

# Tudományos és habilitációs előadások vázlata

Kozsik Tamás

2021. szeptember 27.

Ebben a félévben a *Programozási nyelvek és paradigmák* (hétfő 17:45–19:15, PTI mesterszak, Szoftvertechnológia szakirány, kötelező), az *Imperatív programozás* (kedd 10:15–11:45, PTI alapszak, kötelező) és a *Konkurens programozás* (szerda 12:15–13:00 és 13:15–14:00, PTI alapszak, kötelező) kurzusok előadásait tartom, melyek aktuális előadásait fel tudom kínálni a Szakmai Bíráló Bizottság számára. Amennyiben ezek az előadások nem felelnek meg, vagy nem illeszthetők be a Bizottság munkarendjébe, a jelen dokumentumban vázolt habilitációs előadások valamelyikét örömmel megtartom.

## Tudományos előadás

### Statikus forráskódelemzés és -transzformálás

Előadásom 3 fő részből áll majd. A fő hangsúlyt a második részre helyezem. Az első rész előkészíti a másodikat, míg a harmadik részben nagyon vázlatosan mutatok csak be eredményeket.

#### 1. Refaktoráló eszközök fejlesztése

- 1.1. Elemző és refaktoráló infrastruktúra
- 1.2. Változókötések elemzése
- 1.3. Dinamikus nyelvi elemek elemzése
- 1.4. Lexikális és szintaktikus információk

#### 2. Párhuzamosító átalakítások

- 2.1. ParaPhrase Refactoring Tool for Erlang
- 2.2. *Oszd meg és uralkodj* algoritmusok párhuzamosítása
  - 2.2.1. Mintafelismerés
  - 2.2.2. Refaktorálás

#### 3. Egyéb eredmények

- 3.1. Típusozás
- 3.2. Erőforráselemzés és optimalizáció
- 3.3. Alkalmazásiterület-specifikus nyelvek és speciális könyvtárak
- 3.4. Programok szemantikája és helyessége

# 1. felkínált habilitációs előadás

## Az üres referencia okozta probléma és elkerülése

Ez az előadás a *Programozási nyelvek és paradigmák* kurzuson szokott elhangozni. Ez a kurzus kötelező a PTI mesterszak Szoftvertchnológia szakirányán tanuló hallgatók számára. A kurzuson elsősorban az Eiffel nyelvre koncentrálna, de a példákban sok más programozási nyelvet is érintve tekintjük át a programozási paradigmákat, különös tekintettel az objektumelvű és a szerződésalapú programozásra.

Ebben az előadásban az üres (azaz a „sehova sem mutató”) referenciák által okozott problémát tekintjük át, és megnézzük, hogy az egyes nyelvek milyen eszközöket biztosítanak a probléma kezelésére, esetleg elkerülésére.

- Idézet Tony Hoare-tól (2009)
- Üres referencia
- Segfault versus kivétel
- Informatívabb hibaüzenet (JDK 14)
- Statikus elemzés
- Annotáció
- Ada: subtype
- Eiffelben régen: szerződésekkel
- Fordítási időben garantált Void-biztonság
- Cecil
- Eiffel
- Átmeneti és jelenlegi szintaxis
- Eiffelben statikus típusellenőrzéssel
- Altípusosság
- A kétfajta típus együttélése
- Kotlin (és hasonló)
- Ajánlott gyakorlat a null elkerülésére
- A probléma részben megmarad
- Monadikus stílus
- Funkcionális stílus
- Swift

## 2. felkínált habilitációs előadás

### Szálak és feladatok

Ez az előadás a *Konkurens és párhuzamos programozás* kurzuson szokott elhangzani. Ez a kurzus kötelezően választható a PTI mesterszak Szoftvertechnológia szakirányán tanuló hallgatók számára. A kurzuson a Java nyelvet használjuk a konkurens és párhuzamos programozást támogató szokásos nyelvi elemek bemutatására. Részletesen tárgyaljuk a Java nyelv memóriamodelljét is. Támaszkodunk a PTI alapszakon tanult ismeretekre (főleg a *Programozási nyelvek* és a *Konkurens programozás* tárgyakra), valamint a „Java Concurrency in Practice” könyvre.

Ebben az előadásban azt a kérdést járjuk körül, hogy egy feladatot hogyan célszerű párhuzamosan végrehajtható részfeladatokra bontani: hogyan lehet függetleníteni a részfeladat fogalmát a végrehajtási szálaktól.

- A feladat fogalma
- Executor
- Megvalósítások
- Életcikluskezelés
- Feladatok kiosztása
- Időzített feladatok
- Köteget feladatok
- Párhuzamos számítások: ForkJoinTask
- ForkJoinPool
- Oszd-meg-és-uralkodj algoritmusok
- Egy tipikus vizsgafeladat

### 3. felkínált habilitációs előadás

#### Szinkronizációs feladatok

Ezt az előadást a *Programozási nyelvek (Ada)* kurzus számára fejlesztettem. Jelenleg ezt a nyelvet egy kötelezően választható tárgyban tanítjuk. Ebben az előadásban a nyelv konkurens és párhuzamos programozást támogató eszközeivel foglalkozunk, kiemelten a **select**-utasítással.

- Kritikus szakasz
- Szemafor
- Blokkoló utasítás
- Szemafor Ada taszkokkal megvalósítva
- Probléma a szemaforral
- Kölcsönös kizárás
- Monitor
- Egy monitor Ada taszkkal
- Monitorok és az Ada
- Monitor versus szemafor
- A select utasítás
- Többágú hívásfogadás
- Több műveletes monitor
- Végtelen taszk
- Termináltatás
- A terminate használata
- A monitor javítva
- Feltételhez kötött hívásfogadás
- A feltétel használata
- Általánosított szemafor
- A randevú ideje
- Időhöz kötött hívásfogadás
- Timeout
- Rendszeres tevékenység
- Nem blokkoló hívásfogadás
- Busy waiting
- Időhöz kötött hívás
- Nem blokkoló hívás
- Aszinkron select