

Erdélyi Magyar Műszaki Tudományos Társaság – EMT

Számítástechnika oktatás – cél vagy eszköz?

SZÁMOKT '98

Informatika az oktatásban

Illyefalva, 1998. május 22 – 24.

ITA/369

WOL Web Printed T CAL
WBT WBT CBT

Molens → 77 taml.

GARRY viny.

Jahel D Taml gounat Luni.
Duzon Uqur

Nembi blagstater. NAT
Seimbudo

ITA/369

Erdélyi Magyar Műszaki Tudományos Társaság

Számítástechnika az oktatásban

SZÁMOKT '98

Számítástechnika oktatás - cél vagy eszköz?

Illyefalva, 1998. május 22 – 24.

Szervező:

az Erdélyi Magyar Műszaki Tudományos Társaság
Számítástechnika Szakosztálya

A konferencia védnöke:

Dr. Sima Dezső, a Neumann János Számítógéptudományi Társaság elnöke

Tudományos bizottság:

Albert László – a konferencia elnöke

Égly János

Dr. Kása Zoltán

Dr. Pusztai Kálmán

Székely Tibor

Szervező bizottság:

Szalma Gyórfi Noémi – szervező bizottság elnöke

Bakó Judit

Balogh Csaba

Barabás Péter

Horváth Erika

Pap Éva

Tibád Zoltán

Támogatók:

Illyés Közalapítvány – Budapest

Iskola Alapítvány – Kolozsvár

Pro Technica Alapítvány – Kolozsvár

X-Net Computers – Kolozsvár

Program

péntek, május 22.

16⁰⁰ – érkezés, elszállásolás

19⁰⁰ – vacsora

szombat, május 23.

8⁰⁰ – reggeli

9⁰⁰ – előadások

13³⁰ – ebéd

15³⁰ – műhelyek

19⁰⁰ – vacsora

vasárnap, május 24.

8⁰⁰ – reggeli

9⁰⁰ – kerekasztal megbeszélés

13⁰⁰ – ebéd, hazautazás

Információs társadalom, oktatás

Kiindulópont: információs társadalom, oktatás

Kérdések: mennyire vagyunk készen? hogyan készülünk? mi az információs társadalom? ...

Az információs társadalom körüli viták, beszélgetések leggyakrabban használt kifejezései: számítógép, Internet, oktatás, információs és kommunikációs technológia, kommunikáció, multimédia, stb.

Jelenleg a számítógép olyan eszközzé vált mint valamikor a televízió. Emlékszem még arra a periódusra mikor a szomszédba jártunk televíziót nézni. Ma még a „szomszédba” járunk számítógépezni. Meddig? A számítógép nagy előnye, hogy olyan általános eszköz ami nagyon egyszerűen alakítható új és új „gépezetté”. Lehet tanár, játékmester, televízió, video, CD lejátszó, de számológép, írógép és sok más, vagy mindez egyszerre.

A számítástechnikai szakértők sokszor tekintik célnak a számítógépet, elvont fogalmakat emlegetnek, elámulnak a tudásukkal. Nem sok ember fogja megérteni, hogy ha két „compu-hős” SMTP és MIME labdákkal dobálózik akkor egyszerűen az elektronikus postáról van szó, illetve egy olyan elektronikus postáról ahol az üzenetbe újabb leveleket, dokumentumokat ágyazhatunk. Egy ilyen történet után érdemes végiggondolni, hogy bár a pénzzel mindannyian tudunk bánni, nem szeretnénk mindannyian tőzsdeszakértők lenni. Van aki felismerte és használja azt a lehetőséget, hogy bankszámláról is lehet „fizetni” a fizikai pénz használata nélkül. Van aki arról álmodozik, hogy hamarosan az „eletronikus” pénz fogalmát is megtanulhatjuk és használhatjuk.

Egy ennyire áltánosan használható eszközt (számítógép) érdemes/kötelező az új generációnak megismerni és felsőfokon **KI** használni. Továbbra is vita tárgya a *mit* és *hogyan* tanítani. A kezdők kategóriája (kb 4–80 év között) nem csak számbelileg a legnagyobb. Ők azok, akik tényleg eszközként fogják használni a számítástechnikát (nem írtam számítógépet mivel a számítástechnika sokkal tágabb eszköztárat fed: például az Internet, egy grafikus szerkesztő, egy tervező rendszer, stb)

Állandóan szem előtt kell tartani, hogy az is aki nem készül programozónak (még ha a tanárja csak programozást szeretne is vagy tud tanítani) – hanem orvosnak, eladónak, újságírónak, hivatalnoknak, mérnöknek – komoly mértékben tudja növelni eredményességét ha ilyen eszközöket használ. Ne feledjük el, hogy néhány év múlva azt fogjuk visszakapni amit ma mint tanárok vagy szakemberek nyújtani tudunk, főleg azoknak, akik nem számítástechnikai szakértőknek készülnek. Ennek alapján fogunk több vagy kevesebb időt eltölteni a postahivatalban, a villanyszámla kifizetésénél, ennek alapján házi-orvosunknak állandóan friss információ áll rendelkezésére a legújabb gyógyszerekről, ennek alapján a fiatal generáció nem félve és kétbalkezesen, hanem természetesen és golyóstoll-egyszerűséggel fogja kezelni ezt a nagyszerű eszközt, a számítógépet.

Albert László
a Számítástechnika Szakosztály elnöke

Előadások sorrendje

Dr. Racskó Péter (MKM, Budapest): *Sulinet – Középiskolai Internet Program Magyarországon*

Dr. Farkas Károly (Budapest): *Logo a programozócentrikus eszköz*

Zsakó László (ELTE): *Az informatika oktatásának alapkérdései*

Horváth László (ELTE): *Az oktatást segítő számítógépes eszközök és azok felhasználása a tanítási folyamatban*

Kovács Győző (Budapest): *Teleházak Magyarországon*

Műhelyek

Bács Andrea (ELTE): *Multimédia ToolBook – egy felhasználó és gyerekbarát nézőpont*

Csörnyei Zoltán (ELTE): *Funkcionális programnyelvek oktatása*

Bódi Antal (BGYTF, Nyíregyháza): *Internet hálózat alkalmazások tapasztalatai*

Dr. Oláh Gál Róbert (Csíkszereda): *Számítógépi grafikák*

Kovács Lehel (Kolozsvár): *Vizuális fejlesztő környezetek oktatása*

Oláh Anna (Budapest): *A LOSOWR programnyelv oktatási módszerei*

Vasárnapi kerekasztal

Egyed Albert (MKM, Budapest): *Sulinet a gyakorlatban*

Szabó T. Attila (Szombathely): *Kudarok, remények, tények, tervek a Világhálós kutatásban, közlésben, távtanulásban*

Zsakó László (Budapest): *A Nemes Tihamér Számítástechnikai Verseny kiértékelése, eredmények ismertetése*

Előadások

Sulinet

Középiszkolai Internet Program Magyarországon

dr. Racskó Péter

Magyar Közoktatási Minisztérium, Budapest

A Sulinet program 1996. szeptember 1-jén kezdődött, mikor Magyar Bálint művelődési és közoktatási miniszter a központi tanévnyitón bejelentette, hogy 1998. szeptember 1-ig valamennyi középiskola, 2002-ig pedig valamennyi általános iskola is, rendelkezni fog Internet-kapcsolattal. Globális célként a Művelődési és Közoktatási Minisztérium hatáskörébe tartozó közoktatási, valamint kulturális intézmények és közgyűjtemények internet-használati lehetőségének megteremtését, módszertani támogatását, a határon túli magyaroknak a hazai kultúrához az interneten keresztüli kapcsolatának fejlesztését és egységes, minden érintett számára elérhető közoktatási adatbázisok létrehozását határozták el.

A következő év költségvetésének tervezésekor a Parlament elfogadta az elképzelést, így 1997-ben már jelentős összeg – mintegy három milliárd forint – állt rendelkezésre erre a célra. Közben kidolgoztuk azt az elképzelést, aminek alapján a program megvalósítása 1997 februárban megkezdődött.

A programnak öt alapvető része van, amelyek egymásra épülnek, feltételezik azt, hogy a középiskolákban valamennyi diáknak módja lesz megismerkedni a hálózattal, használni az Internetet.

A fizikai infrastruktúra alapjainak megteremtése: ISDN vagy digitális bérelt vonali internet szolgáltatás

A program első része infrastrukturális beruházás, mely az országos internet szolgáltatás kialakítását tűzte ki célul mintegy 1200 intézményben. Jellemzők: 64 kbps digitális kapcsolat, teljes körű korlátozásmentes internet szolgáltatás és központi minőségbiztosítás. Ez a technikai háttér azt jelenti, hogy a vonal 40-50 gépet is képes egyszerre kiszolgálni és akár hang vagy kép is továbbítható rajta. A vonalhasználat jövőbeli költségeit is mindig a költségvetés fogja vállalni, az iskolák vagy fenntartóik helyett. Ez azért fontos, mert a vonalhasználat iskolánként kb. havi 100-120 ezer forintba kerül (a 964 középiskolára tehát havi 110 millió), amit az iskolák hosszú távon nem vállalnának, korlátoznák a felhasználást. Így azonban minden iskola és diák

tudhatja, hogy anyagi akadályai nincsenek az Internet használatának. A hálózat kiépítését és a folyamatos szolgáltatást egy több cégből álló konzorcium (Elender, Matávcom, Answer) nyerte 1997 augusztusának közepén. A vonalak kiépítése szeptembertől folyamatosan történik. Ilyen méretű informatikai fejlesztés még nem volt Magyarországon. Az 1998. májusi állapot szerint kb. 800 intézményt kötöttünk be az internetbe.

Internetes laboratóriumok kiépítése és a fizikai infrastruktúra megvalósulásának következő lépései

A program második része valamennyi középiskola részére biztosít egy olyan helyi hálózatba kapcsolt kabinetet, ami legalább hat, nagy létszámú iskola esetén akár 16 gépből áll, és közvetlenül kapcsolódik az Internetre. A gépek vásárlása a költségvetés terhére közbeszerzési eljárás keretein belül történik, és amennyiben az iskola még hozzá tud tenni az összeghez, akkor további munkahelyeket vásárolhat kedvezményes áron. (Ez összességében 14 ezer darab korszerű számítógép vásárlását jelenti, ami a teljes hazai piac több, mint 10 százaléka.) E pályázatok beadásának határideje 1997 szeptembere volt, a döntést követően kerül sor a szerződések megkötésére. Az iskolafenntartók a győztes cégekkel szerződtek a szállításra. A gépek leszállítása októberben kezdődött és május végére befejeződik. Ezzel összesen 1200 intézmény kap Internet laboratóriumot.

A tartalomszolgáltatás megtervezése, megszervezése és finanszírozása

Bár az Interneten ma is számtalan dolog elérhető, közvetlenül az oktatáshoz, diákélethez kapcsolódó anyagok csak elvétve vannak, azok is inkább idegen nyelven. Ha van hálózat és vannak gépek is, szükség lenne a tartalomra. Ez a program legnehezebb, hosszabb távon működő része. A Sulinet program már 1997-ben megkezdte korszerű, multimédiás, a hálózaton keresztül elérhető tankönyvek, internetes oktatási szemléltető anyagok, tantervek előkészítését, gyártását, ennek hatása azonban csak ez év közepére tapasztalható. A hálózat lehetőséget nyújt viszont olyan diáktevékenységekre, amelyekhez nem kell ennyi idő: a levelezőlistáktól az egyes iskolák honlapjának szerkesztéséig, a Sulinet napi magazintól a tanárok számára szóló hírlevelekig, már most is több minden elérhető, megtalálható. (www.sulinet.hu) A pályázati kiírás első körében olyan multimédiás anyagokat várt a Minisztérium, melyek interdiszciplináris jelleggel, az utóbbi évek kutatási eredményeit felhasználva tárnak fel egy-egy nagyobb témakört, és feldolgozhatóak egy tanóra alatt. A középiskolai internet programban szereplő oktatási anyagok hozzáférésüket tekintve két nagy csoportra oszthatóak: egy részük olvasható lesz az interneten, míg az anyagok másik része zártkörűnek

minősül és csak a programban résztvevő középiskolák, oktatási intézmények (pl. pedagógiai intézetek) és általános iskolák számára válik elérhetővé. Az anyagok ezen körét elsősorban a szerzői jogi problémák és kritériumok határozzák meg. Egy központi szerver lehetőséget ad arra, hogy onnét – az éjszakai kisebb forgalmú időszakot kihasználva – az egyébként on-line nehezen letölthető anyagok szétküldésre kerüljenek, majd a felhasználók a saját winchesterükről szakmai érdekeltségüknek megfelelően tölthessék le a felhasználni kívánt anyagokat.

A tanárok továbbképzése: a humán infrastruktúra megalapozása

Az internet oktatásban való intenzív használata elképzelhetetlen anélkül, hogy nemcsak az informatikát tanítók, de a különböző szaktárgyak tanárai is legalább felhasználói szinten ismerjék a rendszert és tudják, hogy az számukra miféle lehetőségeket tartogat. Az általános tanár-továbbképzési program keretein belül évi mintegy 3.800 millió forint áll rendelkezésre a tanárok továbbképzésére (ezen összegből hozzávetőlegesen 1 milliárd forintot igényelnek informatikai képzésre). A tanárképzés beindítása az informatika területén a Középiskolai Internet Program humán infrastruktúrájának megalapozását jelenti. Az MKM 50 képző-referenciaközpontot állított fel az ország különböző pontjain. Ezeknek az intézményeknek kötelezettséget vállaltak arra, hogy az elkövetkezendő három évben 500-1500 közti létszám-ban oktatják a tanárokat e tanfolyamokon. A képzés három szinten valósul meg, melynek tematikái a következők: alapfokú internet használat nem informatika szakos tanárok számára, felsőfokú szakképzettséget biztosító oktatási-informatikus képzés nem informatika szakos tanárok számára, iskolai rendszergazda képzés informatika szakos tanárok számára.

Kétirányú adatforgalom: a közoktatási adatbázis

Ha van a hálózat, lesznek gépek, akkor ezt a kitűnő eszközt már nem csak egyirányú forgalomra lehet használni, hanem az adatok az iskolákból kifelé is haladhatnak. Ez történhet egyszerű szabványos Internet-módszerrel (valamennyi iskolának lesz szervere), illetve szabályozottabb módon, adatbázisokon keresztül. Ha az iskola a teljes nyilvántartását egy előre megtervezett, országosan azonos formátumban tárolja, akkor a továbbiakban nem lesz szükség a különböző statisztikák, folyamatelemzések részére állandóan hibásan kitöltött kérdőíveket irogatni. Az egyes iskolák adatbázisa – a megfelelő jogosultságokkal – kívülről is elérhető, naprakész, pontos összesítések készíthetők városi, megyei vagy országos szinten. A közoktatási információs rendszer jelenleg készül.

Egyes határon túli magyar iskolák bekötése az internetbe

30 intézmény került kiválasztásra (16 Romániában, 4 Jugoszláviában, 8 Szlovákiában, 2 Ukrainában) Jelenleg folynak a tárgyalások az Internet szolgáltatásról és iskolánként egy kisebb számítógép laboratórium (1 szerver + 4 kliens gép) beszerzéséről.

Logo a programozócentrikus eszköz

dr. Farkas Károly

Arany János Gimnázium, Budapest

Különbféle témákban a tudásnak az alábbi fázisait ismerem:

1. Az adott témáról nem tudok semmit.
2. Az adott témáról megtudtam mindent.
3. Kétkedek egyes részletek igazságában.
4. Minden részletet cáfolni tudok.
5. Egyes állításokkal egyetértek.

Lehet, hogy vannak további fázisok is, fél évszázadnyi élet eddig vitt. Egy kérdésre válaszolni e fázisokban:

1. Nem tudok.
2. A tudatlanok biztonságával nyilatkozok.
3. Nem merek.
4. Nem illik.
5. Szívesen teszem.

A tudományos gondolkodásmód, a matematika, a számítástechnika tanulmányozása, művelése, fejlesztése az ötödik fázisban a SZÁMOKT '98 konferencia témájában a következő állásfoglaláshoz juttatott:

Számítástechnika-oktatás – cél vagy eszköz?

Erre tudok binárisan válaszolni: eszköz. A közoktatásban – ebben elég nagymérvű konszenzust érzek – nem cél a programozó képzés, a számítástechnika oktatása (Miért nem beszélünk Illyefalván is informatikaoktatásról?) eszköz a személyiségfejlesztéshez, az általános műveltség kialakításához, a jövő emberének formálásához. Milyen legyen a jövő embere? Az átlag ember nem matematikus, nem informatikus, de informatikailag művelt, informatikai szemlélettel rendelkező. Az informatika terén is felhasználó. De kreatívan, alkotóan felhasználó. Felhasználó vagy programozócentrikus oktatás? Nem ellentmondás, ha az első kérdésre adott válaszom után kijelentem, hogy a felhasználó-centrikus oktatással szemben előnyben

részeseitem a programozó, sőt a programozás-centrikusság bizonyos mértékű favorizálást.

A buddhizmusnak is talán egyik jellemzője a megfelelő arányok keresése. Véleményem szerint az informatikaoktatás sem lehet csak felhasználó-centrikus vagy csak programozás-centrikus. A kettő megfelelő arányát kell megtalálnunk. A megfelelő arányt mutató osztópont valószínűleg közelebb van a csak felhasználó véglethez. Viszont a programozás elemeivel való ismerkedés, a programozás elemeivel való játék, a Játékos Informatika Oktatás (JIO) elsősorban nem is a számítástechnika, nem is az informatika szempontjából hasznos, hanem a gondolkodásmód fejlesztésének egyik – talán napjainkban a legjobb – eszköze. Ha valaki a programozás elemeivel játszik, olyan gondolkodási sémákat ismerhet meg, amelyekről úgy véljük, hogy a jövő embere számára hasznosak. A Logo programnyelv tanítása és még kevésbé a programnyelv elemeivel való játék, nem kemény programozócentrikus oktatás, hanem felhasználói készségeket és alapvető programozói, azaz technikai, tudományos gondolkodási sémákat kialakító képzés.

A Logo az egyik kiváló eszköz a számítógépekkel és a programozható berendezésekkel például a robotokkal való kommunikálás elemeinek gyakorlására, az algoritmikus gondolkodásmód, a rendszerszemlélet, a technikai gondolkodásmód fejlesztésére. Például a teknőc-geometria elemi téziseinek egyike a teljes teknőc tétel: teljes (szép, kerek, befejezett) egy algoritmus, ha annak végén a teknőc visszaérkezik kiindulási helyére és állapota azonos lesz a kezdeti állapotával. Ha ezt tanítjuk, célszerűnek tartott programozói szokást fejlesztünk, de joggal mondjuk azt is, hogy csak egyszerűen rendre, rendszeretetre nevelünk. Az eljárások rövidegére való törekvéssel, strukturált gondolkodásmódot alakítunk, de egyben az egyszerűség keresésére is nevelünk. (Petőfi mondta: „Akiben nincs egyszerűség, abban semmi nincs.”) A Logo nemcsak alkalmazható a matematikai gondolkodásmód kialakítására, azt újszerűen, hatékonyan fejleszti. Papert – a Logo nyelv atyja – gyakran használja a matekfóbia kifejezést. Programnyelve erősen kerül a formális matematikát. Teszi ezt a matematika érdekében, teszi azért, hogy gyógyítsa, hogy megelőzze a matekfóbia kialakulását.

A teknőc-geometriát mindenkor szívesen használom geometriai-matematikai fogalmak megismertetésére, a matematika újszerű, szemléltető megközelítésére. Dinamikus geometriát tanítok. Egyik követőm Német Zoltán tanító (Győr) így fogalmazott: „Amit nem tudok megtanítani a hagyományos matematikával ötödikes diákjaimnak, azt a Logo segítségével harmadikos tanítványaim is megértik.”

A következőkben bemutatok egy példát a „dinamikus geometriára” Szinte mindegyik Logo könyv a teljes teknőc tételt a sokszögek rajzolásával mutatja

be. Általános eljárásra példaként ismerteti, vagy jobb esetben elvezet az "a" oldalhosszúságú "n" szögű poligon rajzolására alkalmas eljáráshoz:

```
Ismételd :n[megy :a jobbra 360 / :n]
```

(Itt és a továbbiakban a példánál a Magyarországon legelterjedtebb Logo nyelvjárás a LogoWriter általunk magyarított változatát, illetve az ennek megfelelő Comenius Logo 2.09 magyar változatát használom.) Az :n értékének változtatásával különféle szabályos sokszögeket kapunk. Ezek a sokszögek konvex sokszögek. Terjesszük ki az eljárást konkáv sokszögekre is! A teljes teknőc-tétel alapján, a sokszög oldalain végig haladva a teknőc összességében 360 fokot fordul. Nem a sokszög belső szögeivel számolunk, hanem a haladási irányhoz képest történő elfordulásokkal! A teknőc relatív koordináta-rendszerben „gondolkodik”. A balra fordulást tekinthetjük negatív irányú jobbra fordulásnak. Használjuk csak a jobbra forgásirányt. Szabályos konkáv sokszögnél egyforma fordulás-párok követik egymást, összesen $n/2$ esetben. A fordulás-párok tehát 360-nak az $n/2$ -ed részével egyenlők. A szabályos konkáv sokszög végigjárására-rajzolására alkalmas algoritmus így az alábbi módon is felírható:

```
Ismétel :n / 2 [megy :a jobbra -1 * :alfa megy :a  
jobbra 360 / (:n / 2) + :alfa]
```

Az :n és az :alfa paraméterek helyére számértékeket írva különféle konkáv sokszögeket kapunk. Az algoritmust eljárásba foglalhatjuk:

```
to konkávsokszög :a :n :alfa  
  ismételd :n[megy :a jobbra -1 * :alfa megy :a jobbra  
360 / (:n / 2) +  
:alfa]  
end
```

Ez után következhet a dinamikus szemléltetés, a sokszögcsaládok végigtekintése. Változtassuk folyamatosan az alfa szög értékét és figyeljük meg, hogyan változik sokszögünk. Mutassuk be, milyen családokat alkotnak a sokszögek, mi köze van egy csillagnak egy konvex poligonhoz vagy egy konkáv poligonhoz.

```
legyen "a 20 legyen "n 6 legyen "alfa 2 ismételd 1000  
[konkávsokszög :a  
:n :alfa vár 100 szivacs legyen "alfa :alfa + 2]
```

Másik példaként a rekurzió bemutatását példázó ábrasarozatra hivatkozok, ami a rekurziót jól érthető vizuális modellsorhoz köti. Ez a dinamikus geometriai képsor a magyar LogoWriter programlemez DEMO részletében található.

További példákat közlök publikációimban, amelyek közül itt a legfrissebbre hívom fel a figyelmet: Comenius Logo példatár. Második bővített kiadás.

A Logo programnyelv elemeit az általános iskolák alsó tagozatán is a gyermekek döntő többsége képes elsajátítani. Ennek birtokában önálló felfedezőútra indulhatnak nem csak a geometria szépségeinek terén. Olyan eszközt adhatunk a gyerekek kezébe, amellyel megszerettetni tudjuk a matematikai, a tudományos gondolkodást. Ez több mint a felhasználó-barát programok alkalmazása, a felhasználói szint kialakítása, amely fontos, nélkülözhetetlen, de az emberfők kiműveléséhez nem elegendő.

Az informatika oktatásának alapkérdései

Szlávi Péter, Zsakó László

ELTE TTK Informatikai Tanszékcsoport

1. Az informatika oktatásának célja, az informatika oktatás téveszméi és azok kritikai

A közismereti informatika tantárgy ismeretköre éles vitákat kelt az informatika tanárok körében is, de ennél sokkal élesebbet az informatikus szakemberek, valamint az informatikához laikus szülők esetében. A tanítandó területeket, azok fontosságát, arányait úgy próbáljuk megvilágítani, hogy megfogalmazzuk a hibás, szélsőséges nézeteket, majd kritikus szemmel megvitatjuk őket. Az alábbi téveszméket vesszük nagyító alá:

- Informatika tantárgy \equiv beszéljessünk az informatikáról általában
- Informatikaoktatás \equiv szakképzés
- Informatikaoktatás \equiv csak alkalmazói oktatás
- Alkalmazói oktatás \equiv alkalmazói program funkcióinak ismerete (az előbbi téveszme „al-téveszméje”)
- Informatikaoktatás \equiv csak programozásoktatás
- Programozásoktatás \equiv egy programozási nyelv oktatása
- Informatikaoktatás \equiv azonos súlyú ismeretkörök oktatása
- Informatikaoktatás \equiv az informatika önkényesen kiválasztott néhány részterülete
- Informatikaoktatás \equiv valamilyen informatikai eszköz teljes megtanulása
- Informatikaoktatás \equiv független ismeretkörök egymás utáni megtanítása
- Mindig a legújabb informatikai eszközt tanítsuk
- Az informatikatanár tudása kisebb, mint a tanulóé
- Az informatikaoktatáshoz nem kell Informatika tantárgy

Mindegyik téveszme tartalmaz természetesen valamilyen igazságelemet, de a végletes alkalmazásuk biztosan hibás.

Ugyanennek a vitának egy másik eredménye az informatikaoktatás céljának meghatározása, amely az alábbiakban foglalható össze:

– *korszerű alkalmazói készség kialakítása* (a számítógépek, az informatikai kultúra lehetőségeit kihasználni tudó tanulók képzése)

– *az algoritmikus gondolkodás fejlesztése* (a matematikához hasonló gondolkodásfejlesztő szerep, amely az iskolában, s a hétköznapi életben is alapvető fontosságú);

– *önálló munkára nevelés* (a számítógép, mint a tanuló tevékenységére azonnal reagáló eszköz, lehetőséget teremt az egyéni ütemű tanulásra, a tehetségekkel való különleges foglalkozásra, ...);

– *együttműködésre nevelés* (nagyobb számítógépes feladatok megoldása megköveteli a csoportmunkát, feladatok részekre osztását, a másokkal való kapcsolattartást);

– *termékelőállítási folyamat kialakítására, alkotó munkára nevelés* (akár programot írunk a számítógéppel, akár szöveges dokumentumot vagy adatbázist, a végeredmény akkor is egy termék lesz, a készítés folyamatának, s a „termékségnek” minden egyes következményével együtt);

– *az informatika és a társadalom kölcsönhatásának felismerése* (az informatika rohamos fejlődése az egész társadalmat gyökeresen átalakítja, s ebben az állandóan változó világban csak az érezheti otthon magát, aki érti a változásokat, s azok mozgatóit);

– valamilyen közismereti tárgyhoz kapcsolódó témakörben a tanulók önállóan készítsenek el számítógépes anyagokat (program, dolgozat, előadás, bemutatás, ...).

2. Az informatika ismeretkörei, témák ismeretkörökbe és korcsoportokba sorolása

Az informatikaoktatás céljaiból megállapítható, hogy az informatika ismeretek mely részét kell egy informatika tantárgyon belül megtanítani, s mely részét kell alkalmazni a többi tantárgyban, sőt az iskolai élet minden területén. Az informatika tantárgy tárgyalt ismeretkörei az alábbiak: [1]

– **Adatmodellezés, algoritmizálás, programozás** (az iskolai és a mindennapi életben lépten-nyomon algoritmusokat hajtunk végre, adatstruktúrákat – kérdőíveket, nyomtatványokat – töltünk ki, tevékenységsorozatokat, információáramlási folyamatokat tervezünk, s ezt a világot az érti igazán, aki tisztában van ezen tevékenységek alapjaival).

– **A programozás eszközei** (azon nyelvi és egyéb eszközök ismerete tartozik ide, amelyek az algoritmusok, adatmodellek megvalósításához, kipróbáláshoz feltétlenül szükségesek).

– **Alkalmazói feladatok megoldása számítógéppel** (itt a mindennapokban felmerülő problémák informatikai eszközökkel való megoldhatóságával kell foglalkozni: ábrszerkesztés, szövegszerkesztés, táblázatkezelés, adatbáziskezelés, ...).

– **Alkalmazói rendszerek kezelése** (az alkalmazási ismeretektől el kell választani a gyorsan elavuló eszközök kezelésének képességét, bár a tanításuk természetesen párhuzamosan kell, hogy történjen).

– **Gyakorlati problémák megoldása számítógéppel** (ebben az ismeretkörben a felmerülő problémából kell kiindulni – pl. osztálykirándulást kell szervezni –, szervezési feladatként kell vele először foglalkozni, utána ki kell választani az egyes részfeladatokhoz tartozó eszközöket – nem feltétlenül mindegyik informatikai eszköz –, ha szükséges akkor új eszközt kell készíteni, ...).

– **Informatikai eszközök működési elvei és alkalmazásuk** (sokféle hardver és szoftver eszköz áll rendelkezésünkre, amelyek célszerű alkalmazását minden számítógéphasználónak el kell sajátítani).

– **Informatika és társadalom** (érdemes megismerni az informatika, mint a kultúra egy részterülete múltját, foglalkozni az informatika várható fejlődésével, valamint jelenlegi hatásával a társadalomra, adatbiztonsággal, adatvédelemmel, az informatikaalkalmazás etikai kérdéseivel).

– **Az informatika matematikája** (az informatikai ismeretek elsajátításához szükséges matematikai alapok a matematika tantárgyban vagy nem szerepelnek, vagy nem ott, ahol szükség lenne rájuk – ez így jó, pl. a matematika nem foglalkozik mátrixokkal, a táblázatkezelési ismeretek azonban e fogalom hétköznapi megfogalmazását szükségessé teszik –, tehát erről, valamint a matematika alkalmazásairól az informatika tantárgynak kell beszélni).

3. Tananyagfelépítés, tantárgyfelépítés informatikából

Ebben a témakörben meg kell vizsgálni az alapvető tananyagfelépítési stratégiákat, s azok alkalmazhatóságát az informatikaoktatásban: [1]

– lineáris (ismeretkörök valamilyen egymásutániségát feltételezi, különböző tanévekhez különböző ismeretköröket rendel),

– ciklikus, ismétlődő (azt feltételezi, hogy az informatika főbb ismeretkörei valamilyen szinten már a tantárgy első tanítási évében meg kell, hogy jelenjenek, s ezeket a következő tanévekben fokozatosan tovább mélyíti).

A másik meggondolandó a tantárgy tanéven belüli felépítésének kérdése. Ebből a szempontból háromféle típusú ismeretkört vizsgálunk: [1]

– az ismeretkör anyaga nagy blokkokban, folytonosan oktatható (pl. nincs értelme szövegszerkesztést úgy tanítani, hogy minden harmadik órán foglalkozunk vele),

– az ismeretkör anyaga egy (esetleg két) órás blokkokban tanítható, hosszabban biztosan nem szabad foglalkozni vele (elrettentő példaként szoktuk hozni azt az iskolát, ahol fél évig folyamatosan DOS ismereteket tanítottak),

– az ismeretkör anyaga maximum 5–10 perces órarészekben tanítható (pl. számítástechnika történetéről egy 45 perces óra már biztosan unalmas, de egy 5 perces mese – egy más típusú, de nehéz tananyag elsajátítása közben – nagyon élvezetes lehet.

Irodalomjegyzék

[1] Szlávi P.– Zsakó L.: Az informatika ismeretkörei. (Lógia 30) ISZE ELTE TTK Általános Számítástudományi Tanszék, 1995.

Az oktatást segítő számítógépes eszközök és azok felhasználása a tanítási folyamatban

Horváth László

ELTE TTK Informatikai Tanszékcsoport
Informatika Szakmódszertani Csoport

Két oldalról szeretném a címben jelzett dolgot megvilágítani: az egyik oldalon áll a tanár, a másik oldalon ott a hallgató. Jellemzően most arról beszélnék, hogy egy számítástechnikát, informatikát oktató tanár milyen módon, milyen software eszközökkel, tudással tud abban segíteni, hogy a különböző tantárgyakban ésszerűen használhassuk a számítógépet. A nem szakos tanár szempontjából itt alapvetően három viselkedésforma képzelhető el:

1. Aktív hozzáállás, képes, és hajlandó olyan egyszerűen használható programok kezelésére, amelyek segítenek saját szaktárgyi anyaguk számítógépen való *összeállítására, az anyag bemutatására,*

2. Passzív hozzáállás, saját maga ugyan nem vesz a számítógépes anyag létrehozásában, de igényeit pontosan megfogalmazza, az elkészült bemutató anyagot a megfelelő instrukciók alapján *használja.*

3. Elutasító hozzáállás, még csak gondolkodni sem hajlandó azon, hogy milyen módon vonhatná be az oktatásba a számítógépet, mint eszközt.

A számítástechnikát, informatikát oktató tanár feladata tehát az is, hogy az első két kategóriába tartozó embereket, a közös cél érdekében – az oktatás színvonalának emelése –, segítse. A harmadik kategóriába tartozókban (itt zárójelben azért jegyezzük meg, hogy vannak olyan tantárgyak ahol esetleg csak nagyon erőltetetten lehet alkalmazni a számítógépet, mint oktatási

segédeszközt) pedig saját munkájával, ébresszen érdeklődést a számítógép használata iránt.

Röviden két olyan software eszközt, programot jellemeznék, amelyek segítségével oktatási segédanyagok színvonalasan előállíthatóak. Az egyik multimédiás anyag előállítására szolgál, a másik pedig főleg előadásszerű órák megtervezéséhez, elkészítéséhez, nyomtatott anyagának előállításához nyújt segítséget.

1. A TOOLBOOK, mint multimédiás demonstrációs anyagok előállítására szolgáló eszköz. Windows alatt futó alkalmazások, demok elkészítésére jól, kényelmesen használható eszköz. Kétféleképpen használható: ha megfelelő mennyiségű kép és hanganyag áll rendelkezésünkre, archívumokból, könyvtárakból, egyéb média helyekről, nincs más dolgunk, mint ezeket megfelelő módon egymáshoz csatolni, másrészt ha kellő szinten megismertük az előbbi lehetőségeket, és tovább akarunk lépni, akkor megtanulhatjuk a 'programnyelvet' is, melynek segítségével professzionális anyagokat készíthetünk. Felépítési egysége a könyv, ami lapokból áll. Feladatunk tehát nem más, mint ezeket a lapokat megszerkeszteni, könyvvé összeállítani. Ehhez alapvetően két dologra van szüksége a rendszernek: úgynevezett objektumokra (úgy is szokták jellemezni a TOOLBOOK-ot, mint objektumorientált környezet) illetve úgynevezett eseménykezelőkre. A használható objektum típusok a következők: egyrészt lehetnek a már meglévő nyersanyagaink, rajzobjektumok, hangobjektumok, képjobjektumok, de a vannak billentyű és mező objektumok, valamint speciális objektumok. Az elkészült anyag működésének alapja pedig az eseményvezéreltség. Ez a következőket jelenti: vár egy eseményre, ami lehet bármi, például egy billentyű lenyomása, az egér elmozdítása, stb. majd – természetesen ha definiáltunk hozzá műveletet – végrehajtja azt. Ezt úgy teszi, hogy egy esemény bekövetkezésekor az adott esemény üzenetet, vagy üzeneteket generál és elküldi azokat. Az így keletkezett üzeneteket ezek után az ún. eseménykezelő, amit esetleg éppen mi írtunk, képes elfogni, és ha van hozzárendelve végrehajtási utasítás elvégzi a műveletet. Mi történik akkor ha az adott eseményhez nem tartozik végrehajtási művelet? Ilyenkor az üzenet elindul 'felfelé'. Az objektumok ugyanis egy jól meghatározott struktúrába, fába rendezettek, melynek gyökere maga a TOOLBOOK rendszer, levelei pedig az objektumok, közben pedig vannak a csoportok, a lapok, és maga a könyv. A kapcsolat természetesen hierarchikus és egyirányú: nevezetesen az üzenet csak az ősök felé, a magasabb szinten lévők felé mozog. A rendszer menüvezérelten kényelmesen használható.

2. A POWERPOINT mint előadások készítésének hatékony eszköze. Ennek a programnak a segítségével lehetőségünk van arra, hogy mondandónkat,

strukturáltan tervezzük, megfelelő alakra hozzuk. Ezért a program készítői a szerkesztési műveletet a következő lépésekre osztották:

Indítási rész. Hogyan kell előkészülni egy ilyen dokumentum, alkalmazás készítéséhez, hogy az minél hatékonyabban, egyszerűbben menjen.

A bemutató típusa. Mire használjuk a készülő dokumentumot: értekezlet, tudományos előadás, terv – például tanterv – bemutatás, üzletpolitikai bemutató, közlemény, valamilyen szórólap, tanórai alkalmazás. Itt lényegében arra ösztökélnek bennünket, célratörően tervezzünk.

A bemutató eszköze. Tulajdonképpen, a mi szempontunkból az egyik legfontosabb kérdés. *Itt, ezen ponton kell azt eldönteni, hogy a számítógépet 'aktívan' akarom-e használni az óra során, vagy csupán arra használok, hogy elkészítsem vele az előadandó anyagot.* Látványos megoldás lehet, hogy az egymást követő lapokat számítógép segítségével jelenítem meg, például kivetítem, vagy ha van valamilyen számítógépes hálózat akkor minden hallgató láthatja a képernyőjén. Természetesen ez eszköz és pénz kérdése, de célként mindenképpen kitűzhető. Kevésbé pénzigényes, és még mindig elég hatásos a fólián való megjelenítés – lehet színes, fekete fehér –, diaképekkel egybekötött használat. Még látványosabbá érdekesebbé tehető a dolog megfelelő hangaláfestéssel színesítve.

Végül egy adminisztrációs rész következik, amelyben rögzíthetjük magunknak a bemutató jellemzőit. Hol kívánjuk elmondani, milyen helyszínen, készítsünk egy rövid tartalmi kivonatot, ha nem tanóráról van szó kell-e emlékeztető vagy nem kell.

Lehetőségünk van az előadás anyagába különböző objektumokat beilleszteni. Ilyenek a diaképek, lehetőségünk van – természetesen számítógépen történő bemutatás estén – mozgóképek beillesztésére. Lehetőségünk van hanganyagok beillesztésére is, valamint annak vezérlésére, csak időlegesen szóljon-e, vagy állandóan mintegy aláfestésként. Saját rajzoló része is van, így ennek segítségével is készíthetünk rajzokat. Nagy előnye az is, hogy kapcsolható az Internet-hez azaz lehetőségünk van internetes anyagok beépítésére a saját előadásunkba, illetve a program segítségével készített anyagok (például honlapok) az Internet rendszerbe való beillesztésére (ha erre megfelelő jogunk van).

Vizsgáljuk meg, hogy az így elkészült anyagokat milyen módon használhatjuk fel az órákon. Mindkét módon elkészített anyag alkalmas információátadásra, azaz tanításra. Ilyenkor azonban ügyelni kell arra, hogy a tanulót legalább kétféle inger éri: az egyik maga az előadó aki természetesen beszél, illetve a látvány, akár a számítógéppel, akár fóliával segített az oktatás. Ezért célszerű időt hagyni arra, hogy a mindkettőt kellően áttekinthesse. Célszerű időnként hatásművelő szüneteket tartani a beszédben míg a hozzá tartozó képi információt fel nem dolgozza, illetve célszerű időnként a fel-

használt eszközt kikapcsolni, és vagy összefoglalni, elemezni a látottakat, vagy csak egyszerűen folytatni az anyagot. A multimédiás anyagok használatakor egy másik lehetőségünk a hallgató aktív bevonása az új anyag elsajátításába. Ennek egyelőre a leghatékonyabbnak tűnő eszközei a multimédiás CD anyagok. Ezek általában egy teljes tantárgyi anyagot, vagy legalábbis annak nagy részét felölelik. Ezek használatakor, a tanár megfelelő irányításával és megfelelő magyarázataival, valamint a felmerülő kérdések megválaszolásával kellően érdekessé tehető az óra. Ha már magunk is képesek vagyunk megfogalmazni egy-egy kisebb témakör, vagy csak egyetlen óra ilyen eszközökkel történő megvalósításával szemben támasztott követelményeinket, akkor nekiállhatunk annak elkészítéséhez, vagy továbbadhatjuk igényeinket egy olyan embernek aki elkészíti azt nekünk.

Nézzük most a tanuló, diák, hallgató oldalát. Ő is többfeleképpen viselkedhet egy számítógépes környezetben:

1. Csak a számítógép érdekli és az nem, hogy ezen most éppen oktatás folyik. Ez még akkor is igaz ha a gép nincs bekapcsolva.

2. Csak a számítógép érdekli, de azért kíváncsi arra, hogy a tanár mit akar vele, mármint a géppel csinálni, de ha már bekapcsolták a gépet nem igen érdekli más.

3. Az érdeklődő, aki figyelemmel kíséri mindent, megcsinálja azt amit mondanak neki.

4. Az aktív, aki az adott tananyag feldolgozásával kapcsolatban, az órán használt demonstrációs eszközzel kapcsolatban visszajelez, ami a következő ilyen anyag kidolgozásához nyújthat segítséget.

5. Akit teljesen hidegen hagy, hogy milyen eszközzel folyik az oktatás.

A 3. és 4. csoportba tartozó diákok azok akikkel gyorsan együtt lehet dolgozni. Őket célszerű bevonni még az elkészítési fázisba is, hiszen eközben tanulhatnak a legtöbbet. A tapasztalat az, hogy a tanulók szívesen használják 'egyedül' a számítógépet (kellő előismeretek után). Ezért az Ő szemszögükből nézve olyan anyagok készítése is célszerű, melyeket gyakorlásként önállóan használhatnak. Például különböző nyelvi játékok, biológiából növény vagy állat meghatározó játékok, földrajzból felismerési játékok, stb. Ha ehhez a végén még egy gépes értékelést is kap, akkor bizonyosan törekedni fog arra, hogy legközelebb jobban sikerüljön neki.

A célunk tehát nemigen lehet más, mint ilyen eszközök használatának elsajátítása olyan szinten, hogy magunk is képesek legyünk demonstrációs anyagok készítésére.

Teleházak Magyarországon

Kovács Győző

a Magyar Teleház Szövetség elnöke – Budapest

A számítástechnika, ami századunk második felében megforgatta a világot – Magyarországon a korszak 40 évvel ezelőtt, az első jelfogós számítógép, a MESz-I és az első elektronikus számítógép, az M-3 megépítésével kezdődött – először a tudományos világ, később az üzleti élet eszköze volt, mind a mai napig nehezen vagy pedig egyáltalán nem jutott el a kistelepülések embereihez. Az persze nyilvánvaló, hogy a falvakban is tudtak arról, hogy van egy nagyon intelligens gép, amit számos helyen használnak, elsősorban az iskolában tanuló gyerekek meg talán a televízió révén még azt is megtudták, hogy milyen sokféle célra lehet a számítógépeket használni, de a falvak gyakorlatában, a napi munkában vagy a tanulásban a számítógépet szinte sehol sem alkalmazták. A számítógép és a számítástechnika – természetesen nem csak nálunk, de világszerte – elkerülte a legkisebb településeket.

Ez a helyzet 1985-ben egy váratlan ötlettől hirtelen megváltozott, megszületett ugyanis az első teleház. Az esemény messze északon Svédország egyik kis falujában – Vemdalenben – történt, az alapító Henrich Albrechtsen nyugdíjas olajmérnök egy olyan intézményt hozott létre, ahol a szavai szerint: „minden olyat meg lehet csinálni, amire egyébként a település lakói egyenként nem lennének képesek”. A mozgalom rohamosan terjedt a világban, ma már nem csak Európában vannak teleházak, de az amerikai és az ausztráliai kontinensen is. A teleház program – az UNESCO és a Nemzetközi Telekomunikációs Unió valamint a Teleházak Világszövetségének a támogatásával – most indul el Afrika néhány országában.

A hazai teleház történelem első lapjait Blaskó Jánosné a nagymágócsi községi könyvtárban írta, ahol az információs sarokban számítógéppel, fénymásolóval, egy közhasznú kézikönyvtárral és más hasznos eszközökkel létrehozta az első teleházi szolgáltatást. Sajnos – az alapító visszavonulása miatt – a nagymágócsi teleház 1995-ben már nem működött.

A magyar teleház történet második főszereplője Gáspár Mátyás, aki a fővárosi környezetből Csákberénybe „menekült”, ahol – mert a szükség így kívánta – hamarosan elkezdett a településnek és természetesen saját magának is egy olyan intézményt szervezni, akkor még nem tudta, hogy egy teleházat, ami azon kívül, hogy a falu lakóinak az ügyes-bajos dolgait intézi, informatikai csatornát is jelent az ország lakói és intézményei valamint a világ között. Már a csákberényi intézmény működött, amikor Gáspár Mátyás egy előadást hallott Horváth Jánostól (ma a Miniszterelnöki Hivatal helyettes államtitkára), hogy

Ausztriában az elszigetelt tiroli falvakban vannak hasonló intézmények, amiket teleházaknak hívnak.

Ezután nevezték el a csákerényi, ma is sikeresen működő, intézményt teleháznak és indult el a teleház mozgalom Magyarországon.

Gáspár Máttyás 1995-ben alapította meg a Magyar Teleház Szövetséget, aminek ma már több, mint 100 tagja van. A Szövetséget magánszemélyek alapították, 1997-ben történt meg a Szövetség újjászervezése, amikor az országban – elsősorban a „Hálózat a demokráciáért Program”-nak köszönhetően – már egymás után jöttek létre a teleházak. Megváltozott a tagság is, ettől kezdve a Szövetség rendes tagjai – az Alapítókön kívül – csak teleházak lehetnek, magánszemélyek pártoló tagságra pályázhatnak. Más szervezetek társult tagként léphetnek be a Szövetségbe. A Magyar Teleház Szövetség alapító elnökének ugyancsak Gáspár Máttyást választották, ma a Szövetség egyik tiszteletbeli elnöke, a másik Zám Mária a Hálózat a Demokráciáért program igazgatója, a Szövetség újjáalakulása óta az elnöki tisztséget én látom el.

A világon mindenütt a teleházakat – elsősorban civil szervezetek, de sokszor magánemberek is – külső támogatással hozzák létre, ugyanis a teleházak olyan szolgáltatásokat igyekeznek a kistelepülések lakóinak nyújtani, amihez az igény még ebben a környezetben nem jelentkezett. A teleházaknak – éppen ezért – az egyik legfontosabb feladata az informatika és a telekommunikáció alkalmazása iránti igénynek a felkeltése.

A teleházak üzleti vállalkozás formájában való kialakítása és üzemeltetése általában sikertelennek bizonyult. A külföldi és a hazai tapasztalatok szerint is a teleházak – az alapításuk után – 3-4 évig még az üzemeltetési költségeiket is képtelenek voltak megkeresni, ezután váltak önfenntartókká. Magyarországon, ahol főleg a kistelepülések önkormányzatai állandóan anyagi gondokkal küzdenek, egyáltalán nem volt lehetőség arra, hogy a teleházak felszerelését (számítógépek, fax, másológép, telefon, később Internet csatlakozás) önkormányzati pénzből megvásárolják, a fontossági listán a teleházakat megelőzte a szociális problémák megoldása, a járda, a gáz és a csatorna kiépítése, és hasonló más tervek, amikkel az élet elemi feltételeit, de legalább a minőségét biztosítani tudják.

Az elmúlt évben 23 teleház nyitotta meg a kapuit Magyarországon, amelyeknek a túlnyomó része az US-AID támogatásával, a már említett „Hálózat a Demokráciáért” Program pályázatán elnyert támogatásból kapta meg azt a kezdeti segítséget, amivel a tevékenységét el tudta indítani. A Program már kihirdette a már elbírált, következő pályázat eredményét, ami újabb 17 teleház megnyitásához nyújt anyagi, pontosabban eszköz támogatást. Néhány teleház létrejöttét – elsősorban azokat, amelyek széleskörű szociális programot indítottak – a Népjóléti Minisztérium is támogatja. A gazdaság-

talanul működő kis postahivatalok közül 10-15-öt a magyar Posta kísérletképpen teleházként is elkezd üzemeltetni, így hamarosan valamivel több, mint 60 teleház várja a látogatókat szerte az országban.

A teleházak többsége – elsősorban az elhelyezéshez, a fűtés, világítás stb kifizetéséhez valamint az üzemeltetők alkalmazásához – sok segítséget kapott a helyi önkormányzatoktól is.

A kormány „vidék modernizációs programja”-nak a keretében – a Miniszterelnöki Kabinetiroda, a Közlekedési, Hírközlési és Vízügyi valamint a Környezetvédelmi és Településfejlesztési Minisztériumok támogatásával – hamarosan elindul az Országos Teleház Program, ma már csak némi adminisztrációs probléma akadályozza a Program hivatalos bejelentését. Az országos program keretében már az első évben 50-100 teleház kaphat – pályázati formában – támogatást az induláshoz elsősorban informatikai eszközök (számítógép, Internet, másológép, fax, telefonvonal stb) beszerzésére.

A teleház maga is az informatika korának a tipikus „terméke”, az informatika eszközeinek a segítségével egy egészen új perspektívát nyit még az egészen kis települések fejlődésének is. A teleházat nevezhetjük a falu informatikai központjának, mások inkább közös irodának hívják, lehet falusi fórum is, ahol az emberek kicserélik a tapasztalataikat és a véleményüket. Miután a teleházakban ott van az Internet, a fórumot – ha a falu lakói úgy akarják – ki lehet terjeszteni az egész országra, de akár az egész világra is. A teleház a falu civil szervezeteinek a találkozóhelye, azok sem tévednek azonban, akik a teleházat a település távtanulási és távmunkavégzési központjának vagy csak egyszerűen a világháló egyik kicsi, de fontos hálózemének nevezik. A teleház hírt hoz a kicsiny településekre a világról és a falu a teleház segítségével mutatja meg saját magát a világnak. A teleház a falu kapcsolata a világhálón át a nagyvilággal, és a világról szóló információk forrása a faluban.

Számos példa mutatja, hogy a teleházas kistelepüléseken igen gyorsan fellendül a falusi turizmus, a honlapokon bemutatott helyi vállalkozók munkája vagy a mezőgazdasági vállalkozások terményei iránt a kereskedelem elkezd érdeklődni, de nem elképzelhetetlen – amikor Magyarországon is létrejön a távmunka – hogy a teleházak segítségével új munkahelyek is keletkeznek. Ma még illúzióknak tűnik, de hamarosan valóság lehet, a falu lakói úgy tudnak távoli, akár más kontinenseken lévő munkahelyeken is dolgozni, hogy ki sem mozdulnak a házukból, de a távtanulás segítségével akár diplomát is tudnak szerezni oly módon, hogy még a vizsgáikat is – az Internet segítségével – a teleházakban teszik le.

A teleházak révén a települések közvetlen kapcsolatba kerülhetnek egymással, a közintézményekkel, a kormányzati szervekkel, az ország és a

világ jelentős információs központjaival. Miután a kapcsolat kétirányú és cenzúrázatlan, ezért lehetőség van a közvetlen tapasztalatcserére bárkivel, akár az országot vezető politikusokkal is. Az a falusi polgár, aki korábban legfeljebb a falu ügyeibe szólhatott bele, a teleház-kapcsolattal közvetlenül fejtheti ki a véleményét a megye, de akár az ország ügyeit eldöntő fórumokon is. Létrejöhet egy országos tapasztalatcsere, sőt különféle témákban még együttműködés is kialakulhat akár az ország két távoli települése, de más országok hasonló közösségei között is.

A teleház lehet a végállomása a távolról történő ügyintézésnek, mód lehet arra, hogy a hivatalos ügyeket a falusi emberek ne személyesen – például a városi és a megyei önkormányzatoknál, a földhivatalokban, a társadalombiztosításnál, az adó- és más hasonló közintézményekben – intézzék, hanem a hálózatokon keresztül a falu teleházainak a közreműködésével oldják meg a problémáikat.

A teleház segíti a civil szervezeteket, telekommunikációs lehetőségeivel és irodatechnikai felszerelésével a helyi vállalkozókat, kiadója lehet a helyi újságnak és így – talán nem túlzás kijelenteni – motorjává válhat a település közösségi életének.

A Magyar Teleház Szövetség ebben az évben tartotta meg az I. Országos Teleház Szimpóziumot valamint a Szövetség újabb tisztújító közgyűlését is. A szimpóziumon Lotz Károly és Baja Ferenc miniszterek jelentették be az Országos Teleház Program indulását, Magyar Bálint miniszter pedig a teleház mozgalom és az MKM ugyancsak országos jelentőségű Sulinet iskolai programjának az együttműködését.

A Szövetség – eddig – kétszáznál több olyan települést tart nyilván, amelyek már jelezték, hogy komolyan foglalkoznak teleház alapításának a gondolatával és nagyon várják az Országos Teleház Program indulását. A teleház mozgalom fejlődésnek újabb állomása, hogy a már működő teleházak elkezdnek foglalkozni a környező településekkel és egymás után jönnek létre a teleházas intelligens régiók, amelyekben a régió első teleházának példája nyomán több faluban is teleházakat alapítanak.

Magyarországon – eddigi tapasztalataink szerint – nincs két egyforma teleház, ami azt is mutatja, hogy a teleházak teljesítik a kis településen lakók kívánságát és olyan feladatokat oldanak meg, amelyekre éppen az adott településen szükség van. A teleházak nem csak az ott lakóknak, de a helyi önkormányzatnak is segítenek elsősorban azzal, hogy átvállalják – az egyébként az önkormányzatra háruló – civil feladatokat. Nagyon sok teleház van, amelyik felvállalja a falu szociális programját, más teleházak vezetőkes televíziózás vagy Internet szolgáltatóvá lépnek elő, vannak teleházak, amelyek a saját stúdiójukból önálló helyi műsort is sugároznak. Találkozunk olyan kis településekkel, ahol a helybéli, főleg nyugdíjas szakemberek vállalják az

adó-, a könyvelési, az orvosi, a vállalkozói és ki tudja még milyen tanácsadást. Azokban a falvakban, ahol az iskolában még nincs számítógép, a teleházban folyik a számítástechnikai oktatás, aminek a vezetője általában megszervezi a felnőttek számítástechnikai képzését is. A teleház a helyszíne a helyi civilszervezetek megbeszéléseinek, foglalkoznak a falu jövőjével és azzal is, hogyan lehetne a faluban elegendő munkahelyet teremteni.

Ha nem született volna meg a nyolcvanas években – Svédországban – a teleház, akkor most ki kéne találni.

A teleházakkal folytatott konzultáció eredményeként a hazai teleházakban közel 50 különféle tevékenységet sikerült összeírni, ami azt mutatja, hogy szükség van a teleházakra, amelyek egyre több olyan feladatot látnak el, amire minden faluban igény lehet.

A teleházak nagyon fontos „missziója” az informatikai szolgáltatások iránti igény megteremtése azokban a falvakban, ahol addig az informatikai kultúrának még a nyoma sem voltak meg.

A teleházak tipikus non profit szervezetek, a Szövetség véleménye szerint az volna a célszerű, ha ilyeneknek is maradnának. A teleház program az egész társadalomnak szól, ma még csak kezdeti eredményei vannak, ezért az igazi jelentősége ma már látható, de még felmérhetetlen.

A Magyar Teleház Szövetség most alakítja ki együttműködési politikáját – a kormány és a minisztériumok után – elsősorban az informatikai és a más civil szervezetekkel. Az első kapcsolat a MTESz-szel és a MTESz közvetítésével a hazai tudományos egyesületekkel jön létre. Az utóbbiak a MTESz tagszervezetei. A Magyar Teleház Szövetség fontos feladata, hogy létrehozza a saját szövetségi politikáját és megteremtse a kapcsolatot a világ más részein szervezett teleház szövetségekkel. Miután ebben a régióban, a volt szocialista országokban csak Magyarországon vannak teleházak, fontos feladata a szövetségnek a teleház mozgalomnak a szomszéd országokra való kiterjesztése. Ezt a tevékenységet a Teleházak Világszövetsége és a Nemzetközi Távközlési Unió azzal támogatja, hogy – valószínűleg – még ebben az évben Magyarországon rendezzi meg a térség országai részére a teleházak tevékenységét bemutató nemzetközi szemináriumot.

Ha az Országos Teleház Program sikeres lesz és miért ne lenne az, akkor Magyarország, mint sok más intellektuális tevékenységben, a teleházakban is „európai nagyhatalommá” válik miközben közösen kialakítjuk azt a teleház típust, ami valóban záloga lehet, az informatikai szempontból teljesen elhanyagolt kistelepülések további fejlődésének.

Műhelyek

Multimédia ToolBook – egy felhasználó- és gyerekbarát nézőpont

Bács Andrea

ELTE TTK, II. éves doktorandusz hallgató

Általános jellemzők

Szerzői rendszer **multimédiás** anyagok szerkesztésére

Elsősorban **oktató és demóprogramok** írására alkalmas

Egyszerűen **látványos** alkalmazások készítése

Nem igényel programozási ismereteket

DE: **programozható** (OpenScript)

Jól kezelhető fejlesztői környezet

Kiegészítő programok az objektumok létrehozására (menü, ikon, dialógus-
ablakok, színpaletta-szerkesztők)

Támogatja a más Windows programokkal való **kommunikációt** (DDE,
OLE)

Exe állomány készíthető az alkalmazásból (a környezettől ez azonban nem
teljesen független)

Bővíthető (DLL-ekkel való bővítés kívülről, DLL-t nem tudunk készíteni
benne)

Multimédiás lehetőségek

Állókép

Animáció, Útvonal-animáció

Hang

Vágás (klip-, színpad)

Fejlesztői környezet

Jellegzetesen Windows-os

Széleskörű lehetőségek objektumok rajzolására

Szerzői és felhasználói szint (alkalmazás létrehozása illetve használata)

A ToolBook működése

Objektumorientált (minden, amit létrehozunk egy objektum)

Objektum= megjelenési forma + eseményekre való reagálás

Eseményvezérelt – általában a programozás a specifikus események lekezelését jelenti.

Az alkalmazások felépítése

Könyv metafora

Objektum-orientált, vagyis minden TB komponens egy objektum, amelyhez egy tulajdonsághalmaz tartozik (ez szerzői szinten megváltoztatható).

A tulajdonságok lehetnek **vizuális jellegűek** (keret, szín, formátum, stílus) és **azonosító jellegűek** (név, azonosító, típus, a laphoz való viszonya).

Oldal, lap (page) = képernyő, amelyek között váltogathatunk ún. linkek segítségével.

egy alkalmazás alapvető egysége
objektumok vannak rajta,
háttere van

Könyv (Book) = (képernyő) oldalak összessége

Cím, méret
Védelem (jelszó)
Script

Viewer = Windows ablak, amelyen keresztül az alkalmazást „nézzük”.

Egy lap felépítése

Háttér (background) – általában egységes a könyv nagyrészére

Maga a tartalmat hordozó lap (page)

Egy háttér – több lap

Több háttér nem tartozhat egy laphoz

A háttéren és a lapon is objektumokat helyezhetünk el:

Grafika (bmp, .dib formátum)

Gombok (radio, check, nyomógombok)

Szövegmezők (szerkeszthető és statikus) – hotwordok is

Lenyíló listák

Hyper-hivatkozások

OLE container

Az alkalmazás készítője és a rendszer által rajzolt objektumok

Minden objektumhoz tulajdonságokat rendelhetünk (név, szín, méret, stb...)

Viselkedés (az események lekezelése)

Fontosabb objektumok:

Könyv (program és objektum egyben)

Alap-**lapméret** megadása

Kulcsszók megadása (védelem)

Tartalmaznia kell legalább egy oldalt és egy ablakot (Viewer)

Háttér (background)

Több laphoz rendelhető hozzá

Megadható a lapméret, különben a könyv lapméretét használjuk

Ablak (Viewer)

Az ablak beállítása mindkét szintre érvényes

A Windows- ból megszokott tulajdonságokkal rendelkezik

Gombok (push, label, checkbox, radio button).

Scriptek - események kezelése

Minden objektumnak van tulajdonsága

Egy objektum több eseménykezelőt is tartalmazhat

Egy objektum másolásával a scriptet is illesztjük

AutoScript: témák szerint csoportosított, előre megírt scriptek halmaza

Recorder – felvétel, lejátszás, másolás a Script Editorba.

Objektum-hierarchia

A hierarchia egy alkalmazás szerkezetét követi.

Az üzenet a hierarchián "utazik" felfele az eseménykezelőt tartalmazó ősig.

Egy üzenet végleg elfogható (forward utasítás).

Az OpenScript nyelv

teljes programozási nyelv

beépített matematikai, stringkezelő,

grafikus függvényrendszert tartalmaz

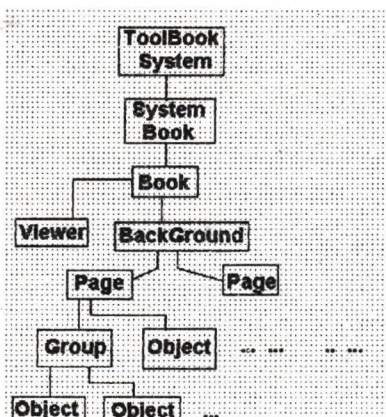
kis komponensekből, ún. **script**ekből építjük fel az alkalmazást az eseménykezelő script :

```
to handle buttonup
```

```
.....
```

```
end
```

felhasználói üzenetek (pl. a felhasználó tudásszintjének beállítása)



Miért az iskolában?

könnyen kezelhető

„Vizuális programozás” – a programozásban nagyon gyakran látható objektumokkal dolgozunk, de magasabb szinten kódolásra is szükség van látványos, motiváló

rugalmasság, átalakíthatóság – már megírt programokból gyorsan lehet újat összeilleszteni

bonyolult fogalmakat tesz kézzelfoghatóvá (objektumok, osztályok)

interdiszciplinaritás, sokoldalúság

humán műveltség terén is felhasználható

Hátrányok

drága szoftver, nincs oktatási verzió

hardver-igényes

a generált .exe programok futtatásához szükséges a futtatókörnyezet (ez azonban ingyenes)

Funkcionális programnyelvek oktatása (Funkcionális programnyelv az első programnyelv?)

Csörnyei Zoltán

Eötvös Loránd Tudományegyetem Természettudományi Kar, Általános
Számítástudományi Tanszék

Az informatika elmúlt évtizedeiben több programozási stílus alakult ki, az egyik lehetséges osztályozás a számítási modell szerinti csoportosítás. Az egyik stílus a Neumann-elvű számítógép tulajdonságain alapul, és a hardver optimális működtetésére koncentrál, a többi stílus alapja pedig a matematika, elsősorban a logika, a függvények elmélete, vagy akár az automaták elmélete.

A továbbiakban a két fő programozási stílussal foglalkozunk. Az *imperatív* (vagy *procedurális*) *programnyelvek*, mint például az assembly nyelvek, a Modula, C, Pascal vagy az Ada, változóban tárolt adatok manipulálásán keresztül adják meg az eredményt, és ezeken a nyelveken a manipulációt, azaz a megoldást végző algoritmust utasítások sorozatával lehet leírni.

A változó fogalma a hardver regiszterekből és a memóriarekeszekből alakult ki, a szekvenciális utasítássorozat elve a gépi kódú program végrehajtására vezethető vissza. A gépi kódú utasításokkal a változók értékeit változtatjuk meg, az imperatív nyelvű programok tipikus utasítása az

$x = \text{kifejezés}$ vagy $x := \text{kifejezés}$

formájú értékadó utasítás. A program utasításai a programozó által megadott sorrendben hajtódnak végre, és egy adott feladat megoldásakor az optimális program megírásához a hardver és szoftver rendszer működését, dinamikáját is ismerni kell. Vannak azonban olyan utasítások, amelyek problémát okozhatnak, azaz nem a kívánt működést eredményezik. Ilyenek az ugró utasítások, amelyek megbontják az utasítások leírt, statikus szekvenciáját, ennek a problémának a megoldására szolgál Dijkstra struktúrált programozási módszere. De problémát okozhatnak az egyszerű értékadó utasítások is, hiszen mindenki könnyen írhat olyan programot, amely különböző futtatásokra különböző eredményt ad, miközben a programot természetesen nem változtattuk meg.

A másik programozási stílushoz a *funkcionális (vagy applikatív) programnyelvek* tartoznak. A funkcionális program egy kifejezés, és a program végrehajtása a kifejezés értékének meghatározását jelenti.

A funkcionális programok nem használnak változókat, az, hogy egyes korábbi funkcionális programnyelvben definiálhatók változók és ezek értéket vehetnek fel, csupán "mellékhatások", az értékadás a program végrehajtásának egy speciális mellékhatása. A funkcionális programban nem adhatunk meg szekvenciális utasítássorozatot, itt csak magasabb-rendű függvények vannak, amelyek az argumentumaikon keresztül kapnak értéket, és a függvények értéke függvény is lehet. Nincs ugró utasítás, nincs ciklusutasítás, viszont van a matematikából jól ismert rekurzio.

Nem okoz problémát a végtelen fogalmának megvalósítása sem, végtelen adatszerkezetek, például végtelen listák kezelése a funkcionális programnyelvekben természetes módon valósítható meg. Korlátot csak a hardver, a rendelkezésre álló memória mérete jelent.

Egy funkcionális program mindig ugyanazt az eredményt adja, függetlenül attól, hogy hányszor, vagy milyen környezetben futtatjuk. Mivel nincs mellékhatás, a program helyessége könnyen bizonyítható matematikai módszerekkel.

Egy funkcionális program általában rövidebb, mint az ugyanazt a feladatot megvalósító imperatív program, de a lényeges különbség a két stílusban írt programot nézve azonnal szembetűnik: ez a funkcionális program olvashatósága. Ezt egy példán mutatjuk be. Adjuk meg a $fac(n) = n!$ programját C nyelven és két funkcionális programnyelven:

A C nyelvű program:

```
int fac(int n)
{ int x = 1;
  while ( n > 0 )
    { x = x * n;
```



```

        n = n - 1;
    }
    return x;
}

```

A Miranda nyelvű program:

```

fac n = 1          if n = 0;
n * fac( n - 1)  if n > 0;

```

A Clean nyelvű program:

```

fac 0 = 1
fac n = prod [ 1 .. n ]

```

A funkcionális nyelven írt programok olvashatósága nyilvánvaló, a program szinte azonos a függvény matematikából jól ismert definíciójával. Egy adott algoritmust az imperatív nyelvekben szekvenciális utasítássorozatra kell alakítani, és meg kell határozni a részeredmények tárolására szolgáló változókat. A funkcionális nyelveken a programot közvetlenül a matematikai objektumokból lehet felírni, a program szemantikájának leírása maga a program.

Meg kell említeni természetesen a funkcionális stílusú programozás hátrányait is. Egy funkcionális nyelven írt program általában lassabban fut, mint az imperatív nyelven írt megfelelője, ennek oka az, hogy a futtatandó kód nem az adott hardverre optimalizált kód. A modern módszerek, például a gráf-redukciós módszerek, ebben lényeges javulást hoztak. A funkcionális nyelvek másik hátrányának az hozható fel, hogy vannak olyan feladatok, amelyek megoldása imperatív nyelveken könnyebben leírható. Erre a választ az új funkcionális nyelvek adják meg, amelyeknek a tervezésénél már figyelembe vették ezt a problémát.

A funkcionális programozási stílus elméleti háttere a lambda-kalkulus, a függvények elmélete, amit az 1930-as évek elején Church dolgozott ki. Minden funkcionális program átalakítható a lambda-kalkulus egy kifejezésére. Church, Kleene és Turing kutatásainak eredménye, hogy már 1937-ben bebizonyították, a Turing-kiszámítható függvények osztálya azonos a lambda-kalkulusban definiálható függvények osztályával, azaz minden effektív kiszámítható függvény leírható funkcionális programnyelven.

A lambda-kalkulus – vagy még inkább a vele ekvivalens kombinatorikus logika, a kombinatorok elmélete – rendkívül egyszerű, és használhatóságát az adja, hogy a rekurzió benne a rekurzió explicit megadása nélkül leírható.

Összefoglalva, úgy tűnik, hogy azokra a matematikai alapismeretekre alapozva, amelyeket a tanulók alaposan elsajátítanak a képzés korábbi

éveiben, célszerűbb elsőként egy, a matematikához közel álló programozási stílust és programnyelvet tanítani. A funkcionális programok matematikai ismeretekre alapozott tiszta szintaktikája és szemantikája lehetővé teszi, hogy a tanulók könnyen elsajátítsák ezt a programozási stílust. Tréfásan azt lehet mondani, hogy egy funkcionális nyelv megtanulható néhány óra alatt, de ha valaki már ismer egy imperatív nyelvet, ez akár napokig is eltarthat. A funkcionális programozás elméleti háttérének megismerése sem okozhat problémát, az iskolában tanult matematikai függvényfogalomra építve a lambda-kalkulus függvényjelölése és függvénytranszformációi könnyen elsajátíthatók.

Tehát azt javaslom, hogy az első programnyelv, amivel a tanulók megismerkednek, egy funkcionális programnyelv legyen. Például az egyik legmodernebb, mellékhatás-mentes, lusta paraméterkiértékelésű nyelv, a Clean. A Clean-programok fejlesztéséhez szükséges környezet, a fordítóprogram, szabadon használható, másolható és terjeszthető. És ekkor talán majd az első programozás órákon a tanulókat nem éri az a meglepetés, hogy az $x = x + 1$ -ben nem lehet egyszerűsíteni x -szel...

Irodalom:

1. Simon L. Peyton Jones: The Implementation of Functional Programming Languages, Prentice Hall, 1987.
2. R. Plasmeijer, M. van Eekelen: Functional Programming and Parallel graph Rewriting, Addison Wesley Publishing Company, 1993.
3. Functional Programming Languages in Education, First International Symposium, Nijmegen, 1995., LNCS-1022.
4. Concurrent Clean, University of Nijmegen, 1997.

Megjegyzés: A Sepsiszentgyörgyön megrendezendő Bolyai Nyári Akadémia informatika szekciójában ebben az évben a főtéma a „nem-imperatív programnyelvek” lesz, amelynek keretében a Prolog logikai, a Logo és Clean funkcionális programnyelvről és ezek tanításáról lesznek előadások és géptermi gyakorlatok.

A Szab-I-Net Területi Számítógépes Hálózat Fejlesztő és Internet Szolgáltató Közhasznú Társaság tevékenysége

Bódi Antal
ügyvezető

A Szab-I-Net Program

A nyíregyházi Bessenyei György Tanárképző Főiskola Számítóközpontja által kidolgozott program – a Szab-I-Net Program – egy összehangolt informatikai fejlesztési cél elérését jelenti Nyíregyházán és Szabolcs-Szatmár-Bereg megyében.

A fejlesztéssel lehetővé válik a felsőoktatásban felhalmozott fejlett kommunikációs technológia széleskörű eljuttatása a város és a megye intézményeihez, vállalkozásaihoz és a magánszemélyekhez is. Az alapvető megvalósításhoz egy egységes regionális informatikai rendszer létrehozásán keresztül vezet az út. Ehhez megkeressük a lehető legoptimálisabb megoldást és a meglévő infrastruktúrákat igyekszünk maximálisan kihasználni. Erre a legjobb példa a kábeltelvíziós hálózat kísérleti adatátviteli célú használata.

Szab-I-Net Program Célja

Az általunk végzett fejlesztések eredményeként fogalmazódott meg bennünk az ötlet, hogy ezt a lehetőséget próbáljuk meg átadni a környezetünknek is. A Szabolcs-Szatmár-Bereg Megyei Közalapítványnál sikerült egy pályázatot elnyernünk, amelynek eredményeként az országban elsőnek sikerült kialakítanunk 10 Mbps sebességű Internet kapcsolatot a nyíregyházi Kábelkom Kft. kábeltévé hálózatán 1995 novemberében. Ez azt jelenti, hogy potenciálisan egy 25.000 csatlakozási ponttal rendelkező nagysebességű számítógépes hálózat alapjait raktuk le. Természetesen az országos akadémiai hálózati alapelveket semmilyen körülmények között nem akartuk megsérteni, így igyekeztünk szövetségeseket keresni ezen eredményeknek a kiaknázására. Sikerült megnyernünk ennek a célnak a megvalósításához a Nyíregyháza Megyei Jogú Város és a Szabolcs-Szatmár-Bereg Megyei Önkormányzat segítségét és támogatását. Megtálaltuk a kedvező szervezeti formát a Szab-I-Net Közhasznú Társaság megalapításában és amennyiben a terveink valóra válnak, elérjük célunkat, hogy az általunk kidolgozott fejlesztési programba minden érdeklődő kapcsolódhat és költséghelosztásos elven megfizethetővé válik az Internet mindannyiunk számára.

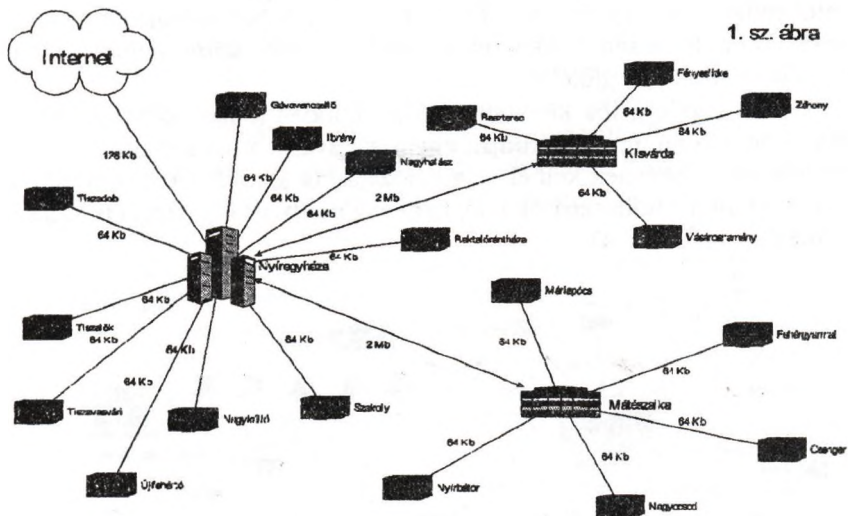
A Szab-I-Net hálózat nem akadémiai ügyfelei az akadémiai rendszertől függetlenül csatlakozhatnak az internethez, a HUNGARNET (akadémiai egyesület) tagok pedig az Nemzeti Információs Infrastruktúra Fejlesztési

Program Régióközpontján (NIIF) keresztül válhatnak a H-BONE (országos akadémiai hálózat) rendszer felhasználóivá. Az NIIF „Intelligens Város” kísérleti pilot program keretében 20 nyíregyházi HUNGARNET intézményt kapcsolunk be a hálózatba. A fejlesztési eszközöket az NIIF biztosította 7.5 millió forint értékben.

Szab-I-Net Program megvalósítása

Jelenleg a városi hálózaton 37 db router (kapcsológép) működik. Ez azt jelenti, hogy ennyi intézmény számítógépes hálózata kapcsolódik össze egy nagyobb hálózatba Nyíregyházán. A routereken keresztül az intézmények hálózata teljesen vagy részben szabadon kommunikálhat a világgal az Internet segítségével.

A megyében a szolgáltatások beindítását elsősorban dial-up (telefon vonalon keresztüli kapcsolódás) szolgáltatások biztosításával kezdtük el. Jelenlegi konstrukcióban helyi hívási díj ellenében biztosítjuk a hálózati hozzáférést a felhasználóknak. A tervezett hálózat az 1. sz. ábrán látható.



A MATÁV Rt. Debreceni Igazgatósága nagy szakmai hozzáértéssel segíti munkánkat és közösen igyekszünk a felmerülő problémákat a lehető legjobban megoldani. Nagyon komoly probléma a megyében, hogy a távközlési hálózat minősége nagyon heterogén. Nagyon fontos eredménynek tartjuk, hogy a vidék megmozdult. Az ügyfeleinknek több mint 40 %-a, már nem Nyíregyházáról veszi igénybe a szolgáltatásainkat. Azzal, hogy a városi számítógépes hálózaton a saját tartalmukkal már megjelentek a nagy megyei

közüntézmények (Megyei Önkormányzat, ÁNTSZ, Megyei Kórház stb) nagyon élénkítőleg hatott a megyei hálózat fejlődésére.

A nagyobb felhasználók igényének kielégítésére bérelt vonalak és ISDN kapcsolatok kialakítását tervezük létesíteni a pont-pont közötti összeköttetésekre.

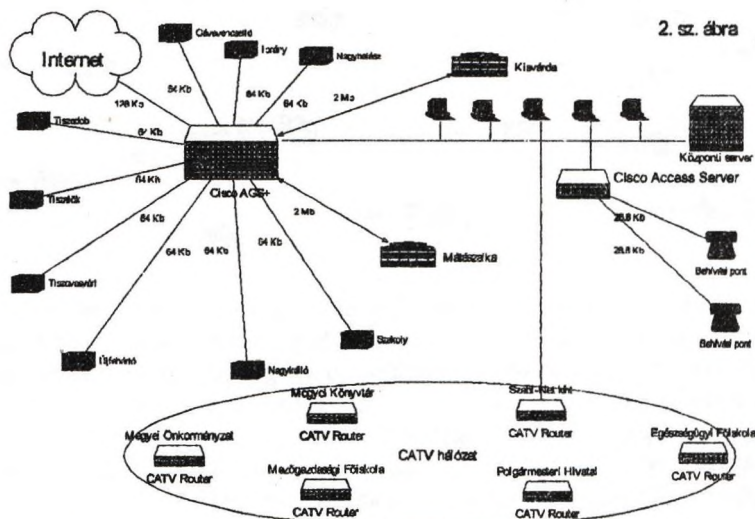
Elmondhatjuk, hogy a megyében az internet alapú kommunikáció polgárjogot nyert a hétköznapi életben. Ez nagyon biztató számunkra a jövőt illetően, hogy egy működő regionális hálózat kialakuljon, amely termelő infrastruktúráként szolgálja a gazdaság fejlődését és hozzájárul a távközlési és kábeltéves hálózat nyereségességéhez.

Szab-I-Net Program Technikai lehetőségek

A városi számítógépes hálózat létrehozásában nagyon nagy szerepet kap a városi kábeltévé hálózat. Ennek a kialakítása nagyon magas műszaki követelményeket támaszt a kábeltévé hálózat működtetőivel szemben és egy nagyon erős kooperációt feltételez az Internet szolgáltató és a kábeltévé hálózatot működtető cég között. Ahhoz, hogy egy kábeltévéhálózaton ezt a fejlesztést meg lehessen valósítani három alapvető kritériumnak kell a kábeltévéhálózatnak megfelelni.

Csillagpont topológiába kiépített hálózat (minden felhasználóhoz egyedi vezetéken jut el a jel, így nem tudják egymást zavarni a felhasználók).

Visszirányú erősítésnek kell élnie a hálózaton (a jeltovábbítás nem csak a egy pontból terjed a felhasználók felé, hanem a rendszer végpontjai is tudnak jelet küldeni a központba).



Minőségi követelmény (csak a technológiailag korrekten szerelt, zajvédett hálózaton van esély a kommunikáció kialakítására).

Nagyon fontos lenne, hogy a jövőben csak olyan kábeltelevíziós hálózatok épüljenek, amelyek teljesítik a fenti kritériumot.

Szab-I-Net Program Fejlesztési lehetőségek

Az eddigi fejlesztések további kiaknázása a távközlési ISDN szolgáltatás igénybevétele és más béreltvonali alkalmazások kiaknázása lehet. Alapvető célunk a továbbiakban is az Internet kultúra terjesztése és az olcsó hozzáférés biztosítása.

Jelenleg a Szab-I-Net Kht. szolgáltatása minőségben nem marad el a megyében jelen lévő profitorientált szolgáltatókétól. Árban ugyanazon szolgáltatást a versenytársak árának 40 %-ért tudjuk biztosítani. Ezzel a megyében sokkal többen tudják megfizetni az Internet szolgáltatást.

Ha megpróbálunk egy kicsit a jövőbe tekinteni, akkor lelki szemeink előtt egy olyan vízió képe látszik, hogy Magyarország városaiban a kábeltévékre alapozva nagysebességű városi számítógépes hálózatok működnek. Ezek a hálózatok között nagysávszélességű kapcsolatok jönnek létre és kapcsolódnak hozzá a vidéki rendszerek bérelt vonalakon keresztül vagy normál telefon modemek segítségével. Azokban a városokban, ahol a kábeltévé alapú hálózat segítségével nagy ügyfélkör alakul ki, szükségessé válhat a jóval költségesebb és nagyobb kapacitású optikai hálózat kialakítása, mert a helyi piac már megtudja azt fizetni, a működtetését fenn tudja tartani és értelmes alkalmazásokra igénybe is tudja azt venni.

Az Intelligens Város alkalmazásainak kialakítása jelenleg folyik Nyíregyházán, amelynek keretében egy széleskörű összefogással igyekszünk olyan új szolgáltatásokat bevezetni, amelynek közös jellemzője lesz a digitális kultúra alkalmazása. Erre a legjobb példa a városi információs kioszk felállítása, amely intranet jellegű hozzáférést biztosít a városi hálózat tartalmi szolgáltatásaihoz. A jövőben meg lehet találni ennek a megyei alkalmazási lehetőségét is. Az Infokioszk érintőképernyős terminálján keresztül elérhetőek a városi és megyei információk. Azokat, akik még csak kóstolgatják az internet hálózat lehetőségeit és nem elégednek meg az Infokioszk jellegéből adódó szűkített lehetőségekkel, Nyíregyházán működő Cyber Café várja. A megye lakossága felé szeretnénk segíteni az önkormányzatokat, hogy elsősorban a kistelepüléseken technikai segítségünkkel ki tudjanak alakítani Teleházakat.

Megalakult és megkezdte működését a Városi Kommunikációs Tanács Nyíregyházán amelynek célja, hogy a város és a régió számára fontos tartalmi szolgáltatások kialakulását ösztönözze, segítse elő.

Új alkalmazások kialakítására nyílnak lehetőségek a távoli munkavégzés és a felsőfokú intézményekre alapozott távoktatási formák kidolgozásával is. Eh-

hez nagyon fontos, hogy beinduljon a városi számítógépes hálózat elérésének egyéni módja is. Ezzel tömegessé válhat a városi kommunikáció. Ehhez nagyon fontos lenne az országos fejlesztések összehangolása. Az országban még nagyon kevés helyen indult meg hasonló fejlesztés (Budapest V. kerület, Pécs, Szolnok, Eger, Gyöngyös), amelyek vagy a mi vagy a budapesti kollégák munkájának eredménye a helyi szakemberekkel közösen. Sajnálatos módon szembe kell néznünk egy nem túlságosan vígasztaló képpel, hogy az országban található kábeltévé hálózatoknak csak a töredéke alkalmas erre a technikára nagyobb és költségesebb beavatkozás, átépítés nélkül.

A Szab-I-Net Kht. pályázatot nyújtott be a Távmunka Kht. pályázatára csökkent munkaképességűek távmunka eszközökkel történő alkalmazására. Lehetőséget látunk arra is, hogy a jövőben az EgyMásért Rehabilitációs, Szakképző és Foglalkoztató Programban kiképzett gyerekeket is tudjuk foglalkoztatni. Nagyon nagy mennyiségben kellene feldolgoznunk és felvinnünk a hálózatra a jelenleg még papíron tárolt információkat, hogy ezeket a megye egészével meg lehessen osztani és ezzel a megye versenyképességét fokozzuk.

Nagyon fontos cél az önkormányzati alkalmazások megjelentetése a hálózaton. Itt nem gondolunk teljesen új dolgokra, csak szeretnénk a mai gyakorlatot gyorsabbá és könnyebben elérhetővé tenni azzal, hogy virtuális ügyintéztést, információszerzést teszünk lehetővé a hálózaton.

A hálózat fejlesztésére a fokozatosan bekapcsolódók is nagyon nagy hatással vannak, mert elsősorban a tartalmi szolgáltatások kialakításával bővítik az információs kínálatot is. Tehát, nem tudjuk pontosan megfogalmazni, hogy milyen lesz ez a hálózat 4-5 év múlva. Abban bízunk, hogy az elindított fejlesztések fokozatosan bővülnek és megismételhetjük azt a fejlődést, amelyre a legfejlettebb államokban találunk példákat.

A mi feladatunk most a hálózat kialakítása és a digitális kommunikáció felgyorsítása Nyíregyházán és Szabolcs-Szatmár-Bereg megyében.

Számítógépi grafikák

dr. Oláh-Gál Róbert
Infohar R.T. Csíkszereda

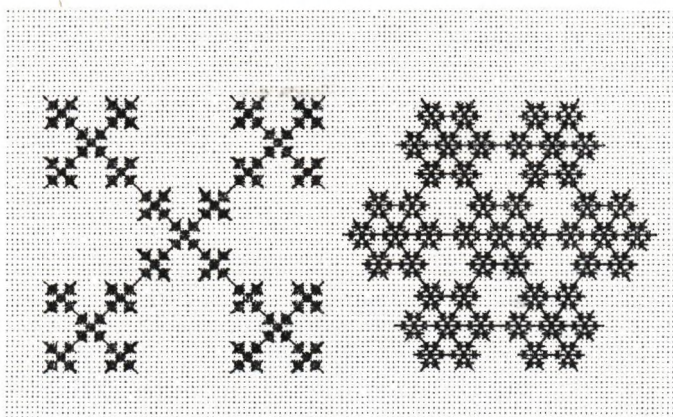
A SZÁMOKT 1996-os rendezvényén való hozzászólásomban bemutattam, hogyan lehet megrajzoltatni számítógéppel 4-dimenziós objektumokat. Tehát a számítógép egyik funkcióját úgy is elképzelhetjük mint a mikroszkópét, amellyel betekinthetünk új dimenziókba, új struktúrákba.

A számítógépnek ezt az oldalát az oktatásban is hasznosítani tudjuk és hasznosítják is. Most egy pár olyan képet szeretnék bemutatni, amelynek szerintem esztétikai és matematikai értéke van.

A mi kép és térszemléletünk a megszokott mindennapi látványainkhoz igazodik. De a világ ennél jóval gazdagabb és a felszínen lejátszódó folyamatok megértéséért, be kell pillantanunk egy mélyebb vagy magasabb szintű struktúrába. A fizikusok és matematikusok azt tartják, hogy a Világmindenség törvényei egyszerűek és néhány alapigazságra épülnek. Mi ezt a folyamatot szeretnénk egy pár vázlattal szemléltetni. Kiindulásul vegyük a lehető legegyszerűbb alakzatokat, mint amilyen a háromszög, négyszög, szabályos ötszög, hatszög, szabályos n -szög, és alkossunk ebből komplex világot elemi elvek segítségével (ahogy ez lezajlik a biológiában is).

Konkrétan: a fent említett elemi sokszögekből első lépésben készítsünk fraktálokat. Meg fogjuk látni, hogy a háromszög esetén megkapjuk a híres Sierpinski csipkét, négyzet esetén megkapjuk a Vicsek fraktált, a többi általánosítás tőlem származik.

Két egyszerű elvet vettem alapul e képek megrajzolásánál, egyik a rekurzivitás a másik a transzformáció.



A Vicsek-fraktál öt részből áll (és ezt a dimenziója is tükrözni fogja), ezért a v - eljárás ötször fogja hívni önmagát, minden esetben más, más pontban, de ugyanakkora méretben. Az első hívás a középpontban történik, utána megrajzoljuk a jobb alsó részt (tekintettel arra, hogy x a képernyőn jobbra, y lefele nő) ami abból áll, hogy vonalat húzunk a középponttól, az $x+h$, $y+h$ pontba, majd egy kis ágacskaát húzunk rá keresztben. Ezután a nagy ág és az ágacska metszéspontjába visszük a grafikus kurzort és átadjuk a továbbrajzolást az előtte lévő generációnak (az apának), a méretet háromszorosan csökkentve. Nagyon fontos, hogy a rekurzív hívás után a kurrens pontot

visszavigyük az eredeti kezdőpontba. Ezt még háromszor megismételjük, a négyzet átlóinak irányában, meg is kapjuk a kívánt Vicsek-fraktált. Természetesen a leállást az i szabályozza, mert, ha $i=0$ akkor a v eljárást elhagyjuk. A főprogram banális. Ezt a fraktál rajzolást az egyszerűsége és a tanulás szempontjából tartjuk közlésre érdemesnek. Úgy tudjuk, hogy Vicsek Tamás bizonyította be a hópehely fraktál tulajdonságát és ez adta az ötletet egy ugyanilyen egyszerű hópehely fraktál megrajzolására. A hópehely fraktál még a Vicsek-fraktálnál is egyszerűbb, de szépsége nem marad el tőle. A lényege, hogy nem négy, hanem hat irányban nő, mindig egy-egy hatágú keresztet növesztve az aktuális pontban. A megszerkesztése valamivel könnyebben érthető, mint négyzetes elődjének, mert mind a hat rekurzív hívás előtt csak egy vonalat húzunk, majd egy harmadot visszajövünk az ágon, a többit rábízunk az „ősökre”. Itt sajnos számít, hogy milyen képernyőfelbontással dolgozunk, mert egy körbeírható hatágú kereszt koordinátái irracionális számok. Ha a Vicsek-fraktált egy négyzetbe ágyazzuk, akkor észrevehetjük, hogy a négyzet oldalának háromszoros növelésével, a fraktálegységek száma ötszörösére nő, míg a négyzetek száma kilencszeresére. A fraktáldimenzió épp a növeléssel szembeni viselkedés mértékét fejezi ki:

$$d = \frac{\log 5}{\log 3} = 1,464973494$$

mert 3-as növelésnél az alapegységek száma 5-re növekedett, míg sík esetében a négyzetek száma 9-re nőtt és ezért a sík fraktáldimenziója

$$d = \frac{\log 9}{\log 3} = 2$$

Ha a hópehely fraktálnál az alapegységet tartalmazó kis kör sugarát háromszorosára nagyítjuk az alapegységek száma hétszeresére nő, míg a kis kör területe kilencszeresére nőtt. Ezért a dimenziója:

$$d = \frac{\log 7}{\log 3} = 1,771243686\dots$$

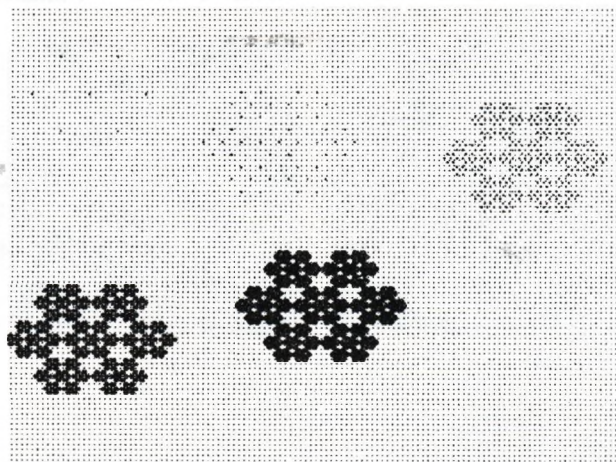
Ez közönséges görbénél 1, mert ott a sugár háromszoros növelésénél a görbe vonal is háromszorosára növekedik. Tehát a dimenzió

$$d = \frac{\log N}{\log r}$$

ahol r a nagyítás mértéke, N e mellett a nagyítás mellett keletkezett egységek száma.

Amint már szóltunk róla ezt lényegében öt rekurzív hívással rajzoltattuk meg. Csak a nehézség az, hogy jól oda kell figyeljünk a kis András-keresztek

megrajzolásának koordinátáira. Vagyis a program megtervezésénél nekünk kell a koordináták változását követnünk, hogy hogyan adjuk át a rekurzív hívásokat. Ha pedig az első próbálkozásunkra a program nem azt rajzolja amit elterveztünk, akkor bizony alaposan meg kell bogarásszunk, hogy hogyan adjuk át a koordináták értékeit. Ezzel szemben van egy elegánsabb megoldás is felhasználva a transzformációk fogalmát. E szerint a Vicsek fraktál öt része azt jelenti, hogy mondjuk a szülő reprodukálja önmagát 3-szor kisebb méretben és öt különböző helyre. Ez az öt különböző hely egy négyzet közepe és négy sarka. Ennek matematikai leírása egyszerű! Külön kell leírni az x koordinátára és külön az y -koordinátára, a programban a, b, c, d és e ezt az öt helyre való eltolást jelenti. Vagyis az eredeti képet harmadára csökkentjük és 5 helyre eltoljuk. Ezzel a kis eszköztárral jól el lehet kísérletezni. Csökkentsék négyszer kisebbre, vagy n -szer kisebbre és tolják el különböző akár véletlenszerű helyekre. Sorozatban gyárthatunk szabályos és szabálytalan, véletlenszerű fraktálokat. Mivel ez a transzformáció pontszerű, vagyis pontról pontra kell megrajzoljuk a transzformált pontokat ezért itt nincs értelme a vonalhúzásnak. Csak pontok megjelenítésével kell dolgozzunk. Az ötlet az, hogy egy alakzatot csak akkor rajzolunk meg, ha már egy ponttá zsugorodott. Ekkor rekurzíven visszafűződik az egész struktúra. Itt is öt rekurzív hívás és öt pont kivillantása a program lelke.

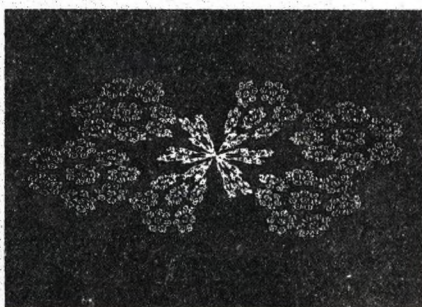
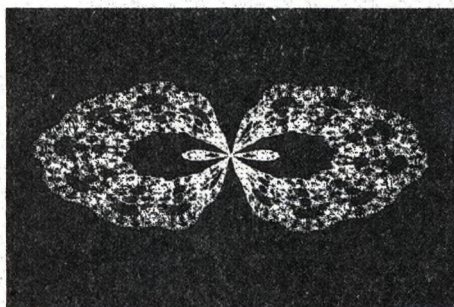
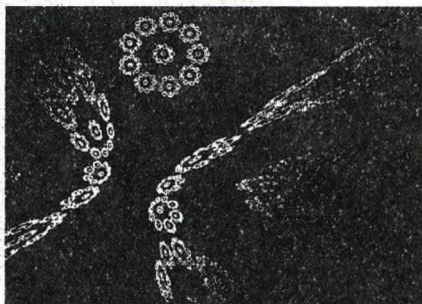
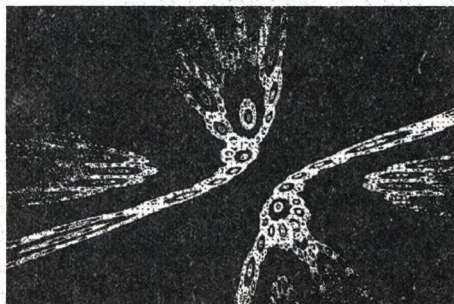


Tehát van egy egyszerű függvényünk, amely pontot tesz a szabályos m szög m csúcsába. Ezt pedig rekurzíven hívjuk az m sokszög m csúcsában, aszerint, hogy hány generációt rajzoltatunk meg. Ezt tömören egy „nyúlfarkú” eljárás megoldja. Ami fontos, hogy az így keletkezett képet egyszerű ponttranszformációkkal átszűrjük. Ilyen egyszerű transzformáció például a Descartes-i rendszerről az áttérés a polárkoordinátákra. Ugyanis mi csak a

paraméter síkot transzformáljuk! Van olyan transzformáció is amely matematikailag nem megengedett, de itt annál szebb képet ad minél szabálytalanabb a ponttranszformáció. (Ez olyan mint a nullával való osztás, vagy ha a transzformáció "jakobiánusa" nulla, szingularitási pontok, stb.)

A transzformáció közül a projektivitást részesítettem előnyben (bár másokat is kipróbáltunk, amelyek Finsler, Riemann, nem-euklideszi világot ad!).

Azért fontos a projektivitás megértése, mert a Világmindenség szerkezetében biztos, hogy szerepet kap. Gondoljunk csak az ellipszis pályákra, az árnyékokra, a Nap vetítő sugaraira, magára a perspektívára.



Vizuális fejlesztő környezetek oktatása

Kovács Lehel

Babeş-Bolyai Tudományegyetem, Kolozsvár

A magasszintű programozási nyelvek gyors fejlődése, a grafikus Windows környezet és az általa szolgáltatott lehetőségek megjelenése, az objektumorientált programozás tökéletesítése gazdag háttérrel biztosított a gyors fejlesztő környezetek (*RAD – Rapid Application Development*) kialakítására. Ezek a környezetek, vizuális alkotóelemeket használva, napjaink programozási felületeivé váltak. Ilyen vizuális fejlesztő környezetek a *Visual Basic*, *Visual Prolog*, *Visual FoxPro*, *Visual C++*, *Microsoft Developer Studio*, *Borland Delphi*, *Borland C++ Builder*, *Borland J++ Builder*.

Jelen pillanatban a Babeş-Bolyai Tudományegyetem Informatika IV. évének az Integrált Programozási és Tervezési Felületek tantárgya fedi le, két féléven keresztül, a vizuális fejlesztő környezetek oktatását, de amennyire lehetőség van rá, már a középiskolában is megkezdődhet a tanulók bevezetése e „parttalan” témába.

Az „alapozás” tulajdonképpen két előfeltételre támaszkodik:

Windows alapismeretek (az ablakok, vezérlőelemek fogalma valamint a Windows filozófiájának ismerete – ezek leginkább a Windows gyakori használatával sajátíthatók el).

A háttér-programnyelv DOS vagy Windows alapú változatának ismerete (minden ilyen vizuális fejlesztő környezet valamilyen, már meglévő programozási nyelv továbbfejlesztése, pl. a *Borland Delphi* a *Borland Pascal 7.0*, a *Visual FoxPro* a *FoxPro 2.6* programozási nyelvek továbbfejlesztése).

Ha a diákok ilyen természetű ismeretekkel rendelkeznek, akkor megkezdődhet a környezet tulajdonképpeni ismertetése. Az ismertetésnél több fázist kell bejárni, mindegyikben más-más módszerre támaszkodva. A következőkben ezeket a lépéseket fogjuk ismertetni, kiemelve a használt módszert és a közölt ismeretanyagot. A lépések így csoportosíthatók:

1. A környezet filozófiájának, alkotóelemeinek ismertetése.
2. A háttérben működő programozási nyelv ismertetése.
3. A használt háttér-objektumok, vizuális alkotóelemek bemutatása.
4. Az illető fejlesztő környezet grafikus lehetőségeinek kiemelése.
5. Az adatbázis-kezelési lehetőségek bemutatása.
6. Az ismeretek elmélyítése, a felület összehasonlítása más, már ismert felületekkel.

A továbbiakban, a **Borland Delpbi** környezetet véve kiindulópontként, felvázoljuk a fent említett lépéseket. Feltételezzük, hogy az oktatás két síkon történik: elméleti előadás valamint laborgyakorlat.

Az első lépés tehát a környezet filozófiájának, alkotóelemeinek ismertetése. A legszerencsésebb választás itt egy összevont óra, ahol lehetőség van arra, hogy számítógépek használata mellett, táblára összesítve mutassuk be a rendszert. A legfontosabb a környezet pontos definíciója: *A Delpbi vizuális, komponens és formalapú, windowsos alkalmazásfejlesztő eszköz, amely a gyors (majdnem csak az egér használatával történő), üzembiztos alkalmazásfejlesztés mellett testre szabható adatbázis-hozzáférést is biztosít. A Delpbi felülete egy jól megtervezett objektumorientált programozási nyelvre támaszkodó windowsos programozási környezet (IDE – Integrated Development Environment). Majd pontosítunk, ismertetjük, hogy miért is illeszkedik tökéletesen a fejlesztő környezet a Windows csemény- és ablakorientált rendszeréhez. Ismertetjük a környezet alapfilozófiáját. Itt tökéletesen felhasználhatjuk az előzetes Windows ismereteket: *Az alkalmazásfejlesztés alapja a form, amely egy ablakot ír le (egy üres ablak, munkaasztal, amelyen kiforrólódik, kialakul az alkalmazás). A programozás nem más mint egy ilyen form definíciója, illetve komponensekkel (alkotóelemek – Windows vezérlők, gombok, menük, címkék, ikonok, stb.) való betelepítése, megadva azt, hogy az egyes komponensek milyen tulajdonságokkal rendelkeznek, illetve hogyan kezelik le az őket érintő eseményeket.* Miután így bevezettük és értelmeztünk egy pár lényeges fogalmat, rátérhetünk a működési elv felvázolására: *Két fontos lépést különböztethetünk meg: az alkalmazás tervezését és az alkalmazás futtatását. A Delpbi objektumorientáltsága megengedi, hogy tervezési időben is hozzáférjünk az objektumokhoz, komponensekhez.* Majd megmagyarázzuk, hogyan is épül fel a működési hierarchia: *A Delpbi az alkalmazások fejlesztése során projekteket hoz létre, ez tárolja az egyes alkalmazásokra vonatkozó információkat, a hozzátartozó egységeket (unit), valamint a formokat hozza létre. A projektek .DPR kiterjesztésű állományokba kerülnek. Minden egyes formhoz egy egység tartozik, ez írja le a működését (.PAS állomány) valamint egy bináris leírás, amely a formon levő objektumokat, komponenseket jelenti meg (.DFM állomány).* Ezen információk birtokában rátérhetünk a környezet bemutatására (számítógépen). Most, osztályozva, a következő elemeket mutatjuk be: *a menüsor, a komponenspaletta, a form, a szövegszerkesztő, az eszközzaletta, az objektumkezelő, részletesen rátérve minden egyes elem működésének, lehetőségeinek magyarázatára.**

Második lépés a háttérben működő programozási nyelv bemutatása. A használt módszer a komparatív elemzés. Összehasonlítjuk a háttérnyelvet a már ismert DOS vagy Windows alapú régebbi változatával, erősen kiemelve

a különbségeket, nagy hangsúlyt fektetve az újonnan megjelent nyelvelemekre. Az órát most a következőképpen építjük fel: egy elméleti előadás, amelyen bemutatjuk a nyelvet, és egy laborgyakorlat, ahol a diákok alkalmazzák az elméleti ismereteket, egyszerű példákat oldva. Először is a nyelv típusait hasonlítjuk össze, hiszen egy új környezet filozófiája nagymértékben befolyásolja a típusok kialakulását és ezáltal a nyelvben használható adatstruktúrákat. Részletesen rátérünk az új típusok magyarázatára: *Megjelenik a Cardinal típus, amelynek mérete függ a rendszertől (16- vagy 32 bit), a Variant típus, amelynek változói dinamikusán változtatható értékeket vehetnek fel.* Majd rátérünk a string típus vizsgálatára: *A Borland Delphi a string típust is kibővítette. Tulajdonképpen három típusú stringről beszélhetünk, amelyeket a fordítóprogram automatikusan kezel és konvertál: a rövid stringek (Pascalban is megszokott maximum 255 karakterből álló láncok), a hosszú stringek (a méretük tetszőleges lehet) és a Windows 95 által támogatott Unicode ábrázolásmódban (karakterenként 2 byte) ábrázolt stringek.* Természetesen itt következnek a megértetésre szolgáló, magyarázó jellegű példák. A típusok ismertetése után rátérhetünk az eljárások és függvények új paraméterátadási lehetőségeinek bemutatására. *A Pascal nyelvben megszokott kimeneti (var), illetve bemeneti paraméterek mellett lehetőség adódott a konstans (const) paraméter átadásra is. A konstans paraméter értéke az eljáráson vagy a függvényen belül nem változtatható meg.* A jelenség megmagyarázásának érdekében fordítóprogram-elméleti ismeretekre van szükség: *A konstans paraméterekről a fordító nem készít lokális másolatot a veremben, emiatt sokkal hatékonyabb kód jön létre tömb, rekord, string paraméterek használata esetén.* Megfelelő magyarázat kíséretében rátérünk a nyitott stringek (Open-String) és nyitott tömb (array of type) illetve konstans tömb (array of const) paraméterátadások tárgyalására. Ezt követően a visszatérő függvényértékek új típusait mutatjuk be: *A Pascalban megszokott típusok mellett a Delphi összetett típusú (rekord, halmaz, tömb, stb.) értékeket is szolgáltathat vissza.* A magyarázattal el is érkeztünk az előadás csúcspontjához, a Delphi objektumorientáltságának a bemutatásához: *A Delphi a visszatérési értékeket implicit mutatóként kezeli, a típusokat pedig osztályokkal írja le.* Az objektumorientáltság új lehetőségeinek feltárása megalapozott háttérismereteket követel a Pascal objektumorientáltsága teréről és számos fogalom (osztály, objektum, példányozás, mutató, adatmező, metódus, konstruktor, destruktorkonstruktor, öröklődés, polimorfizmus stb.) ismeretét feltételezi, amit Pascalból, vagy akár más objektumorientált nyelvekből (C++, SmallTalk, stb.) meg lehetett szerezni. Amennyire a diákok nem rendelkeznek ezen háttérismeretekkel, akkor ezeket is itt kell megalapozni, egy eléggé nagy kitérővel tisztázni kell ezeket a fogalmakat. Erre az esetre a legalkalmasabb, ha a Borland Pascal 7.0 objektumorientáltságát vesszük kiindulópontként és lépésről lépésre, „tégglánként”

építjük fel a Delphi objektum-elméletét. Részletesen rátérünk, hogy az újonnan megjelent *class* típus miben tér el a Pascal *object* típusától. *A Delphiben bevezetett class típus felépítésében és használatában is hasonlít az object típusra. Lényeges különbség az, hogy a class típus példányai dinamikusan jönnek létre és minden új típusnak van öse, a TObject. Az adatmezők mellett a class típusok tulajdonságokat (property) is tartalmazhatnak.* Ebben a kontextusban elmondhatjuk, hogy a tulajdonság definíciója az osztályban egy nevet, egy típust és egy műveletet tartalmaz. A tulajdonságok (amelyeket egyes könyvekben jellemzőknek is neveznek) képezik tulajdonképpen a Delphi által támogatott komponens-orientált fejlesztés alapját. Ezek a jellemzők mind tervezési, mind futtatási időben elérhetők. Ezzel a kijelentéssel elérkeztünk a Delphiben használt adatrejtési elvek ismertetéséhez. *A Pascalból ismeretes private és public adatrejtés mellett a Delphi bevezeti a protected, published és automated mechanizmusokat.* Majd részletesen megmagyarázzuk az új lehetőségeket, rátérve arra, hogy a *published* adatrejtési mód az, aminek értelmében az objektumkezelő tervezési időben is láthatja az objektumok adatmezőit, tulajdonságait. Legszerencsésebb az, ha táblázatot készítünk ezekről a mechanizmusokról, feltüntetve a láthatósági területeket. Ezekután rátérhetünk az objektumpéldányok létrehozására, az objektumhierarchia, az absztrakt metódusok, osztályoperátorok és osztálymetódusok ismertetésére. Külön előadás témáját képezi a Delphi kivételkezelése. Ez teljesen új mechanizmus, ismertetését az alapoktól kell kezdeni: *A kivételek (exception) olyan hibás események, amelyek megszakítják az alkalmazás szabályszerű futását. Ilyenkor a vezérlés a kivételkezelőnek adódik át.* Majd részletesen ismertetni kell a kivételkezelőt (az *Exception* osztályt, a *raise* utasítást, a *try...except...[else...]* *end* blokkot, az *on...do* szekvenciát, a *try...finally* blokkot). Az elméleti előadást összeállított, egyszerű feladatokból álló laborgyakorlat követi.

Harmadik lépés a használt háttér-objektumok, vizuális alkotóelemek bemutatása. Szerencsés választás ismét a számítógépen történő bemutatás. Révén ez is teljesen új, eddig ismeretlen információhalmaz, ismertetését az alapoktól kezdjük. Leghatásosabb, ha a Delphi komponenspalettája kerül „elemzőasztalra”. A komponenseket (alkotóelemek) a Delphi palettája csoportosítva tartalmazza. Ezt a csoportosítást követjük mi is az előadáson, kiemelve az új fogalmakat (*látható, nem látható komponensek*) illetve az érdekesebb, fontosabb komponensek tulajdonságainak bemutatását. Ezután rátérhetünk a háttérkomponensek, a VCL (Visual Class Library) objektumainak ismertetésére. Itt kerül sorra a VCL osztályhierarchiája és a „független” eljárások, függvények bemutatása is.

Negyedik lépésként rátérhetünk a Delphi grafikus lehetőségeinek feltárására. A GDI (*Graphical Device Interface*) által támogatott grafikának

tulajdonítható a Windows jellegzetes kinézete is. Kezdetként a GDI lehetőségeit ismételjük el (TrueType fontok, Bitmap-ek, Ikonok), meg lehet említeni a hagyományos, API (*Application Programming Interface*) függvényeket, eljárásokat is. *A Delphi az összes grafikus objektumot és megjelenítőt a Graphics unitban tárolja. Az eszközkapcsolatot és magát a rajzolási alapegységet is megvalósító objektumot a TCanvas osztály képezi. Minden speciális megjelenítő objektum (TForm, TPrinter, TImage, stb.) tartalmaz egy Canvas nevű, TCanvas típusú tulajdonságot. A rajzolás ezen a Canvas objektumon történik, amely nem más mint az eszközkapcsolat objektumorientált megfogalmazása. A Canvas objektumot egy festőkészletként képzelhetjük el. A Canvas tulajdonságai a rajzolási attribútumokat, a rajzeszközök és a rajzvászon jellegzetességeit állítják, a metódusok pedig a konkrét rajzoláshoz szükséges rutinokat biztosítják. Ilyen értelemben észszerű a következő bemutatási csoportosítás: tollak (A TPen osztály), ecsetek (a TBrush osztály), fontok (a TFont osztály), bitmap-ek (TBitmap), és végül maga a Canvas, amely összesíti az előzőket. Az előadást laborgyakorlat követi.*

Az ötödik fázisban rátérhetünk a Delphi adatbáziskezelő rendszerének bemutatására. Itt is az alaptól kezdve „építkezünk”. Először a BDE (*Borland Database Engine*) filozófiát ismertetjük, részletesen rátérve minden egyes szintre, kiemelve az ODBC (*Open Database Connectivity*) és IDAPI (*Integrated Database Application Programming Interface*) kapcsolatokat. Majd rátérhetünk arra, hogy hogyan is valósul meg Delphiben az adatbáziskezelés, kitérve az ide tartozó komponensekre (*TDatabase, TTable, TQuery, TDBGrid*, stb.). Az adatbázisok lekérdezésére az SQL nyelvet ismertetjük, lebontva a két alrendszerre: DDL (*Data Definition Language*) és DML (*Data Manipulation Language*). Az SQL szabványok és parancsok ismertetése után rátérünk a *SELECT* parancsra, lévén ez a legfontosabb a lekérdezés szempontjából. Részletesen ismertetjük a parancs szintaxisát és a használható függvényeket is. Természetesen ezt az előadást is célirányos laborgyakorlat követi, amelyen a diákok elmélyítik a szerzett ismereteket.

Végül, hatodik lépésként rátérhetünk a „magasabb szintű” ismeretek, fogalomtársítások bemutatására is. Itt mutathatjuk be az objektumorientáltság magasabb szintű tulajdonságait: a parent-owner típusú relációkat, a perszisztencia fogalmát, a container fogalmát, az objektumszervereket és protokollokat (OLE, CORBA) valamint az Internet, ActiveX, DirectX lehetőségek kihasználását is.

LOGO-programozási módszerek Heurisztikus algoritmusok alsó tagozatosok részére (oktatóprogram bemutatása)

Oláh Anna

Kőrösi Csoma Sándor Általános Iskola, Budapest

Mottó:

*A gyermek születésétől kezdve építő ember.
Seymour Papert*

Az ember számára örök titok marad az, hogy a gyermek születésekor milyen modelleket hoz magával, hogy melyik kortól kezdve képes újakat asszimilálni. Annyi azonban bizonyos, hogy a gyermek genetikusan ÉPÍTŐ EMBER. Alig pár hónapos korától kezdve építi a körülötte lévő tárgyi világot. Nem véletlen, hogy a LEGO egyik kedvenc játéka majdnem minden gyermeknek. De hogyan építi gondolatvilágát? Hogyan alakulnak ki sémái, intelligenciájának alappillérei? Abban az információs közegben amelyben élünk, nem mindegy, hogy a gyermek hány éves korban, milyen hatékonysággal sajátítja el azokat az alapkészségeket amelyeket a computerizált jövő megkövetel. Ezeket nem örökölhette őseitől, nem tanulhatta meg szüleitől azért, mert az ők információs közegük más volt a mainál.

Jelen oktatóprogram szándékaink szerint a programozói mentalitás vonatkozásában ezt a folyamatot hivatott gyorsítani, vagy elmélyíteni már a 6-10 éves korosztálytól kezdve.

Az elfogadásra került NAT biztosítani kívánja a jövő generációi számára az informatikus képzést. A hatékony informatika oktatás sokoldalú pedagógust kíván. Egyidejűleg szükségesek a tanuláselméleti, informatikelméleti alapterületek, pszihopedagógiai, valamint módszertani eszközök is. Egy ilyen eszközt kínál a LOGOWRITER programnyelv amelynek a pedagógiai hatékonyságát Seymour Papert világította meg a legfényesebben.

Számítógépes oktatói gyakorlatom során jó néhány általánosítható tapasztalatot gyűjtöttem össze. Ezeket egyesítettem abban az oktatóprogramban, amivel a 6-10 éves korosztály számára a problémamegoldás lépéseit életkori sajátosságaik szintjére próbáltam lebontani. Az oktatóprogram forrásnyelve tehát a LOGOWRITER.

A "LOGO-BARKÁCSKÉSZLET" oktatói program módszertani vonatkozásai

Amint a neve is sugalja, ez az oktatóprogram „programbarkácsolásra”, LOGO építmények, eljárások összeállítására szolgál. Közel 200 apró eljárás képezi a barkácskészlet építőelemeit. Ezeket az órák alatt kialakult programozói igények szerint készítettem. A kezdetben néhány eljárásból álló készletet rögtön használni kezdtük az órákon. A 10-12 osztályos tanulók, akik ismerték a LOGOWRITER alaputasításait, újabb eljárásokat igényeltek, egyeseket ők maguk írtak, így gazdagodott a készlet.

Az eljáráslisták a FLIP oldalon olvashatóak. Kezdő felhasználók számára oktató célt is szolgálhatnak a gyakorlottabbak egyéni igény szerint tovább fejleszthetik.

A „LOGO-BARKÁCSKÉSZLET” oktatóprogramot elsősorban azoknak a pedagógusoknak ajánlom, akik második szakként oktatnak programozást, ezen belül LOGO programnyelvet azután, hogy sok évig egyéb tantárgyakat tanítottak.

Hasznos didaktikai eszköz lehet a alsó tagozaton oktató kezdő pedagógus számára is, aki hozzám hasonlóan beszerzi a meglévő LOGO tanterveket, tanmeneteket, szakkönyveket, példatárakat, játékos informatikai leírásokat, összeollózza a saját meggyőződése szerinti legjobb ötleteket és ezeket szó szerint véve kezdi oktatni a Logót.

Az oktatót az első meglepetés akkor éri, amikor a jól megtervezett óra menete felborul, attól, hogy a gyerek a gép előtt a saját észjárását követve nem úgy hajtja végre a kapott feladatot ahogy a felnőtt pedagógus kívánja. Ezt sokan közülünk fegyelmi kérdésként kezelik, hiszen ezt szoktuk meg fő szakunk oktatása során. Az a pedagógus, aki egy hasonló helyzetben leül a gyerek mellé és megkérdezi, hogy miért csinálta úgy ahogyan csinálta, az megtette a LOGO számára rossz elrontásának első lépését.

Célszerű tehát az óravázlatot a lehető legvázlatosabban körvonalazni, legtöbbször elég megfogalmazni a feladatot. A kérdés után az óra menete többnyire két ágon bontakozik ki. Van olyan gyerek akinek kezdetben nincs saját elképzelése, a többiektől, vagy az oktatótól várja a megoldást. Vannak fegyelmetlenek, akik azért se csinálják úgy ahogy a felnőtt előírja nekik, hanem saját logikájuk szerint szerint dolgoznak.

Amennyiben az oktató olyan szerencsés helyzetben van, hogy minden gyereknek saját gépe van, annak sikerül az elágazásos óramenetet úgy irányítani, hogy a kreatív csoport tagjai saját ötleteik, képességeik szerint dolgozzanak, az ötletelenekeknek egyenként mi javasolhatunk megoldási irányt – itt az irányt hangsúlyoznám – alkalmazkodva az egyéni képességszintjükhöz.

Természetesen vannak „köztes” helyzetekben levők is, akiknek csak egy kis lökés hiányzik ahhoz, hogy elinduljanak egy egyéni megoldás irányába.

Ezeknél a gyerekeknél sokszor célszerűbb volt engedni, hogy az elindulásban az ötletes gyerek adjon segítséget, ne a tanár. Ennek pszichikai háttere az, hogy a közepes gyerek a jókhoz szeretene tartozni, ezért nem szívesen szolgáltatja ki magát a tanárnak azzal, hogy segítséget kér tőle, de ugyanezt nem restelli az osztálytársától. A tanárnak nem is muszály tudomásul venni a beavatkozást, csak a végső megoldást értékelve növelheti a közepes tanuló számára oly fontos sikerélményt.

Egy másik igen fontos tapasztalatom az, hogy óra elején, amikor a feladat megoldásának algoritmusára forr – a kreativitás kibontakozásának legintenzívebb momentuma – semmiféle beavatkozássunk ne legyen, se ötlet, se tanács formájában senki számára, legfeljebb annyi, hogy a feladatot többször, egyre aprólékosabban kifejtjük, esetleg valamiféle modellt is használunk. Ezek után magukra hagyjuk őket addig amíg eldől, hogy melyik munkacsoporthoz fognak tartozni – az önállókhöz, vagy a támogatottakhoz. Ily módon oktatva a LOGO programozást sikerült közelebb kerülni a 6-10 éves korosztály, már meglévő programozói sémáihoz, ők mutatták meg azt, hogyan programoznak, építkeznek, ítélik meg a pl. 6-10 évesek összehasonlítva a 10-12 évesek észjárásával ugyanazt a feladatot tekintve. Igen nagy segítségemre voltak a 13 évesek, akiket társszerzőimnek is tekintek. Részvételükkel egy néhány fős „bírálóbizottság” a legkisebb „szépséghibát” is jelezte és azon nyomban közösen továbbfejlesztettük a programot szem előtt tartva a felhasználók esetleges elvárásait. Így alakult ki pl. az OVILOGO.lwr kínálata.

A LOGOWR programnyelv egyik fő érdeme, hogy a forrásnyelvi eljárások a rajzlap hátán folyamatosan hozzáférhetőek, így minden gyerek saját maga fejlesztheti, gazdagíthatja, változtathatja a saját, vagy mások által készített alkotásokat (*.lwr állományokat) akár sok éven keresztül.

Az OVILOGO.lwr állomány módszertani és pedagógia vonatkozásai

Alsó, felső tagozatos tanulók egyaránt kedvelik. Lényege az, hogy kiiktattuk az oly bonyolult LOGO-primitívek beírását – amint tudjuk angol nyelvi rövidítések, vagy mozaikszavak pl. SLOWTURTLE SETSH, SETPOS, SETC, SAVEPIC, LOADPIC, SAVEPAGE, ERASEPIC, SAVETEXT, LOADTEXT, stb. Egy 6-8 éves gyerek számára az értelmetlen kifejezések memorizálása és helyes leírása hihetetlen módon késlelteti, vagy teljesen elérhetetlenné teszi a motivációhoz oly fontos sikerélményt.

A leggyakoribb LOGO utasítások beírását – egy-egy apró eljárás révén – egyetlen billentyűlenyomásra korlátoztuk. (Hasonlóan a DOS-NC kapcsolatokhoz).

Az OVILOGO.lwr állományában tehát már a 6-10 évesek használják azokat az „építőelemeket”, eljárásokat, amik a bonyolult utasításokat „értelmes” billentyűkkel helyettesítik. Különlegesen sikeres a zongorává „varázsolt”

számozott billentyűsor. Számukra nem a LOGO utasítások megtanulása, elsajátítása a nyilvánvaló, elsődleges didaktikai cél. Nem úgy kezdik az ismerkedést a TEKNŐC kínálataival ahogyan a hagyományos tanmenetek javasolják. Nem az angol eredetű LOGO primitívek memorizálása révén lesznek csak képesek a teknőcöt mozgatni, rajzolni. Életkori sajátosságainak megfelelően a fő cél a játékos, mesevilághoz közel álló alkotás. Egy grafikuszenés-szöveges alkotáson keresztül fokozatosan, gyorsan megismerkednek a teknőc kínálataival. Ily módon könnyedén sikerül a 6-10 évesek számára a tanulás folyamatában oly nélkülözhetetlen MOTIVÁCIÓT magas szintre emelni.

A 6-10 évesek számára igen szép teljesítmény a menüpontok kibetűzése, a SHAPES (alakok) négyzetháló kitöltése különféle figurákkal, azok más-más mozgáselemeivel és a lapozó gombok segítségével mozgóképpé varázsolni a különálló képeket. Pl. nyíló-csukódó tulipán, tornászó pálcikaember (kéz fent-terpeszállás-vigyázz állás), kacsintó bohócfej (szomorú-vidám) stb. Az ÁLARCOK menüpontban ezekből adunk ízelítőt. Gyorsan jutottak sikerélményhez a rajzoló billentyűk használata után bekövetkező látványok révén is. Pl. színes négyzetek, téglafal, házikó, fenyő, tengervíz-kék égbolt nap stb. Néhány óra után könnyedén használták ezeket a billentyűket. Az HÁTTEREK menüpontban néhány ilyen kis képet mutatunk be. (*pic kiterjesztésű álmányok).

Örömmel „zongoráztak” a zenélő számokkal, majd a leolvasott a számskálát könnyedén írták át kottára.

A ZONGORA menüpont tartalmazza a zenélő számok eljárásait.

Az a korosztály, amelyik még nem ismeri a fok fogalmát nehezen tudja a TEKNŐC egyik alapvető utasítását, a JOBBRA FORDUL x fokot, BALRA FORDUL y fokot megérteni és alkalmazni. A mozgásfolyamatok algoritmusában azonban az egyik fontos momentum az iránybeállítás. A programozott billentyűk segítségével a legkisebbek is (6-7 éves korosztály) szinte automatikusan érezték meg az iránybeállítás, -változtatás fontosságát és kivitelezését.

Az OVILOGO.lwr tanítja a tanárt, szülőt gyermekfejjel gondolkozni, játszani. Más – sokszor egyszerűbb, hatékonyabb feladat – megoldási horizontokat nyit ki beszűkült gondolkodásunk számára. Bevonja a legkisebbeket a felnőttek dolgaiba és viszont. Általa közel kerül egymáshoz tanár-gyerek-szülő-számítógép. Meggyőződésem, hogy minden korosztály egy idő után családtagként szereti a LOGO főhősét, a TEKNŐCÖT, s általa képes lesz könnyebben megtalálni a helyét a Cyber Space az „értelmes tárgyak” világában.

A " LOGO-barkácsoló" oktatóprogram felhasználói ismertetése

A program indításakor egy RÖVID OLVASS EL.txt. típusú STARTUP automatikusan indító eljáráshívással kezdődik, majd behívja a főmenüt, amelynek a kínálatai az alábbiak:

FŐMENÜ

0. KÖSZÖNTŐ: egy rövid verses köszöntő
1. ÁLARCOK
2. HÁTTEREK – kínálatát lásd: HÁTTÉRMENÜ
3. MOZGÁSOK - kínálatát lásd: MOZGÁSMENÜ
4. ZONGORA - kínálatát lásd: ZONGORA.TXT
5. RAJZFILMEK - kínálatát lásd: FILMMENÜ
6. DALLAMOK - kínálatát lásd: DALLAMOK
7. JÁTÉKOK - kínálatát lásd: JÁTÉKMENÜ
8. SÚGÓ - kínálatát lásd: SUGO.TXT
9. KILÉPÉS
 Kiválasztás-ENTER
 Kilépés-ESC

1. Álarcok

A SHAPES négyzethálós LOGO-grafikai felületen elkészített 24 apró képet tartalmaz, amit 90 db.-ig tudnak a gyerekek kiegészíteni.

2. Háttérmenü

Az itt bemutatott alkotások az OVILOGO.lwr felhasználására adnak néhány mintát. Minden felhasználó tetszése szerint fejlesztheti, alakíthatja. Mi csupán ötletébresztő szerepet tulajdonítunk programunknak.

1. tájkép
2. hófúvás
3. óceán
4. sípálya
5. vasút
6. képeslap
7. karácsonyfa
8. tengerpart
9. sivatag
10. saját rajzod

Kiválasztás – ENTER
Kilépés – ESC

1-9 almenüpontok mindenike egy-egy előre megrajzolt LOGO-grafikát kínál, amelynek az előállításában 5-7 osztályos gyerekek is segédkeztek.

A 10. pont behívásakor a program meghívja az OVILOGO.lwr állományt, ami a billentyűk átprogramozásának eljárásait tartalmazza. Ezek segítségével a parancsmezőben feltüntetett OVILOGO-billentyű-készlettel előállíthatja a saját tervében szereplő rajzokat.

3. Mozgásmenü

A mozgásmenü pontok mindegyike a négy mozgató alaputasítással (előre, hátra, jobbra, balra) egy-egy ismétlődő folyamatot szerkeszt össze egy egyszerű mozgáshatássá.

1. halad
2. csúszkál
3. ugrál
4. forog
5. pattog
6. ereszkedik

4. Zongora.txt

Ha zongorázni és kottázni akarsz használd az alábbi billentyűket a feltüntetett sorrendben:

q-képernyőtörlés
v-kottavonal húzás
z-zongorázás kezdete

1-dó 2-re

2-mi 4-fá

5-szó 6-lá

7-ti 8-dó

9-re2 0-vége- (A hang addig szól, amíg a billentyűt lenyomva tartod.)

5. Dallamok

- főmenüpont eljárásokban megírt ismert gyerekdallamokat tartalmaz.

Dallammenü

1. kékcsebe
2. börtön
3. boszorkány
4. születésnap
5. kiskarácsony

kiválasztás-ENTER

kilépés-ESC

6. Rajzfilmek

Két rajzfilmet lehet „lejátszani”, amiket a haladó szintű ismeretekkel rendelkező tanulók készítettek.

Rajzfilmmenü

1. A torkos kisegér
2. A torkos halacska

kiválasztás-ENTER
kilépés-ESC

7. Játékok

Ez a menüpont két interaktív programot kínál. Az egyik egy járműkitaláló játék, a másik egy ügyességi teszt, amelynek során a teknőcöt egyre szűkülő labirintuson kell egy célpontig eljuttatnia úgy, hogy ne lépjen ki a pályáról.

JÁTÉKMENÜ

1. labirintus
2. autó kitalálós

kiválasztás-ENTER
kilépés-ESC

8. Sugó

Ez az oktatóprogram HELP-je. Ide visszatérve a felhasználó segítséget kaphat a billentyűhasználathoz, valamint a program működéséhez.

A könyv és oktatóprogram szerzője szeretne szervezni egy olyan oktatócsoportot (vállalja a szervezést is), ahol konzultáció keretében minden LOGO iránt érdeklődő tanítónő, oktató, egymásnak átadhatná tapasztalatait és bemutatná a BARKÁCSKÉSZLET továbbfejlesztett változatait. Várom az érdeklődők jelentkezését.

Hasznos tudnivalók

Sepsiillyefalva a Brassói-medencében fekszik, Sepsiszentgyörgytől 10 km-re, a Brassó felé vezető mellékúton. Autóbussszal közelíthető meg, amely szombaton és vasárnap a következő program szerint közlekedik:

a sepsiszentgyörgyi állomásról indul 7, 13, 16 órakor

Illyefalváról Sepsiszentgyörgyre a busz a következő program szerint indul: 7³⁰, 13³⁰, 17 órakor

Egy menetjegy ára 3500 lej.

Rendezvényünknek a KIDA Keresztyén Ifjúsági Központ ad otthont, amely kb. 70 férőhelyes. A központ telefonszáma: +40-67-351634.

Vasárnap ebéd után azok a vendégek, akik nem sietnek elutazni, városnézésre indulhatnak Sepsiszentgyörgyön. Érdemes megtekinteni a Református vártemplomot, a Székely Nemzeti Múzeumot, amelynek történelmi, népművészeti részlegei állandó kiállítási anyaggal várják a látogatókat. Jelenleg megtekinthető a sepsiszentgyörgyi fiókszervezet által szervezett EMT napok alkalmából rendezett kiállítás, amelynek témája az 1848-as műszakiak.

Sepsiszentgyörgyről közvetlen induló vasútjáratok:

Gyergyószentmiklós–Kolozsvár–Nagyvárad–Budapest (Korona): 18²³

Dés–Nagybánya–Szatmárnémeti (gyors): 0³⁴

Gyergyószentmiklós–Déda–Marosvásárhely (személy): 17⁰⁸

Sepsiszentgyörgy–Brassó: 5, 7¹⁰, 7⁵¹ (Korona), 9¹³, 12²⁸, 16¹⁷, 17⁴², 18²⁴

Sepsiszentgyörgy–Csíkszereda: 4⁴³, 7⁰², 13⁰⁶, 15³⁶ (gy), 17⁰⁸, 21⁰¹

Résztvevők

Albert László – EMT, Kolozsvár

3400 Kolozsvár, CP: 1–140, 21 December u.116; tel/fax: 064–194042;
albert_laszlo@sistec.ro

Bács Andrea – ELTE TTK, ÁSZT, Budapest

4000 Sepsiszentgyörgy, Dec.1 u., bl.15/E/15; tel: 067–322818;
bacsandi@augusta.inf.elte.hu

Baja Zsolt Pál – Spiru Haret Iskolaközpont, Székelyudvarhely

4150 Székelyudvarhely; tel: 066–216185;

Bakó Judit – EMT, Kolozsvár

3400 Kolozsvár, CP: 1–140, 21 December u.116; tel/fax: 064–194042;
emt@emt.org.soroscsj.ro

Bakó Mihály – Soros Központ, Sepsiszentgyörgy

4000 Sepsiszentgyörgy, CP: 1–24; tel: 067–314274; misi@sbnet.ro

Bálint Gáspár – Pantex Impex Székelyudvarhely

4150 Székelyudvarhely, Intrarea Călugăreni u.2/5; tel: 066–214325

Balogh Csaba – Sepsiszentgyörgy

4000 Sepsiszentgyörgy, Kriza János u. 1–4; tel: 067–312618;

Balogh János Csaba – Orbitrade Kft., Kolozsvár

3400 Kolozsvár, Dorobantilor u.78/76; tel: 064–410512;
orbitrade@mail.dntcj.ro

Barabás Péter – X-NET, Kolozsvár

3400 Kolozsvár, Avram Iancu u. 21; tel: 064–195700;

Berecki Kinga – Kovászna Megyei Kis- és Középvállalkozók Szövetsége,
Sepsiszentgyörgy

4000 Sepsiszentgyörgy, Iskola u. 13/C/15; tel: 067–351994,
fax: 067–352 425; amoba@imcv.sfos.ro; kinga@asimcov.sbnet.ro

Bocz Imola – Kolozsvár

3400 Kolozsvár, Trotusului u. 1/R1/1; tel: 064–150746;
bocz@hotmail.com

Bódi Antal – BGYTF Számítóközpont Nyíregyháza

4400 Nyíregyháza, Sóstói u.31/b; tel: +36–42–488171, 30–355494;
toni@bgytf.hu

- Bodor Melinda** – Székely Mikó Kollégium, Sepsiszentgyörgy
4000 Sepsiszentgyörgy, Iskola u.1; tel: 067-312793;
- Csoma Csaba** – Mikes Kelemen Líceum Sepsiszentgyörgy
4000 Sepsiszentgyörgy, Grigore Bălan u. bl. 21/A/7; tel: 092-700698;
csoma@usa.net
- Csörnyei Zoltán** – ELTE TTK Általános Számítástudományi Tanszék, Budapest
1088 Budapest, Múzeum krt. 6-8; tel: 1-266-9833/2978, fax: 266-5196;
csz@maxi.inf.elte.hu
- Deák Zsuzsanna** – Tamási Áron Gimnázium Székelyudvarhely
4150 Székelyudvarhely, Tamási Áron u. 52; tel: 066-212287;
- Dimén Varga László** – Bányai János Ipari Isk. Központ, Székelyudvarhely
4150 Székelyudvarhely, Pietroasa u.18/12; tel: 066-213794
- Dóczy Tamás** – Talentum Alapítvány Marosvásárhely
4300 Marosvásárhely, Furtunei u.13; tel: 065-167593; tdoczy@netsoft.ro
- Egly János** – X-NET, Kolozsvár
3400 Kolozsvár, Avram Iancu u. 21; tel: 064-195700; glytz@usa.net
- Egyed Albert** – MKM, Határon Túli Magyarok Főosztálya, Budapest
1884, Budapest; Pf. 1; Szalay u. 10-14; tel: 1-311-9630,
fax: 1-331-8780; albert.egyed@mkm.x400gw.itb.hu
- Farkas Béla** – Gr. Sc. Ind. Perspektíva, Sepsiszentgyörgy
4000 Sepsiszentgyörgy, P-ta Libertății u. 13; tel: 067-351736;
fabel@lmcv.sfos.ro
- Farkas Károly** – Arany János Gimnázium, Budapest
1118 Budapest, Bodajk u. 10; tel: 1-3194720; drfarkas@hotmail.com
- Furdek L. Tamás** – EMT, Kolozsvár
3400 Kolozsvár, CP: 1-140, 21 December u. 116; tel/fax: 064-194042;
furdek@taylor.dntcj.ro
- György Teréz** – Cosys kft. Sepsiszentgyörgy
4000 Sepsiszentgyörgy, Daliei u. 2, bl. 44; tel: 067-351619;
cosys@cosys.ro
- Herta Micola** – Márton Áron Gimnázium, Csíkszereda
4100 Csíkszereda, Bradului u. 9 A/14; tel: 066-122788;
herta@ehrma.edu.soroscj.ro

Horváth Erika – EMT, Kolozsvár

3400 Kolozsvár, CP: 1-140, 21 December u.116; tel/fax: 064-194042;
emt@emt.org.soroscj.ro

Horváth László – ELTE TTK Informatika Szakmódszertani Csoport, Budapest

1088 Budapest, Múzeum krt. 6-8; tel: 1-266-9833/2501;
lhorvath@ludens.elte.hu

Illés Juliana-Viorica – Csíky Gergely Líceum, Arad

Chişineu-Criş, Privighetorii u. 27; tel: 057-520868

Kis Jenő – Cosys kft. Sepsiszentgyörgy

4000 Sepsiszentgyörgy, Daliei u. 2, bl. 44; tel: 067-351619;
jeno@cosys.ro

Kovács Győző – Neumann János Számítógéptudományi Társaság (NJSZT),
Budapest

1054 Budapest, Báthory u.16; tel: 1-3329349, fax: 1-3689632;
kovacs@mail.datanet.hu

Kovács Győzőné – Budapest

Kovács Lehel – Babeş-Bolyai T.E Kolozsvár

2212 Szecseleváros, Petőfi Sándor u. 77; tel: 068-270351; 064-141194;
kl268@scs.ubbcluj.ro

Lőrincz Csaba – INNO 21 Consulting Kft., Kézdivásárhely

4050 Kézdivásárhely; tel: 067-364516;

Mántó Emese – INNO 21 Consulting Kft., Kézdivásárhely

4050 Kézdivásárhely; tel: 067-364516;

Márton Zoltán – Teleki Oktatási Központ, Szováta

3295 Szováta, Rózsák útja 147. sz; tel: 065-570725; tok@netsoft.ro

Matok Zoltán – Orbitrade, Székesfehérvár

8007 Székesfehérvár, Postafiók 36; tel: 22-327687, 22-327784;
zmatok@orbitrade.hu

Miklós Csaba – Tamási Áron Gimnázium, Székelyudvarhely

4250 Székelyudvarhely, Kuvar u. 43;

Molnár János – Kőrösi Csoma Sándor Iskolaközpont, Kovászna

4055 Kovászna, Ştefan cel Mare u. 40; tel: 067-340863

Nagy Imecs Hunor – Székelyudvarhely

4150 Székelyudvarhely, Kós Károly u.414; tel: 066-214327;

- Nagy Imecs Vilmos** – Spiru Haret Iskolaközpont, Székelyudvarhely
4150 Székelyudvarhely, Kós Károly u.414; tel: 066-214327;
- Nagy Judit** – Brassai Sámuel Líceum, Kolozsvár
3400 Kolozsvár, Trotusului u. 1/R1/1; tel: 064-150746;
nagy@hotmail.com
- Nagy Zsolt** – RMDSZ, Kolozsvár
3400 Kolozsvár, Densuşianu u. 6; tel: 064-414890;
- Nyíró András** – IDG Magyarországi Lapkiadó Kft.
1012 Budapest, Márvány u. 17; tel: 1-1562582, fax: 1-1569773;
nyiro@internetto.hu
- Oláh Anna** – Kőrösi Cs. Sándor Általános Iskola, Budapest
1084 Budapest, József u. 6; tel: 1-1141316
- Oláh Gál Róbert** – Infohar SA, Csíkszereda
4100 Csíkszereda, Petőfi u. 38; tel: 066-121462; ssi.hr@bx.logicnet.ro
- Pethő István** – Globinfo Kft., Sepsiszentgyörgy
4000 Sepsiszentgyörgy, N. Iorga u.2, 3/G/10; tel: 067-326067,
067-326134;
- Pető Lajos** – Baróti Szabó Dávid Líceum, Barót
4023 Barót, Gyöngyvirág u. 11; tel: 067-377130
- Piskolti Éva** – Iskola Alapítvány, Kolozsvár
3400 Kolozsvár, Densuşianu u. 6; tel: 064-414761;
- Racsó Péter** – MKM Informatikai Igazgatóság, Budapest
1884, Budapest; Pf. 1; Szalay u. 10-14; tel: 1-269-1928,
fax: 1-269-1930;
- Sima Dezső** – Neumann János Számítógéptudományi Társaság (NJSZT),
Budapest
1054 Budapest, Báthory u. 16; tel: 1-3329349, fax: 1-3689632;
- Somai József** – Iskola Alapítvány, Kolozsvár
3400 Kolozsvár, Densuşianu u. 6; tel: 064-414761;
- Szabadhegyi Csaba** – Planetary System Kft., Budapest
1031 Budapest, Miklis u. 9. IX/51; tel: 1-3678636;
plansys@mail.datanet.hu

- Szabó T. Attila** – Berzsényi Kollégium, Szombathely
9701 Szombathely, POB 170; tel: 94-324313, fax: 94-312248;
szta@rik.bdtf.hu
- Szakács László** – Mircea Eliade Líceum, Segesvár
3050 Segesvár, M. Viteazul u. 95/13; tel: 065-773543; szlaci@enc.elsig.ro
- Szalma Györfi Noémi** – EMT, Kolozsvár
3400 Kolozsvár, CP: 1-140, 21 December u.116; tel/fax: 064-194042;
emt@emt.org.soroscj.ro
- Szász János** – Kőrösi Csoma Sándor Iskolaközpont, Kovászna
4055 Kovászna, Ștefan cel Mare u. 40; tel: 067-340863
- Szatmári Mónika** – Kölcsey Ferenc Líceum, Szatmárnémeti
3900 Szatmárnémeti, Miron Costin u.12; tel: 061-713504;
fax: 061-747346; szm@lksm.soroscj.ro
- Székely Tibor** – EMT, Kolozsvár
3400 Kolozsvár, CP: 1-140, 21 December u.116; tel/fax: 064-194042;
emt@emt.org.soroscj.ro
- Szél Sándor** – RAGCL Nagyvárad
3700 Nagyvárad, Jean Jaures u. 23; tel: 059-138715;
- Szélyes Csaba** – Banc Post, Székelyudvarhely
4150 Székelyudvarhely, Szent Imre u. 45; tel: 066-217837;
- Szélyes Emőke** – Tamási Áron, Székelyudvarhely
4150 Székelyudvarhely, Szent Imre u. 45; tel: 066-217837;
- Szlávi Péter** – ELTE TTK Informatika Szakmódszertani Csoport, Budapest
1088 Budapest, Múzeum krt. 6-8; tel: 1-2669833/2501;
szlavi@ludens.elte.hu
- Tibád Zoltán** – EMT, Kolozsvár
3400 Kolozsvár, CP: 1-140, 21 December u.116; tel/fax: 064-194042;
angyal@emt.org.soroscj.ro
- Tóth Zenkő** – Székely Mikó Kollégium, Sepsiszentgyörgy
4000 Sepsiszentgyörgy, Iskola u.1; tel: 067-312793;
- Várhelyi Attila** – Cosys kft. Sepsiszentgyörgy
4000 Sepsiszentgyörgy, Daliei u. 2, bl. 44; tel: 067-351619;
cosys@cosys.ro

Vas István – Pantex Impex Székelyudvarhely

4150 Székelyudvarhely, Intrarea Călugăreni u. 2/5; tel: 066-214325

Vass Éva – Matrica Rt., Székelyudvarhely

4150 Székelyudvarhely, N. Bălcescu u. 7/43; tel: 066-213550/128;

Zsakó László – ELTE TTK Informatikai Szakmódszertani Csoport, Budapest

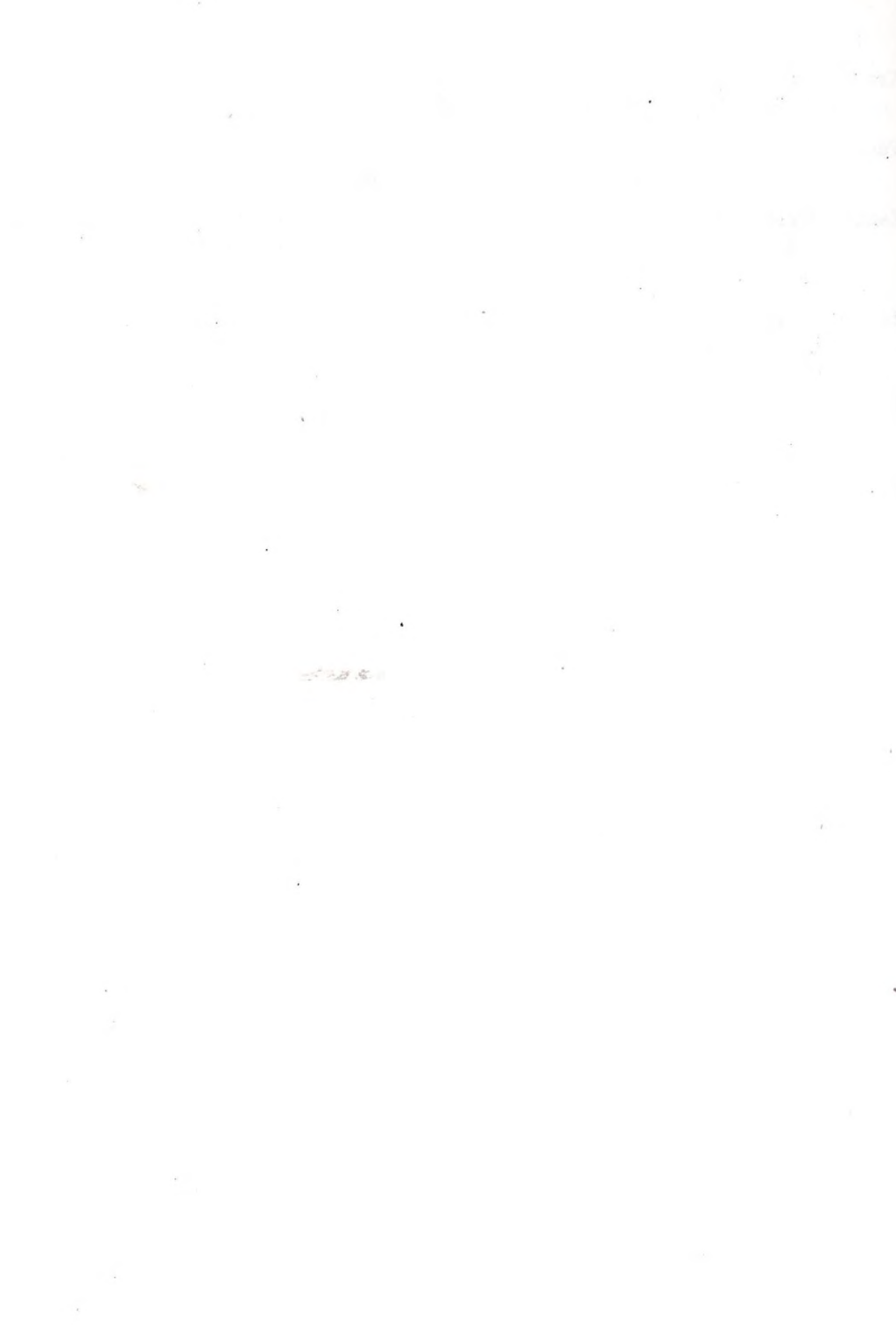
1088 Budapest, Múzeum krt. 6-8; tel: 1-2669833/2501;

zsako@ludens.elte.hu

Zsakó Lászlóné – Zrinyi Miklós Gimnázium, Budapest

1101 Budapest, Pográc z köz 3; tel: 1-2669833/2501;

zsako@ludens.elte.hu





Vann kellek - Julius Könyvtár:
Magyarország, Erdély. Ma 26 + 2 + 15 = 43,
Sulius + th. → Tendes gústi Ar. argheschi új
Erdély 2 kellek
Tudományok és kelet, Erdély új nyelv iskolák.
Júni 1998 március + 38-40 kellek, most mind a jelölés.

GESTA??

Közel TV. 4 / Schmidt
Száj - ábrák + táblák.
EMT - uel költés

Ar információs darabok a
fabrikák.
Julius Meind - új nyelv
Sulius
munkák

World Computer Congress → jelölés

Erdélyi Magyar Műszaki Tudományos Társaság - EMT -

3400 - Kolozsvár (Cluj)

B-dul 21. Decembrie 1989. nr. 116.

Tel./fax: +40-64-190825

+40-64-194042

E-mail: emt@emt.org.soroscj.ro

Bankszámláink:

Societatea Maghiară Tehnico-Ştiinţifică din Transilvania

Banca Comercială Română - Cluj

Lej: 45.10.4.66.2

USD: 47.21.4.16.45.66.0

EMT

Okéteson Műszaki Társaság.