Habilitációs és tudományos előadás vázlata
Porkoláb Zoltán

Habilitációs előadás vázlata:

1. Move szemantika C++11-ben

Az előadás során ismertetem a mutató-(referencia-) és az érték-szemantikájú nyelvek különbségeit. Példákon keresztül megmutatom a C++11-ben bevezetett move szemantika előnyeit és technikai eszközrendszerét: a jobbérték mutatókat, a move kontruktort és operátort, és kitérek a szabványos konténerek és a move szemantika kapcsolatára. Áttekintjük a potenciális hibákat, végül az RVO és az optimalizálás kérdései kerülnek sorra.
(Az előadás a Multiparadigma programozás kurzus része MSc hallgatók számára)

2. C++ fordítás, szerkesztés kérdései

Az előadás során olyan egyetemi kurzusokon kevésbé tárgyalt problémákat tekintem át, mint az include-ok negatív hatása a fordítási időre, vagy a bináris kompatibilitás problémája, amit a PIMPL és gyors-PIMPL technikákkal kezelhetünk. Kitérek a template-ek speciális fordítási és szerkesztési követelményeire. Végül tárgyalom a statikus inicializálás és destruálás kérdését és egy megoldását.
(Az előadás a Multiparadigma programozás kurzus része MSc hallgatók számára)

3. C++ template metaprogramozás

Az előadás során bemutatom a C++ template metaprogramzás kialakulását, az alapvető programkonstrukciókat, és azok fejlődését C++98-tól C++20-ig. Megmutatom miért és hogyan használják fel a metaprogramokat és egy gyakorlati példán keresztül végigviszem a C++ metaprogramozás lehetőségeit és a program karbantartására gyakorol pozitív hatásait. Végül kitérek a típikus hibákra és felderítésük lehetőségeire.
(Az előadás a Multiparadigma programozás kurzus része MSc hallgatók számára)

Tudományos előadás vázlata: C++ Template metaprogramok nyomkövetése és profilozása

Az előadás bevezetésében áttekintem a C++ template metaprogramozás szerepét a modern szoftverfejlesztésben, majd ismertetem azokat a specifikus problémákat, melyek a potenciális hibák és rejtett teljesítményproblémák szempontjából a metaprogramozásra specifikusak. Számba veszem a lehetséges metaprogramozási típushibákat és felvázolok két lehetséges utat felderítésükre. Az első a kód intrumentálásán alapuló módszer, amely fordítófüggetlen, de korlátokkal rendelkezik. A második módszer a fordítóprogram módosítását használja fel, végül ez a módszer bekerült 7.0 verziótól kezdve a Clang++ fordítóprogramba. Végül ismertetem a template metaprogram debugger felhasználásait és a továbbfejlesztés lehetőségeit.