

Egy komplex, ipari környezetben használt szoftver kódjának különböző aspektusokból vett modernizációja lenne a szakmai gyakorlatot végző diák feladata, melynek alapját modern kódelemző eszközök által szolgáltatott adatok adják, az átalakítások tesztelését pedig automaták végzik, így nem csupán magával a kóddal, a modernizáció lehetőségeivel és módszereivel, de a munkafolyamat egészével lehetne megismerkedni a szakmai gyakorlat alatt.

C/C++ fejlesztés

Statikus kódanalízis

Statikus kódanalízis végző eszközök,

- [CLang Static Analyzer](#)
- [Cppcheck](#)
- [Lizard](#)

illetve a fordító programok

- [GCC](#)
- [Clang](#)

által jelzett hibák vizsgálata, a javítások tesztelése.

C++/C++11 átalakítások

A C++/C++11 nyelv részeként létező típusok, osztályok használata a korábbi C, illetve [GLib](#)

- hordozható adattípusok (`guint32`, `gboolean`, ...)
- adattárolók (`GList`, `GHashTable`, ...)
- memória allokációs függvények (`g_new`, ...)
- ...

Python fejlesztés

Statikus kódanalízis

Statikus kódanalízis végző eszközök,

- [Pylint](#)
- [PEP8](#)

által jelzett hibák vizsgálata, javítása tesztelése.

Python3 átalakítások

A Python 2.x-es verzióiban is használható Python 3 eszközök, módszerek (*future* modul) meghonosítása, tesztelése.

Automata tesztek

Python nyelven írt automata tesztesetek esetén,

- refaktorizáció
- adoptálás IPv6 környezetbe
illetve új tesztesetek írása,
- tűzfal tesztelésére
- grafikus menedzsment rendszer tesztelésére