

**SZÉCHENYI ISTVÁN EGYETEM**

**JEDLIK ÁNYOS GÉPÉSZ-, INFORMATIKAI ÉS  
VILLAMOSMÉRNÖKI INTÉZET  
INFORMATIKA TANSZÉK**



A féléves programozási feladatok  
készítésének  
általános szabályai

**INFORMATIKA TANSZÉK**

**2011**

## Tartalomjegyzék

---

<b>1. BEVEZETÉS.....</b>	<b>1</b>
<b>2. FORMAI ELŐÍRÁSOK.....</b>	<b>2</b>
<b>3. A PROGRAMFEJLESZTÉSI MUNKA DOKUMENTÁLÁSA.....</b>	<b>6</b>
<b>4. A DOKUMENTÁCIÓ TARTALMA.....</b>	<b>7</b>
<b>5. MUNKASZERVEZÉS ÉS ELLENŐRZÉS.....</b>	<b>10</b>
<b>6. MELLÉKLET.....</b>	<b>11</b>

### 1. Bevezetés

A jelen dokumentum célja, hogy egységes, mind a formai, mind a tartalmi szempontokra kiterjedő szabályozást adjon a *programozási tantárgyakhoz* előírt féléves programozási feladatokra vonatkozóan.

Célja, hogy a feladatok elkészítésénél és dokumentálásánál felvetődő gyakori kérdésekre, problémákra egységes, írásban is rögzített választ adjon és a feladatmegoldások - mind formai megjelenésükben, mind tartalmi felépítésükben, szerkezetükben - minél egységesebb, az itt folyó szakképzésre jellemző képet mutassanak.

Jelen szabályzat *nem foglalkozik* a konkrét feladat-kiírásokkal, és a munkaszervezési, határidőzési és egyéb előírásokkal is csak általánosan. Minden *tárgyban* és *félévben* külön kiegészítő dokumentumban közöljük az aktualitásokat.

Előrebocsátjuk még, hogy a dokumentáció, mint kiadvány megszerkesztésére vonatkozó formai előírásokat, ahol ez a pontosság és egyértelműség kedvéért szükséges, a Microsoft Word, (OpenOffice, vagy LibreOffice) mint ebben a feladatkörben jelenleg leginkább használt kiadványszerkesztő terminológiájában fogalmazzuk meg. Ez természetesen nem kötelező, más, legalább ilyen szolgáltatási szintű kiadványszerkesztő is használható, a követelmények értelemszerű alkalmazása mellett.

A jelen dokumentum is a benne lefektetett formai szabályok szerint készült, ilyen értelemben mintának is tekinthető.

## 2. Formai előírások

### 2.1. A dokumentáció szerkezete

#### 2.1.1. Felépítés

A dokumentáció **A4** szabványméretű lapokra egy-, vagy nagyobb terjedelem esetén kétoldalasan nyomtatandó ki.

Felépítése:

- *Címlap*: a jelen dokumentáció mellékletében adott minta szerint.
- *Tartalomjegyzék*.
- *Bevezetés* (fő fejezet).
- *Tervezési dokumentáció* (fő fejezet).
- *Telepítési útmutató* (fő fejezet)
- *Felhasználói útmutató* (fő fejezet).
- *Tesztelés* (fő fejezet).
- *Melléklet* (fő fejezet) és lemezmelléklet.

A dokumentációt egy műanyag dossziéba kell lefűzni. A fedőlap mindkét esetben átlátszó legyen. A *lemezmellékletet* a hátsó borítólapon belülről rögzített nyitható - zárható borítékban, vagy a kereskedelemben kapható, kimondottan erre a célra készített tokban kell elhelyezni.

Hangsúlyozzuk, hogy mindenfajta formai jegynek a dokumentáción belül azonosnak kell lenni.

#### 2.1.2. Fejezetek, címek, bekezdések

A dokumentáció tartalmi része - hierarchikus szerkezetben - *fejezetekre* tagolandó. Minden fejezetnek címe van. A címeket külön sorban kiemelve, központosított decimális arab számozással ellátva kell feltüntetni. A címszámozás adja meg a szöveg hierarchikus szerkezetét. Áganként a címszámozás folyamatos.

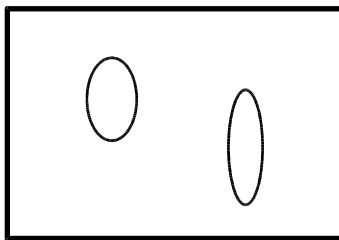
A *fő fejezeteket* egyjegyű, 1-gyel kezdődő sorszámmal kell ellátni, ezeket mindig új oldalon kezdjük. Az alsóbbrendű fejezetekre ez már nem áll, és általában törekedjünk az oldalak kitöltöttségére.

Törekedjünk arra, hogy a cím rövid, egy sorban elférő legyen. Cím előtt és után egy üres sornyi térköz kell hagyni. A cím (a számával együtt) lehet balra, vagy középre igazított. Cím nem állhat egy oldal utolsó soraként. Törekedjünk arra, hogy a cím után még minél több sor legyen az oldalon.

A fejezetek bekezdésekre tagolódnak. A bekezdések közt egy üres sornyi térköz legyen.

#### 2.1.3. Számozás

A címlapot nem kell *oldalszámozni*. Ha a *tartalomjegyzék* egyoldalas, akkor nem kell számozni, ha többoldalas, akkor *római* számokkal számozott. *Folyamatos arab* számozással látandó el a többi rész. Az oldalszám a láblécben, középen vagy jobbra igazítva helyezendő el.



**1. ábra. Példaábra**

Az *ábraszámolás* folyamatos sorszámolás. Az ábraszám után ábracím is írandó. Az ábraszám és cím az ábra alá, középre igazítva helyezkedik el (ld. 1. ábra). Egy ábra legfeljebb 1 oldalt foglalhat el.

**1. táblázat. Példatábla**

A	B	C
1 . 23	2 . 34	3 . 45
-2 . 00	56 . 09	2 . 45

A *táblázatokat* csak akkor számozzuk, ha ez belső hivatkozások miatt szükséges. A számozás és címezés rendszere ugyanolyan, mint az ábráké, de itt a táblázat fölé teendő a szám és a cím (1. táblázat). Törekedjünk a legfeljebb 1 oldalas táblázatokra, de ha a táblázat több oldalas lenne, a fejlécsora minden oldalon megismétlendő, valamint a táblázatban is jelezni kell a megszakítást és a folytatást.

$$c^2 \leq a^2 + b^2 \quad (1)$$

A *képleteket* csak akkor számozzuk, ha ez belső hivatkozások miatt szükséges. A számozás rendszere ugyanolyan, mint az ábráké, de itt a képlet mellett a jobb oldalon és zárójelbe téve számozzunk, ld. fentebb az 1. képlet.

### 2.1.4. Fejléc és lábléc

A dokumentáció számozott oldalain alkalmazzunk fejléct. A fejléct vonallal válasszuk el az oldal szövegétől. A fejléc szövege csak egy soros legyen. Ebben tüntessük fel az aktuális fő fejezet címét és esetleg a feladat címét. A fejléc szövege lehet balra vagy középre igazított, vagy két cím esetén a két szélre ütköztetett.

A dokumentáció számozott oldalain alkalmazzunk lábléct. Ebben csak az oldalszám legyen, amely középen vagy jobbra igazítva helyezendő el. Ne használjunk lábjegyzeteket.

## 2.2. Formátum

### 2.2.1. Általános jellemzők

#### 2.2.1.1. Betűk és sorok

Használható bármely *szolid*, a szakkönyvekben általánosan használt, a magyar nyelvű szövegekhez alkalmas betűtípus. Javasolt betűtípus az alapszövegre: Times New Roman. Nem megengedett semmilyen extra, díszítő jellegű (mint pl. a kézírást utánzó) betűtípus. Az alapszöveg **szimpla** sortávolsággal 12-es betűmagysággal írandó.

A bekezdéseket egy üres sornyi térköz választja el. A bekezdés első sora is a bal szélén kezdődjön. A bekezdés igazítása lehet sorkizárt, vagy balra igazított (de csak egyféle).

### 2.2.1.2. Kiemelések, rövidítések

Az alapszövegen belüli kiemelésre használható a *dőlt* betűvel írás, a **vastagítás**, esetleg más betűtípus. A kiemelt szövegrész betűmérete indokolt esetben lehet **kisebb** mint az alapszövegé, de ne legyen nagyobb annál. Ne használjuk az aláhúzásos kiemelést.

A *rövidítéseket*, betűszavakat első előfordulásuknál oldjuk fel, magyarázzuk meg.

### 2.2.1.3. Címek, felsorolások, hivatkozások

A címeket (fejezetcím, fejlécszöveg, ábracím, táblacím) valamilyen módon emeljük ki. A fejezetcímeknél a kiemelés jelenthet (az egyéb lehetőségek mellett) a szintekkel csökkenő betűméretet is, de egyrészt ne alkalmazzunk az alapszövegnél kisebb betűket, másrészt ne haladjuk meg az alapszöveg betűméretének kétszeresét. Azonos szintű fejezetcím mindig azonos formátumú legyen. A fejlécben lévő címek az alapszöveggel azonos betűméretűek legyenek, kiemelési módjuk a vastagítás és/vagy más betűtípus.

A szövegen belüli *felsorolásokhoz* a szövegszerkesztő megfelelő eszközét használjuk. A felsorolást kezdjük beljebb (behúzás) a bekezdés szélénél. Törekedjünk arra, hogy a teljes felsorolás egy oldalon belül legyen, de ne kezdődjön az oldal magával a felsorolással.

A dokumentáción *belüli* hivatkozásokat a hivatkozott fejezet, ábra vagy táblázat számával adjuk meg, kerek zárójelben, pl. (1. ábra).

### 2.2.1.4. Tartalomjegyzék

A tartalomjegyzékben a dokumentációnak a tartalomjegyzéket követő címeit tüntessük fel. A tartalomjegyzék készíthető akár a szövegszerkesztő megfelelő eszközével, akár e nélkül. A könnyebb követhetőség kedvéért olyan formátumot alkalmazzunk, ahol a cím és a hozzá tartozó oldalszám össze van kötve (tabulátorvezető jel: pontozás, vagy vonal).

## 2.2.2. Specialitások

### 2.2.2.1. Programrészek, programelemek

A szöveg között betétként szereplő *programrészek* nyomtatási képe kövesse a programrésznek az eredeti programszövegben meglévő tagolását, szerkezetét. E célból a programrészeket azonos jelszélességű (monospace) betűtípussal és megfelelő betűmérettel írjuk. Javasolt: Courier New típus és 10-es vagy 8-as méret.

Ha szövegben valamely *programbeli azonosítót* vagy programelemet írunk le, ezt mindenképpen emeljük ki, különböztessük meg az alapszövegtől. Javasolt, hogy ugyanolyan betűtípussal írjuk, mint a programrészt.

Ugyanílyan módon (monospace) emeljük ki a *számítástechnikai környezetre* vonatkozó azonosítókat is. Ilyenek például a fájlazonosítók, programnevek, globális fájlazonosítók (pl. \*.TXT).

A nagyobb terjedelmű *programlistát* (pl. a mellékletben) - helytakarékoság és az eredeti programszerkezet megtartása céljából - azonos jelszélességű betűtípussal és kicsi betűmérettel írjuk. Javasolt: Courier New típus 8-as vagy 6-os méretben.

Mind a programlistában, mind a programrészek írásánál kövessük a megfelelő programfejlesztő rendszer szövegszerkesztőjének stílusjegyeit, pl. az alapszavak kiemelése, a kommentárok dőlt betűs írásmódja.

### 2.2.2.2. Képletek, képek

A képleteket, matematikai formulákat a szövegszerkesztő megfelelő eszközével készítsük el. Ha szövegben valamely a képletben szereplő *azonosítót* vagy szimbólumot írunk le, ezt mindenképpen emeljük ki, különböztessük meg az alapszövegtől. Ha lehetséges, ugyanolyan betűtípussal írjuk, mint a képletet. A képletszerkesztőben is javasoljuk valamely azonos jelszélességű betűtípus használatát.

Ne használjunk "sötét" hatású képeket. Ha a kép egy képernyőképből származik és a képernyőn sötét háttérű és a rajta lévő szöveg és/vagy rajz a világos, a dokumentációban inkább az invertált képet szerepeltessük (különösen akkor, ha csak fekete-fehér nyomtatást alkalmazunk).

Ugyanígy ne használjunk olyan ábrát, amely csak nagyon csekély információval bír (tipikus hiba például egy program üres kezdőképernyője). Ebben az esetben csak azt a részletet jelenítsük meg, amelyet a szövegben megemlítünk, illetve egyéb információval bír.

### 2.2.2.3 Szakirodalmi hivatkozások

A feladat elkészítéséhez felhasznált idegen forrásból származó részeket kötelező megjelölni. Az egyes állítások, algoritmusok és tervezési minták és egyéb források felhasználásakor, annak bizonyításaként a hivatkozásokat úgy kell elhelyezni, hogy azok egyértelműek és visszakereshetőek legyenek. A folyó szövegben a hivatkozás szögletes zárójelek között szerepel. Formátuma a műszaki gyakorlatban használt sorszámozott, vagy a szerző nevét és a könyv kiadásának dátumát tartalmazza (pl: [1], [Kernigham 1993]). A dolgozatban csak az egyik, illetve a másik jelölés alkalmazandó, a vegyes használat kerülendő.

Szó szerinti idézet esetén az idézőjel után helyezzük el a hivatkozást, de ekkor az oldalszámot, vagy oldalszám-tartományt is jelölni kell! [2 p.34.] vagy [Flanagan 2005, p.45.], [Botzheim 2008, pp.23-24.]

A dolgozat végén külön „Irodalomjegyzék” főcím alatt a hivatkozott forrásokat a szerzők nevének ABC sorrendjében kell felsorolni (a megfelelő jelölésrendszer alkalmazásával).

[1] Sziray József Dr. - Varga Ágnes: Szakértői rendszerek. Győr : Széchenyi István Egyetem, 2006. 151 p.

[Bauer 2006] Bauer Péter - Hatwágner F. Miklós: Programozás II. Győr : Széchenyi István Egyetem, 2006. 204 p.

Elektronikus formában létező dokumentumok hivatkozásakor meg kell adni a szerzőt, a címet és a dokumentum jellegét (online, CD-ROM, DVD, e-jegyzet), a dokumentum teljes URL-jét, és a letöltés dátumát. Az egyértelmű visszakereshetőség végett a forrásokról egy másolatot a lemezmellékletben kell elhelyezni!

Részletes szabályok az MSZ ISO 690:1991 szabványban találhatók.

### 3. A programfejlesztési munka dokumentálása

A program tervezésénél betartandók az alábbi általános szempontok:

- Az objektumorientált fejlesztési alapelvek betartása.
- Ellenőrzött billentyűzet-képernyő input, menü-orientált, a felhasználót segítő, tájékoztató működés.
- Elsősorban saját készítésű vagy a tananyag részét képező programelemek használandók.
- Preferáljuk a közös (tananyagbeli) elemek használatát, vagyis: ha valamely funkcióra (pl. ellenőrzött input, menü) van közös megoldás, akkor egyéni, vagy külső fejlesztőtől származó megoldás csak akkor fogadható el, ha alkalmazása indokolható.
- Nem tartalmazhat a program a tananyagban szereplő általános célú eszközöknél alacsonyabb színvonalú megoldásokat.
- Ne alkalmazzunk a feladathoz nem szükséges, csak a "látványosságot" célzó megoldásokat (pl. bejelentkező animáció, felesleges kép- és hangeffektusok stb.)
- Törekedjünk a program moduláris felépítésére, az egyes funkciók, funkciócsoportok formailag is különüljenek el.

Maga a kifejlesztett program legyen jól tagolt, áttekinthető. Objektumorientált tervezésnél betartandók az ide vonatkozó alapelvek, szabályok (egységbezárás, öröklődés adta lehetőségek kihasználása, láthatóság megfelelő szabályozása, stb.). Objektumorientált fejlesztésnél törekedjünk az objektumszerkezet és a modulrendszer összhangjára, párhuzamosítására.

Ha a megoldás egy előírt vagy ajánlott alaprendszerre, API-ra épül, akkor a létrejött programban világosan *különüljön* el a saját fejlesztés, ne keveredjen más forrásból származó, nem saját fejlesztésű programelemekkel. A felhasznált forrásokat minden esetben meg kell jelölni. **A dokumentációban a fejlesztőnek nyilatkoznia kell arról, hogy a saját fejlesztésű programkódokat valóban saját maga hozta létre, illetve a külső források eredete a feltüntetettnek megfelelő.** Ezt a nyilatkozatot saját kezű aláírásával hitelesíti.

A programírásnál, elkülönítendő a fejlesztő rendszer által adott programelemektől, az általunk bevezetett programelemeknek adjunk *magyar* nyelvű elnevezéseket, rövidítéseket, lehetőleg olyanokat, amelyek emlékeztetnek az illető elem programbeli funkciójára. Természetesen, ebben alkalmazkodni kell a fejlesztőrendszer szintaktikus szabályaihoz is, például az ékezetek elhagyásával, vagy az írásmóddal.

Külön felhívjuk a figyelmet arra, hogy a program *tesztelését* az ellenőrző jellegű opciók *bekapcsolt* állapotában, a legszigorúbb beállítások mellett végezzük.



### 4. A dokumentáció tartalma

#### 4.1. A bevezetés

A bevezetésben az alábbiakkal foglalkozunk:

- A kitűzött feladat pontos ismertetése.
- A program által nyújtott szolgáltatások tömör, lényegre törő, a technikai részleteket mellőző leírása. Az esetleges korlátozások megadása.
- Az elméleti (pl. matematikai) háttér ismertetése, ha a feladat olyan.
- Az alkalmazási környezet (hardver/szoftver).
- Ha van a tananyagon túli, érdemben felhasznált *szakirodalom*, erre is itt térjünk ki.

#### 4.2. A tervezési dokumentáció

A tervezési dokumentációban adjuk meg a *fejlesztő környezet* specifikációját, ideértve:

- a fejlesztést érdemben befolyásoló *hardver - szoftver* környezet, a *fejlesztő rendszer* meghatározását (gyártó, verzió stb.),
- a *saját fejlesztő* környezet leírását: a programcsomag moduljai, ezek rövid tartalmi jellemzése, a köztük lévő kapcsolatok leírása/ábrázolása, az alkalmazott speciális terminológia (pl. fájl kiterjesztések).

Ha a megoldás egy előírt vagy ajánlott alaprendszerre épül, akkor elsősorban a saját fejlesztéssel kell foglalkozni. Ha a leírás logikai összefüggései megkívánják az alaprendszer bizonyos szintű tárgyalását is, akkor ezen a szinten is világosan különüljön el a saját fejlesztés. Külön jelöljük meg a *nem saját* fejlesztésű szoftver erőforrásokat, a származás helyének feltüntetésével.

Dokumentálandó az objektumrendszer. Minimálisan megkövetelt az *osztálydiagramok* és *osztályleírások* megadása. Világosan látható legyen meg az alaprendszerhez való kapcsolódás, és különüljön el a saját fejlesztés. A lényeges metódusokat részletesen (paraméterezés, funkció) dokumentáljuk. Az osztályrendszer bemutatása a tervezési dokumentáció egyik legfontosabb része!

Amennyiben a tervezés más adatszerkezeteket is tartalmaz, akkor adjuk meg pontos leírással (esetleg rajzos ábrázolással), és a funkció megjelölésével a megoldás lényeges részét képező adattípusokat, *adatszerkezeteket* (fájlleírások, adatbázis, stb.) is.

Ha a megoldásban lényeges szerepet játszó, *saját* algoritmusfejlesztés is áll, akkor szövegesen ismertessük és mutassuk be *struktúradiagrammal*, vagy valamely alkalmasan választott *UML* diagrammal.

Ha az objektumorientált fejlesztés *komponensalapú*, akkor a tervezési dokumentációban jelenjenek meg a formok tervezési állapotai is. Célszerű lehet a tervezési dokumentáció egy részének formonkénti tagolása.

#### 4.3. A telepítési dokumentáció

Abban az esetben, ha a program, vagy programrendszer telepítése nem egyértelmű (pl. több független komponensből áll), akkor ebben a fejezetben kell kifejtetni a szükséges

szoftverkomponensek telepítési és konfigurációs beállításait logikailag egyértelműen követhető formában.

A fejezet részei:

- A programhasználat *hardver* és *szoftver* feltételei (minimális és ajánlott konfiguráció), értelemszerűen a szerver- és kliens oldal szétválasztásával.
- A programrendszer(ek) *telepítése*.
- A *futtatási környezet*: a futtatáshoz szükséges architektúra, szoftverkomponensek, könyvtárszerkezet, a program és adatfájlok neve és funkciója, esetlegesen a rendszer által használt/elvárt tipikus fájl nevek, kiterjesztések.
- A feladat egyes jellemző paramétereire vonatkozó feltételek, *korlátozások*.
- A futási idő, a szükséges operatív és háttértár *becslése* a feladat egyes jellemző paramétereinek függvényében (ha lehetséges).

#### 4.4. A felhasználói útmutató

Ebben szerepeljenek az alábbi (egymástól formailag is elkülönített) részek:

- *Általános kezelési útmutató*. Itt adjuk meg a program használata folyamán, több helyen is előforduló, általános jelleggel érvényesülő kezelési szempontokat, tennivalókat.
- A *szolgáltatások rendszere* összefoglalva (menürendszer, menüpontok).
- A több szolgáltatásban is használt, a tömörebb rövidebb leírás kedvéért *kiemelhető részfunkciók* kezelése (ha ilyenek vannak).
- *Szolgáltatásonként* (menüpontonként):
  - A megvalósított funkció (röviden).
  - A felhasználói teendők, speciális kezelési szempontok.
  - Jellegzetes képernyőkép (ha van).
  - Esetleges kivételes esetek, hibaüzenetek és ezek felhasználói lekezelése.

Ha a megoldás egy előírt vagy ajánlott alaprendszerre épül, akkor itt is elegendő csak a saját fejlesztéssel foglalkozni, illetve ha a leírás logikai összefüggései megkívánják az alaprendszer bizonyos szintű tárgyalását is, akkor ezen a szinten is világosan különüljön el a saját fejlesztés.

#### 4.4. Tesztelés leírása

Ebben a fejezetben kell bemutatni a program teljes életciklusát a forráskódoktól a valós környezetben való elindításig terjedően. Az olvasót végig kell vezetni, hogy az adott programot milyen hardver és szoftver környezetben, milyen módszerekkel lehet előállítani. (Fejlesztő környezet verziószámmal, szükséges modulok és komponensek, emulátorok, szerverprogramok, hálózati és egyéb konfigurációk, stb.). A felsorolásban részletesen kell kitérni az oktatásban nem szereplő összetevőkre!

Ezután következhetnek az előállított szoftvertermék futási tesztjei, próba adatai, a tesztadatok megválasztásának indoklása és a mérési eredmények összegezése.

Ebben a fejezetben kell bemutatni a fejlesztés során előálló **hibajelenségek** leírását és azok javítását is!

Ki kell térni a program hatékonyságának vizsgálatára (pl: memória- és háttértárigény, futási idő mérése) és fel kell tüntetni a futást befolyásoló tényezők szerepét (géptípus, felbontás, erőforrások).

#### 4.5. A melléklet

A melléklet tartalmazza a teljes forrásprogram-lista saját fejlesztésű részét. Ez is legyen áttekinthető, megfelelően tagolt (modulok) és az adott fejlesztőnyelvi szabályoknak megfelelően szabályosan és részletesen kommentált. Ha a megoldás egy olyan előírt vagy

## A dokumentáció tartalma

---

ajánlott alaprendszerre épül, amely forrásnyelvi formában is rendelkezésre áll, és az érthetőség és áttekinthetőség szempontjából célszerű ennek egyes részeit is listázni, akkor a listában is világosan különüljön el a saját fejlesztés. A más forrásból származó kódrészek esetén a forráskódban is jelölni kell a kód forrását.

A mellékletben adjuk meg a lemezmelléklet tartalmát is. Ez egy fájl-lista, a fájl tartalmának, funkciójának rövid, egy-két mondatos leírásával.

A lemezmelléklet tartalmazza:

- *kiemelve a fent említett fájl-listát,*
- *a dokumentációt, mint számítógépes dokumentumot,*
- *a forrásprogramokat és egyéb, a programnak a megadott fejlesztő környezetben való előállításához és installálásához szükséges szoftver anyagokat,*
- *a teszt és példaadatokat.*
- *a felhasznált elektronikus szakirodalmi hivatkozások mentéseit (e-jegyzet, weblapok, stb).*

A lemezmelléklet tartalma lehet (valamely általánosan használt tömörítő programmal) tömörített.

Ha feladatkiírásban a lemezmelléklet helyett elektronikus beküldés szerepel, akkor a fenti módon összeállított és összecsomagolt fájlokat nem fizikai adathordozón, hanem a kiírásban megadott módon kell beküldeni.

## 5. Munkaszervezés és ellenőrzés

### 5.1. Feladatkiadás és munkavégzés

A feladatkiadás általában az adott félév első felében (legkésőbb a 8. oktatási héten megtörténik). A feladatok kiírásáért a tantárgy előadója felelős.

A kiírás történhet irányítottan (minden hallgató megkapja a neki szóló feladatot) vagy választhatóan (egy adott készletből lehet választani). Az utóbbi esetben szabályozásra kerül az is, hogy ugyanazzal a feladattal maximum hány hallgató foglalkozhat és mennyi lehet az egy csoportban együtt dolgozó hallgatók száma.

A csoportmunkában megoldott feladat a csoporttagok közös produktuma. *A csoporton belüli munkamegosztástól függetlenül a csoport minden tagja egyénileg, a teljes produkcióért felelős, beszámoltatható és azonos értékelést kap.*

A feladatot pontosan abban a fejlesztőkörnyezetben (nyelv, verzió, kiépítettség) kell megoldani, amely az adott félében és tantárgyban az oktatásban szerepel.

### 5.2. Feladatbeadás és értékelés

A feladat elfogadható szintű teljesítése a félév lezárásának feltétele. Ha a félév vizsgával zárul, akkor a vizsgára bocsátás a feladat *előzetes* teljesítése.

A feladatot (dokumentáció és programlemez) a szorgalmi időszakban a gyakorlati foglalkozásokon adható be. A vizsgaidőszakra vonatkozóan külön időbeosztást adunk. A beadott feladatra értékelése legkorábban 3 nap, legkésőbb 1 hét alatt megtörténik. Előírható általános, vagy esetenkénti személyes beszámoló is. A konkrét ütemezést a mindenkori aktuális kiegészítő kiírás közli.

A feladat értékét pontozással állapítjuk meg. Ez a jelen általános szabályzat részét képező értékelőlap szerint történik. Ha a feladat valamely értékelési szempontból nem megfelelő, kijavításra visszaadjuk. Az újbóli beadásra is az eredeti ütemezés vonatkozik.

A feladat súlyát a féléves jegyben a mindenkori aktuális kiegészítő kiírás közli.

A féléves feladat elkészítése a féléves követelmény része, tehát az ehhez kötődő lehetőségek (pl. kerestféléves vizsgázás) csak azok számára biztosítottak, akik a féléves feladatot a kiadási félévben, legkésőbb a vizsgaidőszak végéig teljesítik.

## **6. Melléklet**

A melléklet tartalma:

- Címlap minta.
- Értékelő lap.

# SZÉCHENYI ISTVÁN EGYETEM

## JEDLIK ÁNYOS GÉPÉSZ-, INFORMATIKAI ÉS VILLAMOSMÉRNÖKI INTÉZET INFORMATIKA TANSZÉK



Tantárgynév (tantárgykód)  
féléves programozási feladat

**Feladat címe**

**Név (neptun kód)**  
BSc fokozatú Mérnök informatikus Szak, nappali tagozat

**2010/2011. II. félév**

## ÉRTÉKELŐLAP FÉLÉVES PROGRAMOZÁSI FELADATOKHOZ

Hallgató neve és NEPTUN kódja:.....

Értékelő neve:.....

Feladat címe: ..... Összpontszám: .....

**Kijelentem, hogy valamennyi nem hivatkozott programelem saját fejlesztésű, valamint a külső forrásból származó szoftverkomponensek forrásmegjelölése helytálló.**

..... (hallgató aláírása)

1. A dokumentáció formája .....pont

- 8..10: A dokumentáció gondosan szerkesztett, áttekinthető, stílus és nyelvhelyességi szempontból is jó.
- 5..7: A dokumentáció megfelelően szerkesztett, áttekinthető.
- 1..4: Kisebb hiányosságok vannak a jó esethez képest.
- 0: Lényeges hiányosságok vannak a jó esethez képest (pl. 'ömlesztett' szöveg, sok elírási, helyesírási hiba)

2. A dokumentáció tartalma .....pont

- 8..10: A dokumentáció kellő részletességgel tartalmazza az előírt részeket.
- 5..7: A dokumentáció minden fejezete kifejtett, de néhol hiányos.
- 1..4: Egyes fejezetek elnagyoltak, hiányosak.
- 0: Lényeges hiányosságok vannak a jó esethez képest (pl. erősen keverednek vagy teljesen hiányoznak egyes részek)

3. A program felépítése .....pont

- 8..10: A program jól illeszkedik a tananyaghoz és az előírt alaprendszerhez (ha ilyen van), a megkívánt eszközökkel és módszerekkel lett előállítva, objektumorientált.
- 5..7: A program szerkezete megfelelő.
- 1..4: Kevés önálló elemet tartalmaz.
- 0: Lényeges hiányosságok vannak a jó esethez képest (pl. nem ellenőrzött input, a tanultnál alacsonyabb színvonalú menükezelés, 'ömlesztett' programlista, paraméterezetlen szubrutinok, 'széteső' objektumrendszer )

4. A program működése .....pont

- 8..10: A program biztonságosan és helyesen működik, nincs hibajelenség.
- 5..7: A program csak részben valósítja meg a leírt funkciókat.
- 1..4: A program 'normál' tesztadatokkal biztonságosan és helyesen működik, de a 'szélső' esetek kezelése bizonytalan.
- 0: A program a megadott forrásanyagokból az előírt környezetben nem állítható elő (pl. szintaktikus hiba), vagy 'normál' tesztadatokkal is helytelen eredményeket ill. futási hibákat produkál.

Megjegyzés: *a tesztelésnél a hibaellenőrzési opciókat be kell kapcsolni.*

5. Védés (0..10 p.) .....pont

**Értékelés:**

- **Ha valamely értékelési szempont szerint a munka 0 értékű, akkor nem elfogadható.**
- **Az egyes szempontok azonos súllyal számítanak, tehát a maximális összpontszám 50.**
- **Ponthatárok: 0-26: 1, 27-32: 2, 33-38: 3, 39-44: 4, 45-50: 5**