

A PROGRAMOZÁSI TECHNOLÓGIA TANTÁRGY OKTATÁSA A GÁBOR DÉNES FŐISKOLÁN

THE TEACHING OF TECHNOLOGY OF PROGRAMMING IN DENNIS GABOR COLLEGE

Kaczur Sándor¹

Összefoglaló: A Gábor Dénes Főiskola mérnök-informatikus képzésének első évfolyamának tantárgyai között található a Programozási alapok és Programozási technológia.

A szerző hosszú évek óta tanítja e két tantárgyat; a 2010/2011-es tanév őszi félévében az alapozó Programozási alapok tantárgyban elért eredmények alapján kiválogatott hallgatóknak tartott laborgyakorlatot a tavaszi félévben a ráépülő Programozási technológia tantárgyból.

A cikk ismerteti a két tantárgy tematikáját, az egyes témakörök főbb kapcsolódási pontjait, a hallgatók kiválogatásának szempontjait, a követelményeket, a vizsgáztatás menetét, a félévi munka menetét, a kiadott házi feladatokat, a hallgatók projektmunkáit, végül az elért eredményeket is.

A szerző bemutatja, hogyan menedzselte a félév folyamán a hallgatókat az ILIAS e-learning keretrendszerben, valamint összefoglalja e laborgyakorlathoz kötődő módszertani tapasztalatait és javaslatait.

Kulcsszavak: felsőoktatás, programozás, oktatás, e-learning

Abstract: The subject Basic of Programming and the subject Technology of Programming are of the first year of the Engineering in Information Technology training in Dennis Gabor College.

The author has been taught for many years in these two subjects. There are few students, who have achieved good results in the Basic of Programming in the fall semester of 2010/2011 academic year. The author led them course subject Technology of Programming in the spring semester of that year.

This article describes the topics of course of two subjects, the connections of topics, the student-selection criteria, the requirements, the testing/examing process, the semi-annual session, the homework tasks for students, the project works, and finally the results achieved as well.

The author shows how the managed of the students during the ILIAS e-learning framework (LCMS) and a summary of this methodology lab related experiences and suggestions.

Keywords: higher education, programming, teaching, e-learning

1. Az alapozó programozás tantárgyak célja

A Gábor Dénes Főiskola mérnök-informatikus képzésén az első és második szemeszterek egymásra épülő alapozó programozás tantárgyainak neve Programozási alapok és Programozási technológia. Mindkét tantárgy célja azonos.

Az alapozó programozás tantárgyak célja:

- megismertesse a programozás alapfogalmait,
- megismertesse az alapvető strukturált és objektumorientált programozási technikákat,
- megismertesse egy programozási nyelv és osztálykönyvtár filozófiáját,
- tervezési és programozási stílust alakítson ki.

¹ Gábor Dénes Főiskola, Informatikai Intézet
kaczur@gdf.hu

A hallgató legyen képes:

- egyszerűbb programokat strukturált és objektumorientált módszerrel megtervezni és megvalósítani konzolos, majd grafikus környezetben,
- egyszerű objektumorientált programok szerkezetének és funkcióinak, illetve a megoldás kivitelezési lépéseit megtervezni,
- a feladat és a megoldási elképzeléseket dokumentálni, szabványosan leírni,
- alapelgoritmusokat használni,
- biztonsággal használni egy integrált fejlesztői környezet alapelemeit.

Programozási nyelvként a Java SE-t (JDK 6u25), fejlesztői környezetként a NetBeanst 7.0-t, dokumentálásra UML-t használjuk.

2. Az alapozó programozás tantárgyak tematikája

A két tantárgy tematikája szorosan kapcsolódik egymáshoz. Az MI szakon a Programozási alapok tantárgy témaköreinek sorszáma 1-11., a Programozási technológia esetén ugyanez 12-19. A tananyag elektronikus változata SCORM-os formában az ILIAS LCMS rendszerben (<http://ilias.gdf.hu>) elérhető. Két ajánlott tankönyvet is használunk (Kaczur 2009, Kaczur 2010), két elektronikus melléklettel: JavaPA.zip, JavaPT.zip.

Az első fejezet általános elméleti jellegű bevezetést tartalmaz. A második fejezet végigvezet a telepítésen, a Java szoftveres konfigurációján át a HellóVilág.java-ig. A harmadik fejezettől kezdve a témakörök felépítése a következő: elméleti összefoglaló, típusfeladatok (feladatspecifikáció, tervezés vagy UML ábra, Java forráskód, részletes elemzés), gyakorló feladatok megoldásokkal.

A 15. témakör feldolgozásáig konzolos felületen dolgozunk, a 16-19. témakörben pedig egyaránt készítünk konzolos és grafikus felületű programokat. Az egyes témakörök az 1. ábrán láthatók.



1. ábra – Az alapozó programozás tantárgyak témakörei

A programozási tételek oktatását kulcsfontosságúnak tartjuk. Adatszerkezet nélkül – végleges feldolgozással – már a 6. témakörben előkerül a hat elemi programozási tétel. A 8. témakörben tárgyaljuk újra – primitív típusú elemeket tartalmazó tömböt használva – az elemi programozási tételt valamint előkerülnek a másolás, kiválogatás, szétválogatás összetett programozási tételek is. A 11. témakörben kerül sor a halmazt és/vagy rendezettséget igénylő összetett programozási tételekre: unió, metszet, összefuttatás. A halmazt ekkor még tömbbel valósítjuk meg.

A 12 programozási tétel visszaköszön a 13. témakörben, amikor már a kollekcio keretrendszerben dolgozunk (Vector, HashSet, TreeSet, List). Végül a 19. témakör esetén szintén újratárgyaljuk a 12 programozási tételt, de ekkor már állományokat és feldolgozásukra főként objektumfolyamokat használunk.

3. Az alapozó programozás tantárgyak időbeosztása

A nappali és távoktatás tagozaton jelentős eltérés van a kontakt órák számában. A továbbiakban a nappali tagozatra térek ki. Az őszi félévben futó Programozási alapok tantárgy kontakt órái 30 óra előadásból és 30 óra géptermi gyakorlatból állnak, amely megegyezik a tavaszi félévben tartott Programozási technológia tantárgyai kontakt óráinak számával. A két tantárgy előadásai mindig a szorgalmi időszak első hetében kezdődnek, heti 3 órában 10 hétig tartanak. A gyakorlatok a szorgalmi időszak második hetében kezdődnek, szintén heti 3 órában 10 hétig tartanak. Így 4 hetük van a hallgatóknak a szorgalmi időszak végén, amelyet elővizsgákra igénybe tudnak venni. Mivel sok-sok önálló, otthoni gyakorlás is szükséges a tananyag témaköreinek elsajátításához, így a 6 kredit = 180 egyéni munka elve alapján az időbeosztás megfelelőnek bizonyult.

A tantárgyak vezetőtanáraként a kontakt óráim 2/3 része előadás, 1/3 része géptermi gyakorlat szokott lenni. A 2010/2011-e tanév tavaszi félévében 7 gyakorlati kurzusra osztottuk az elsőéves hallgatókat, amelyből az egyiket én tartottam. A továbbiakban az általam tartott gyakorlati kurzust fogom bemutatni.

4. A hallgatók kiválogatása a gyakorlati kurzusra

A gyakorlati kurzusra vonatkozó jelentkezési időszakot megelőzően elkészült az őszi félév tantárgyainak vizsgaeredményeit tartalmazó statisztika. Kiválogattam a Programozási alapok tantárgyból jó és jeles osztályzatot kapott hallgatókat, valamint szempont volt az Analízis I. és A számítástudomány alapjai tantárgyak vizsgaeredménye is.

21 hallgatót választottam ki a gyakorlati kurzusomra. Az első előadáson mindössze bemutattam a névsort és kértem, hogy a névsorban szereplő hallgatók egy adott időpontra meghirdetett gyakorlati kurzusra jelentkezzenek és járjanak. Az első gyakorlaton ismertettem a kiválasztás szempontjait, a feltételeket, a követelményeket, a számonkérés módját, a félévi munka tervezett menetét. A hallgatók egymás között elit csoportnak hívták ezt a gyakorlati kurzust.

5. A gyakorlati kurzus félévi tervezett munkája

Az előadással szinkronban haladva, a 12-19. témakörökhöz kötődően hétről-hétre, előre ütemezetten haladtunk a tervezési és gyakorlati feladatokkal, valamint a 10 gyakorlati alkalom mindegyikén kaptak házi feladatot a hallgatók. Összesen 15 házi feladatot terveztem. Egységesen mindegyikért legfeljebb 10 pontot lehetett szerezni. Legalább 10 házi feladatot el kell készíteni és 100 pont felett jelest adtam. Mindegyik házi feladatnak volt beadási határideje, amit többnyire rugalmasan kezeltem. Házi feladatot indokolt esetben utólag is engedtem pótolni.

A kiadott házi feladatok öt csoportra oszthatók:

- önálló feladat: konkrét vagy rugalmasan megfogalmazott feladatspecifikációt tartalmazó programozási feladat objektumorientált módon való megtervezését követően Java nyelvű forráskód, elemzés önálló elkészítése és publikálása,
- csoportmunka: 2-3 fős csoportban dolgozva a fenti – ebben az esetben nagyobb – feladat és ennek bemutatása a csoport előtt,
- továbbfejlesztés: korábbi feladat adott szempont szerinti kiegészítésének, továbbfejlesztésének megvalósítása és publikálása,
- korábbi hallgatók által beadott vizsgafeladatok elemzése, tanulási javaslatok összeállítása,
- tesztkérdés, kódolási feladat készítése: önállóan egy konkrét témakörhöz kötődően tesztkérdés, kódolási feladat készítése, szerkesztése és publikálása.

A házi feladatok értékelésének szempontrendszerét a hallgatók közösen állították össze, csupán ezek megfelelő csoportosítására volt szükség.

A kurzus hallgatói egymással és a gyakorlatvezetővel az ILIAS-on keresztül tartják a kapcsolatot. Erre a célra létrehoztam egy csoportobjektumot, amelyhez az első alkalommal megadott jelszóval csatlakoztak a hallgatók, ezután a csoportot lezártam. A lezárt csoportot csak a tagok látják, így egy kiváló, jogosultsággal, szerepkörrel jól definiálható, rugalmasan szerkeszthető, fórumot és chatet is tartalmazó felület áll rendelkezésre. A csoportobjektum megtekinthető, a szükséges azonosító és a jelszó az IF2011 Konferencia szekció előadásán elhangzott.

6. A gyakorlati kurzus házi feladatai

A házi feladatok az ILIAS közösen szerkeszthető elektronikus tananyagobjektumában jelentek meg (2. ábra). A tananyagobjektum ideális megoldás a hasonló típusú feladatok megoldására. A feladatok tanár általi értékelése egy pontszámból állt, amely ha nem maximális, akkor részletes indoklást is tartalmazott, esetleg javaslatot a kiegészítésre, továbbfejlesztésre.

1. feladat:

- Cél: szakértő és vezérlő osztályok együttműködésének gyakorlása objektumok tömbben való tárolásával megvalósítva (2. ábra).

The screenshot shows the ILIAS web interface. On the left is a navigation tree with 'Tartalomjegyzék' (Table of Contents) and 'Feladatok' (Tasks) expanded. The main content area displays the details for '1. feladat'. The task text is as follows:

1. feladat

A program szakértő osztályt definiálja egy objektum tárolásához szükséges adatokat és metódusokat!

Például:

- Könyv (cím, szerző, ár, kiadó),
- ÁltalánosHáromszög (a, b, c),
- Joghurt (fajta, gyártó, ár, szövettség).

Értelemszerűen:

- kell a tulajdonságokhoz a megfelelő láthatóság,
- szükségesek a get, set, toString, equals, compareTo metódusok.

A konstruktor végezze el a szükséges ellenőrzéseket, ne engedjen hibás esetleg hiányzó adatokkal objektumot létrehozni! Ha kell, erőszakkal dobjon kivételt!

A program vezérlő osztályának tömbje tartalmazzon néhány objektumot! Ha lehet véletlenszerűen generálni a konstruktorok paramétereit, akkor 20-30 objektum legyen; egyébként kézzel megadva minimum 5 db.

A feladat valósítsa meg valamilyen programozási tételt! Ha ehhez kell, akkor két/három tömböt is létre lehet hozni.

A feladat bonyolultságától, összetettségétől függően a vezérlő osztály kezelheti a szakértő konstruktorában keletkező kivételeket.

Ötlet:

- JavaPA.zip/fejzet09/Háromszög?.java
- JavaPA.zip/fejzet09/TengerimalacEgyNapja.java
- JavaPA.zip/fejzet08/"Tétel".java
- JavaPA.zip/fejzet11/OsszetettProgramozasiTetelek.java
- JavaPT.zip/fejzet12/"".java

Határidő: 2011.02.17. 5:00

2. ábra – A gyakorlati kurzus 1. házi feladata

2. feladat:

- Cél: az 1. feladat továbbfejlesztése kollekción adatszerkezetre.
- Másik csoporttárs 1. feladatát kell módosítani úgy, hogy az 1. feladat értékelését figyelembe kell venni; meg kell egyezniük egymás között, ketten nem csinálhatják ugyanazt!
- A feladatnak tartalmaznia kell tömb helyett valamilyen kollekción a feladat jellegének megfelelően; a szakértő osztály konstruktorában, vagy setter metódusaiban kivételkezelést; programozási tételt.

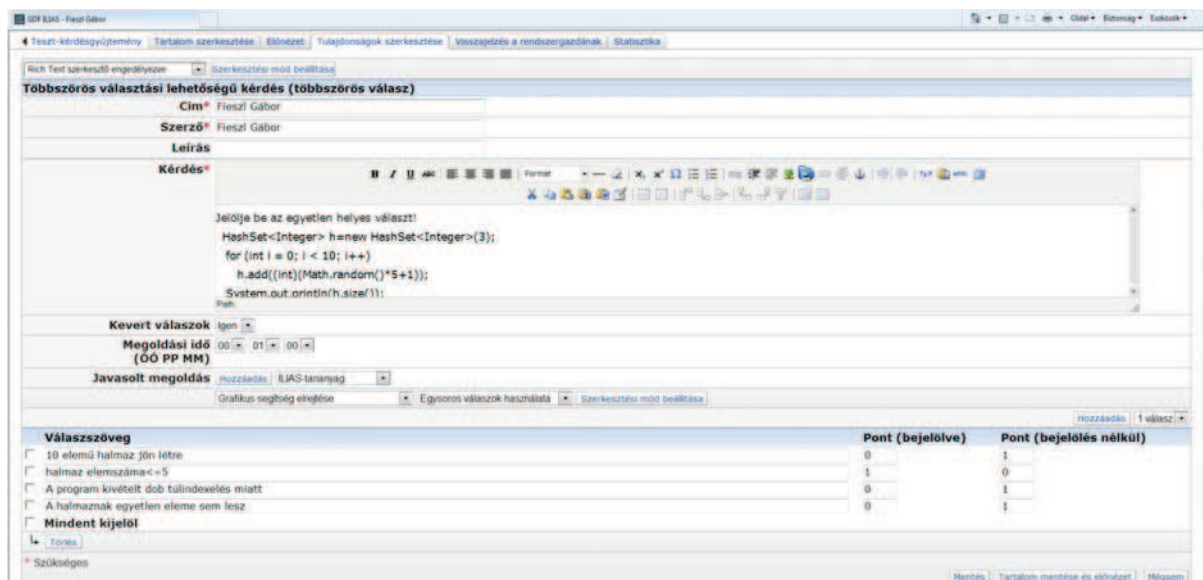
3. feladat:

- Cél: egy adott programozási tétel alkalmazása többféle adatszerkezettel.
- A program specifikációja és forráskódja legyen rövid, egyszerű, közérthető és tartalmazzon pontosan egy programozási tételt!

- A programnak legyen háromféle megoldása: adatszerkezet nincs, ciklus van, végjeles feldolgozás vagy véletlenszerű bemenet (pl.: JavaPA.zip/fejezet06/NéhánySzámÖsszege1.java); tömb adatszerkezet (pl.: JavaPA.zip/fejezet08/ElemiProgramozásiTételek.java egy-egy metódusa); kollektív adatszerkezet (pl.: JavaPT.zip/fejezet13/ÖsszetettProgramozásiTételek?.java).

4. feladat:

- Egy tesztkérdést kell készíteni a kollektív témakörből!
- A kérdés szövegében legyen egy rövid (adatszerkezet deklarációját is tartalmazó) forráskódrészlet (pl.: a main() metódus belseje), amelyre vonatkozóan 4 válaszlehetőség készüljön el. A négy válaszlehetőség ne mondjon egymásnak ellent, ne legyen közöttük összefüggés. A válaszlehetőségek vonatkozhatnak az adatszerkezet tulajdonságaira, az elemekre, a bennük lévő objektumok adataira, jellemzőire. Legyen legalább 1 és legfeljebb 3 helyes válasz a válaszlehetőségek között.
- A gyakorlaton bemutatott módon kell elkészíteni a feladatot (3. ábra)!



3. ábra – A teszt-kérdésgyűjtemény többszörös választási lehetőségű kérdésének tulajdonságai

5. feladat:

- Cél: a gyakorlaton elkészített feladat továbbfejlesztése öröklődéssel.
- Ki kell egészíteni/javítani az Ember osztály funkcióit. Le kell származtatni az Ember osztályból a Tanár osztályt, amelynek új adataja a String tantárgy[], egyben ez a konstruktorának utolsó paramétere is; új/felüldefiniált metódusai: vizsgáztat(), tanít(), beszél(). A Személyek vezérlő osztály vektorába vegyessen kell embereket, diákokat, tanárokat tenni! Ebbe az osztályba írni kell egy metódust, amely a személy generikus vektorban lévő objektumokra alkalmaz egy összetett programozási tételt!

6. feladat:

- Cél: korábbi hallgatók által beadott konzolos vizsgafeladatok elemzése, tanulási javaslatok összeállítása.
- Kódolási feladat: A program olvasson be a konzolról egy szöveget. Számolja meg és írja ki, hogy a megadott szövegben hány A, B, C, ..., Z betű van. Csak azokat a betűket írja ki,

amelyekből legalább egy van. A magyar ékezetes betűket nem kell számolni. A kis- és nagybetűk ne különbözzenek. Példa: Szöveg: alma A: 2, L: 1, M: 1

- Ez a kódolási feladat sokféleképpen megoldható attól függően, hogy a vizsgázó hallgató milyen szinten tart, hogyan gondolkodik, milyen adatszerkezetet használ (tömb, objektum, kollekció, fájl), milyen programozási tételt alkalmaz.
- Ki kell választani a feltöltött hibás megoldások közül valamelyiket! Ki kell javítani, vagy adni kell egy olyan megoldást, ami a többiek megoldásától elvben különbözik!
- Le kell írni: mit nem tud az a hallgató, aki erre a kódolási feladatra az adott megoldást adta; mit (esetleg milyen sorrendben) kell megtanulnia, hogy meg tudja oldani ezt a feladatot; hogyan segítenének a hallgatónak, aki az adott hibás megoldást adta; tessék belekötni a legapróbb dolgokba is!

7. feladat:

- Cél: játék fejlesztése csoportmunkával.
- A feltöltött minták alapján 2-3 fős csoportmunkával kell tetszőleges játékot készíteni! A játék alkalmazáslogikájában lehet ember-ember közti, vagy ember-gép közti játszmát is "leprogramozni". A játék swings alkalmazás legyen, kizárólag egyszerű komponensek felhasználásával készüljön! A következő gyakorlati órán a csapatoknak be kell mutatni a feladat tervét, a megvalósítás lépéseit, össze kell foglalni a csapatmunkához kötődő tapasztalatokat, meg kell mutatni a működőképes programot és annak forráskódját is. Az értékelés közösen történik majd.
- A csoportmunka esetén kiemelten fontos az egyéni felelősség meghatározása, az építő egymásrautaltság időbeli és részfeladat szintű kezelése, továbbá törekedni kell az egyenlő arányú részvételre (Kagan 2004).

8. feladat:

- Cél: grafikus felületű oktatóprogramot kell készíteni!
- Az oktató/demo program swings alkalmazás legyen, kizárólag egyszerű komponensek felhasználásával készüljön! A program valamilyen elemi vagy összetett programozási tételt, ezek egymásra épülését, valamilyen adatszerkezet vagy algoritmus/módszer működését szemléltesse/oktassa. A feltöltött fájlban található a heteslottó oktatóprogramja, amely a logikai tömbös megvalósítást szemlélteti. Ötletek: 10 konzolos demo program van a JavaPA.zip fájlban (*Demo.java), ezek könnyen swingsíthetők; Tenger Iván informatika oktatóprogramjai: <http://tenger.web.elte.hu/flash/index.htm>; <http://oktatoprogram.linkcenter.hu>.

9. feladat:

- Cél: elméleti rész számonkérése.
- A Programozási technológia tantárgyhoz kapcsolódó tesztkérdésekből lesz a következő gyakorlaton egy helyben, segédeszközök nélkül kitöltendő teszt 20 db tesztkérdéssel. A maximális pontszám 10 lesz, az ILIAS által automatikusan adott pontszám arányában. Erre kell tréningezni a héten.

10. feladat:

- Cél: korábbi hallgatók által beadott swings vizsgafeladatok elemzése, tanulási javaslatok összeállítása. A részfeladatok megegyeznek a 6. feladatnál leírtakkal.

11. feladat:

- Cél: egy tesztkérdést kell készíteni a swings témakörből! A részfeladatok megegyeznek a 4. feladatnál leírtakkal.

12. feladat:

- Cél: korábbi feladat továbbfejlesztése.
- A csapatban elkészített 7. feladatot (játékprogramot) kell kiegészíteni úgy, hogy fájlkezelést is tartalmazzon. Pl.: toplista, eredmények, statisztika, történet a játékokról.

13. feladat:

- Cél: egy tesztkérdést kell készíteni a kivételkezelés témakörből! A részfeladatok megegyeznek a 4. feladatnál leírtakkal.

14. feladat:

- Cél: egy tesztkérdést kell készíteni a fájlkezelés témakörből! A részfeladatok megegyeznek a 4. feladatnál leírtakkal.

15. feladat:

- Cél: kódolási feladat készítése.
- Egy Programozási technológia vizsgán bevethető kódolási feladatot kell készíteni: viszonylag legyen egyszerű; ésszerűen rövid legyen, hogy a megoldása beleférjen a vizsga idejébe; a tananyaghoz tartozó témakörökhöz kapcsolódjon; egyértelműen megfogalmazott feladat-specifikáció legyen benne (olyan, amit ha megkap a hallgató a vizsgán nem tud kérdezni); a forráskód legyen beszédes, következetes jelölésrendszerrel, szép változó-, metódus-, osztálynevekkel.

7. Tapasztalatok

- A hallgatók kellően érdeklődtek és motiváltak voltak.
- A hallgatók igyekeztek a házi feladatok határidejét betartani. Amikor más tantárgyakból is számonkérés volt (általában a szorgalmi időszak 5. és 10. hetében), akkor előfordult, hogy késtek a beadással.
- A hallgatók számára nem jelentett problémát használni az ILIAS rendszer tananyagszerkesztő és tesztkészítő interaktív felületét. Egy tipikus házi feladat szerkesztés közben látható a 4. ábrán. Csupán az egységesség kedvéért került sor egy-egy példa bemutatására a kurzuson.



4. ábra – Egy házi feladat megjelenési formája szerkesztés közben

- Az ILIAS korábban említett objektumai kiválóan alkalmasnak bizonyultak a gyakorlati kurzuson való használatra, egyben a csoportmunka támogatására is.
- A csoportmunka során többféle webkettes szolgáltatást is igénybe vettek a hallgatók, például: közösségi oldalak, szakmai blogok és fórumok, linkmegosztó szolgáltatások, online tárhelyszolgáltatók, online irodai alkalmazások, SCORM szerkesztők, Skype, Windows Live Messenger, join.me szolgáltatás. A párhuzamos interakciók így könnyebbnek bizonyultak.
- A házi feladatok elkészítése során – feladattól függően – szükséges lehet koordinálni: ötletek egyeztetése, megvalósítási módszer kiválasztása, dokumentáció részletességének eldöntése, „jó lesz-e ez a feladat”. Ezekre alkalmas a fórum, amelyben mindez utólag könnyen nyomon követhető, az automatikus e-mailbeli értesítési funkciónak köszönhetően egy adott felmerült kérdést többnyire elegendő egyszer megválaszolni. A hallgatók megfelelően használták a kurzushoz tartozó fórumot, egymás hozzászólásaira is aktívan reflektáltak. Ez kellően gyors válaszokat és időbeli rugalmasságot igényel a tanártól.
- Kiemelten fontos, hogy a hallgatók és a tanár világosan értse szerepét a tanítási-tanulási folyamatban. A hallgató és a tanár egyenrangú partnerek, közös a cél. A tanár már régen nem két lábon járó lexikon, aki mindent tud és nála jobban azt senki sem tudja, hanem minden korszerű technológiát bevetni kész hozzáértő, kiválóan felkészült, naprakész szakember, aki szervezi, koordinálja, irányítja a folyamatot.
- A hallgatók összesített véleménye: praktikus, hogy egyenletes a terhelés a félév során; aki szorgalmas volt, az szerezhethet megajánlott jegyet a kurzus végén, így nem kell ezzel a tantárggyal foglalkoznia a szorgalmi időszak végén és a vizsgaidőszakban; reális az értékelés és egymás beadott házi feladatait is elérik, így könnyű az összehasonlítás; a csoportmunka tipikus nehézségei ellenére jól sikerültek a házi feladatok, sokat veszítettek azok a hallgatók, akik nem erre a gyakorlati kurzusra jártak.
- A kooperatív csoportmunka szemléletmódja újdonságot jelentett a hallgatók számára.
- Összességében kiemelkedően pozitív a mérleg. A gyakorlati kurzus koordinálása nagyon kreatív feladat. A beadott házi feladatok sokfélék, nincs két egyforma, így rengeteg időt igényel az értékelésük.

8. Eredmények

A gyakorlati kurzusra járó 21 hallgató közül 18 kapott jeles osztályzatot. Szinte minden hallgató minden házi feladatot elkészített, erre egymást is motiválták. A hallgatók kiválogatása megfelelőnek bizonyult. Bízom abban, hogy a továbbiakban is hatékonyan együtt tudunk működni, ezek lehetőségei: Web-programozó Diákműhely, Mobil Alkalmazásfejlesztő Diákműhely, TDK munka, regionális és országos tanulmányi versenyek, szoftverfejlesztéshez kötődő szakmai gyakorlat, szakdolgozati konzultáció.

9. Köszönetnyilvánítás

Szeretnék köszönetet mondani az ILIAS munkacsoportban dolgozó azon kollégáknak, akik segítettek az elmúlt évek során a keretrendszer megismerésében: Berecz Antónia, Kállai Miklós, Littvay László. Köszönöm azon oktató kollégáknak, akik a félév során online vagy óralátogatás során követték az eseményeket és ötleteikkel segítettek: Lengyel Borisz István, Nádai Gábor. Továbbá köszönöm a gyakorlati kurzusra járó hallgatók aktív közreműködését.

Irodalomjegyzék

- Kaczur, S., 2009. Programozási alapok, Budapest, ISBN 978-963-06-8122-3
Kaczur, S., 2010. Programozási technológia, Budapest, ISBN 978-963-06-8628-0
Kagan, S., 2004. Kooperatív tanulás, ÖNKONET, Budapest, ISBN 9632166598