

## **Informatika 4**

**0/0/4/f/4**

Tárgyfelelős: Pröhle Péter

További oktatók: Wettl Ferenc

**ALCÍM:** Egy nagyteljesítményű programozási rendszer megismerése, és a szoftverfejlesztés alapjai

**A CÉL** egy, a természettudományos és nagy gyakorlati problémák kezelésére gyakran használt nyelv (pl.: C++) megismerése, és segítségével egy összetettebb feladat megoldása.

**RÉSZLETES TEMATIKA:**

A nyelv haladó szintű megismerése, konstruálás orientált interfészek (flex, bison, XML parzerek, ...), választás orientált interfészek (portábilis GUI, C++ esetén pl.: wxWidgets, Qt, ...).

Nagy projektek és programok részekre bontása. Programrészek kommunikációs felületei, interfészek, absztrakt osztályok, serializáció. Eseményvezérlés programozás. Grafikus és web-es felhasználói felület, XML web-szolgáltatások. Modell-view-kontroller architektúra. Integrált fejlesztő rendszerek (pl.: GNU-Emacs, KDevelop, Eclipse).

Felhasználóbarát szoftverfejlesztés. Szoftvertesztelés, szoftver minősége (regressziós teszt, fordítási figyelmeztetések, típusosság, futási idejű memóriahasználat ellenőrzés, futási idejű nyomkövetés). Modell alapú szoftverfejlesztés (Petri háló, UML).

Irodalom:

Online dokumentációk nagy választékban és a három klasszikus könyv:

Brian W. Kernighan, Dennis M. Ritchie: A C programozási nyelv, Műszaki, 2004

Brian W. Kernighan, Rob Pike: A Unix operációs rendszer, Műszaki, 1999

Stroustrup, Bjarne: A C++ programozási nyelv, Budapest, Kiskapu, 2001

## **Informatics 4**

**0/0/4/f/4**

Course coordinator: Péter Pröhle

Other instructors: Ferenc Wettl

**SUBTITLE:** A programming language of high performance and the basics of software development.

**THE GOAL** is to study a programming language (e.g.: C++) frequently used for handling scientific calculations and huge practical problems, and to solve a more complex problem as a case study.

**DETAILED SYLLABUS:**

Advanced study of the language in question, construction oriented interfaces (flex, bison, XML parsers, ...), selection oriented interfaces (portable GUI, in case of C++: wxWidgets, Qt, ...).

Reducing the huge programs and projects to a structure of modules. Structural interfaces and communication protocols between the modules, abstract classes, serialisation, event driven flow control. Graphical and WEB-oriented user interfaces, XML. Model-view-controller architecture. Integrated development environments (e.g.: GNU-Emacs, KDevelop, Eclipse).

User friendly software development. Testing, evaluating and the quality of softwares (regression test, compiler warnings, type checking, run time check of memory usage and flow control). Model based software development (Petri nets, UML).

References:

The vast amount of online documentations and the a three classical books:

Brian W. Kernighan, Dennis M. Ritchie: The C Programming Language, (Hungarian: Műszaki, 2004)

Brian W. Kernighan, Rob Pike: The Unix Programming Environment, (Hungarian: Műszaki, 1999)

Stroustrup, Bjarne: The C++ Programming Language, (Hungarian: Kiskapu, 2001)

---