegliche Zahl geschrieben wird; und wie die Zahlen mit den Händen gemerkt werden sollen, und von der Einführung der Rechenkunst.“
 - „Von der Multiplikation natürlicher Zahlen.“
 - „Von der Addition derselben miteinander.“
 - „Von der Subtraktion kleinerer Zahlen von größeren.“
 - „Von der Teilung natürlicher Zahlen durch natürliche Zahlen.“
 - ...
 - „Von den Lösungen vieler Fragen, die wir als erratische bezeichnen.“

- „Fibonacci illustrierte die Folge durch die einfache mathematische Modellierung des Wachstums einer Population von Kaninchen nach folgenden Regeln:
 - 1 Jedes Paar Kaninchen wirft pro Monat ein weiteres Paar Kaninchen.
 - 2 Ein neugeborenes Paar bekommt erst im zweiten Lebensmonat Nachwuchs (die Austragungszeit reicht von einem Monat in den nächsten).
 - 3 Die Tiere befinden sich in einem abgeschlossenen Raum, sodass kein Tier die Population verlassen und keines von außen hinzukommen kann.
- Fibonacci führte den Sachverhalt für die zwölf Monate eines Jahres vor (2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377) und wies auf das Bildungsgesetz der Reihe durch Summierung jeweils zweier aufeinanderfolgender Reihenglieder ($2 + 3 = 5$, $3 + 5 = 8$, $5 + 8 = 13$ usw.) hin. Er merkte außerdem an, dass die Reihe sich nach diesem Prinzip für eine unendliche Zahl von Monaten fortsetzen lässt, was dann allerdings unsterbliche Kaninchen voraussetzt.“ [Wikipedia]



CMake

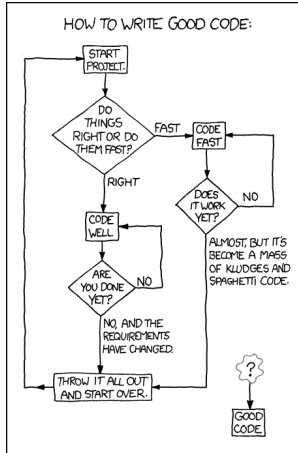
- „CMake is an open-source, cross-platform family of tools designed to build, test and package software. CMake is used to control the software compilation process using simple platform and compiler independent configuration files, and generate native makefiles and workspaces that can be used in the compiler environment of your choice.“
- „The `cmake` executable is the CMake command-line interface. It may be used to configure projects in scripts. CMake is a cross-platform build system generator. Projects specify their build process with platform-independent CMake listfiles included in each directory of a source tree with the name `CMakeLists.txt`. Users build a project by using CMake to generate a build system for a native tool on their platform.“

NMake

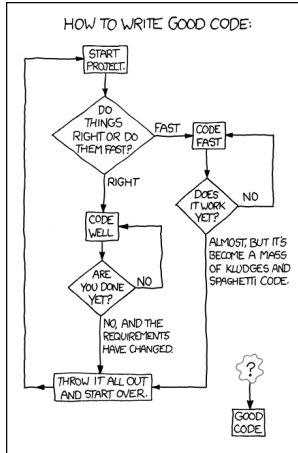
- „The Microsoft Program Maintenance Utility (NMAKE.EXE) is a command-line tool that builds projects based on commands that are contained in a description file.“

Übung 1 | Kompilieren des Programmrahmens unter Windows

- 1 Erzeugen eines Unterordners `build`
- 2 Wechseln in den Unterordner
- 3 Ausführen von `vcvarsall.bat x86_amd64`
 - konfiguriert Compiler für richtige Umgebung
- 4 Ausführen von `cmake .. -G "NMake Makefiles"`
 - erzeugt Makefile
- 5 Kompilieren mit `cmake --build .`
 - erzeugt ausführbare Dateien



How to write good code. [xkcd]



How to write good code. [xkcd]

- Aussagekräftige Variablen- und Funktionsnamen verwenden
- Code kommentieren
- Compiler-Warnungen beachten