

Turcsányiné Szabó Márta

## Tanítsunk-e programozást?

Összefoglaló elemzés a 17 éves diákok informatikai tudását mérő vizsgálatáról

### Bevezetés

Az informatika iskolai tanításának célja egyre kevésbé világos. Nehéz olyan általános és örökérvényű tananyagrészeket találni, melyek feltehetően tíz év után is hasznosak lesznek. A középiskolák jelentős hányada ráállt az ECDL vizsgák anyagának a tanítására, de a végső cél nem valamilyen általános informatikai tudás megszerzése, hanem sokkal inkább a naprakész ismeretek közlése, a ma népszerű programok felhasználói szintű oktatása, hiszen ez az, ami leginkább piacképes. Ezért a közeljövő programjai várhatóan nemcsak elvi, hanem marketing okokból is hasonlítani fognak a ma futó programokhoz, így a jövő programjai azok számára is könnyen tanulhatók lesznek, akik most tanulnak informatikát. Ennek ellenére felmerül a kérdés, hogy nincse-e az informatikának egy olyan „örökérvényű” alapja, amit érdemesebb lenne tanítani.

Az iskolák egyik jellemző adata lett (a nyelvvizsga-bizonyítványok száma mellett), hogy hány tanuló tesz ECDL-vizsgát középiskolai tanulmányai alatt. Ennek a mérőszámnak még nagyobb jelentősége lett, amióta a felsőfokú intézményekbe már nem annyira nehéz bejutni (vagyis a felvételi arány inkább az ott tanuló diákok szándékát jelzi, és nem minősíti a képzést). Általános elvárás, hogy az iskola befejezésével a tanuló minél több programot tudjon használni. Ennek megfelelően a programozás tanítása visszaszorult, sok helyen meg is szűnt. A számítástechnika ilyenfajta tanítása olyan, mintha a biológia órán csak az emberi test működéséről, a matematika órán csak a racionális számokról tanulnának, az irodalom órán pedig csak a napi sajtót olvasnák a diákok.

Ma még nem látszik, hogy az informatikai eszközök mennyire lesznek kényszerítő erejűek, térhódításuk hol áll meg, mennyire lesz mindenki számára létfontosságú, hogy ezeket *készség-szinten* tudják alkalmazni. A mai felnőtt generáció úgy tud mobiltelefonon „tárcsázni”, hogy ez a művelet figyelmének tört részét sem köti le, azonban egy-egy dokumentum kinyomtatásához vagy az Internetről való letöltéséhez (ami korántsem bonyolultabb feladat) oda kell figyelnie. Akkor tudjuk megérteni, hogy mit jelent *készség-szinten* ismerni a modern eszközöket, ha elképzeljük, hogy legalább olyan kényelmesen és épp olyan kevés figyelemmel tudunk megnézni a weben egy menetrendet, áthelyezni egy bekezdést a fogalmazásban, rádióállomást váltani, kommunikálni vagy bevásárolni, mint ahogyan ezt régen a hagyományos eszközökkel tettük.

Más szaktárgyak nagyobb hagyományokon alapuló módszertanára alapozva várható, hogy csak annak a számítástechnikának a tanítása, ami „a levegőben van”, hosszú távon valószínűleg nem jó megoldás. Az olyan – dinamikusan fejlődő – szakterü-

leteknél, mint az informatika, elengedhetetlennek látszik, hogy valami olyat is tanítsunk, ami örök. Ha meglátjuk az eddig általunk használt program legújabb verzióját, lehet, hogy alig (vagy egyáltalán nem) fogunk ráismerni. Ami ilyenkor segítségünkre lehet, az egyfajta rutin, illetve valamilyen nehezen megfogalmazható *elvi alap*, ami tudat alatt a gondolkodásunkat formálja. Érezzük, hogy hova kell kattintani a jobb egérgombbal, anélkül, hogy ezt valaha bárkitől láttuk volna. A mai programok gyakorlatilag kiismerhetetlenek, klasszikus értelemben megtanulhatatlanok, a kezelőfelületek mögött azonban egységes *navigációs filozófia* áll, melynek elsajátításával sokaknak szinte tanulni sem kell a számukra ismeretlen programok használatát.

Ahhoz, hogy sikeresen bánni tudjunk a géppel hosszú távon is, szükségesnek látszik, hogy ismerjük a működését, hogy elvárásainkat a lehetőségekhez tudjuk igazítani. Mind a közvetlen navigációs folyamatok előhívása, mind az, hogy milyen parancsokat kezdünk el keresni, milyen opciókat, beállításokat tartunk természetesnek, illetve elképzelhetőnek, mind pedig a számítástechnikai kifejezőképesség megléte vagy nem megléte jól megfigyelhetően megosztja a társadalmat, de a törésvonal nem egyezik meg a számítógépet használók és nem használók közötti választóvonalal. Meghatározó továbbá az is, hogy az egeret és a klaviatúrát mennyire tudjuk kényelmesen kezelni. Mindannyian természetesnek tartjuk, hogy ahhoz, hogy valaki automatizálódott írás-készséggel rendelkezék, már hatéves korban el kell kezdenie az írástanulást, illetve két-hároméves korától legalább firkálnia kell, számítástechnikai viszonylatban azonban a hosszú tanulási időt korántsem tartjuk nyilvánvalónak. Feltehetően a gépelés tanításának, illetve az egérkezelés elsajátításának is volna olyan módszertana, ami egy magasabb szintű, spontánabb, gyorsabb, kényelmesebb, kevésbé fárasztó, a kézíráshoz hasonlóan automatizálódott gépkézelést tenne lehetővé.

Európa keleti felében nagy hagyománya van az alapismeretek olyan oktatásának, ami gyakran nem kapcsolódik közvetlenül a mindennapi gyakorlati kérdésekhez, mégis olyan széleskörű alapot nyújt a tanulóknak a legtöbb szakmához, hogy előnyre tesznek szert a gyakorlati szemléletű oktatással képzett szakemberekkel szemben. Ez talán azt bizonyítja, hogy a programozás oktatása éppen ahhoz a mögöttes tartalomhoz tesz hozzá valamit, ami az *általános programismereti készséget* fejleszti, tehát különösen fontos szerepet kaphatna az informatika tanításában. Ebben a tanulmányban azt a kérdést igyekeztünk megvizsgálni, hogy a programozás tanítása, illetve tanulása miként befolyásolja az informatikával kapcsolatos egyéb képességeket és attitűdöket. Munkánk nem ad egyértelmű igen-nem választ arra, hogy melyik nyelven, hány órát, hány éves korban, milyen tanulóknak, milyen formában kell tanítani a programozást ahhoz, hogy az optimális eredményt elérjük; reményeink szerint azonban megfelelő tudományos alapot tudunk nyújtani az olvasónak ahhoz, hogy ebben a kérdésben meghozza a saját döntését.

## A tervezett mérés – célok, megvalósítás

A következő kérdésekre kerestük a választ: a programozás oktatása milyen hatással van az informatikával kapcsolatos különféle egyéb területekre, van-e gyakorlati haszna; a „programozó diákok” viszonya a számítógéphez hogyan változik; melyik nyelv az, illetve egyáltalán van-e olyan nyelv, amellyel úgy lehet megtanulni progra-

mozni, hogy az egyéb (nem programozási) területekre a lehető legjobb hatást érjük el? Egy adott környezetben milyen programozási nyelveket célszerű tanítani, melyik nyelvvel érdemes kezdeni, illetve azonosítható-e olyan csoport, amelyik jobban jár az-  
zal, ha nem tanul programozást?

Általános célunk tehát egy olyan mérőeszköz elkészítése volt, amellyel meg lehet határozni, hogy egy-egy csoportnak melyik nyelvet, milyen módszerrel és milyen életkorban érdemes tanítani.

## Elvi megfontolások

Egy közel negyvenezer diákot érintő nemzetközi felméréslánc részeként – ahhoz saját vizsgálatokkal kapcsolódva – hazánkban háromezer diákot teszteltünk száz programozással kapcsolatos kérdéssel. A vizsgálatba Magyarország középiskolái közül 40 iskolát vontunk be, amelyeket egyenletes eloszlásban választottunk ki az ország földrajzi régióiból. Ezeknek az iskoláknak minden 17 éves tanulóját teszteltük. Mivel az iskolák jelentős méretű mintát képviselnek (az összes gimnázium kb. 8%-át), a programozók és nem programozók, illetve a különböző informatika-pedagógiai koncepciókkal tanított diákok számaránya a reprezentatív mintán belül megfelel az országos arálynak. Természetesen figyelembe kell venni azt a tényt is, hogy a programozás tanításának ténye nem független az iskola több olyan paraméterétől, amelyek mind befolyásolhatják a különböző informatikai képességek kialakulását és a tanulás körülményeit, illetve hatással lehetnek a programozás oktatásának a formájára. Mivel a programozás témakörének valamely adott iskolában történő tanítása nyilvánvalóan elvileg sem lehet független egyéb tényezőktől, ilyen mérési eljárással nem lehet tisztán meghatározni a programozás tanításának a következményeit. Ennek mérése elvileg nem lehetetlen, a megvalósítás során azonban nyilvánvaló gyakorlati akadályok jelentkeznek, tehát legfeljebb kismintás vizsgálat válna lehetővé, és a tesztelés eredményét is leghamarabb évek múltán várhatnánk.

## Teszt a programozás oktatásának hasznáról

A teszt száz kérdést tartalmaz a programozás tanításának és tanulásának témaköréből. A diákoknak a programozáshoz való viszonyáról és a témakörrel kialakított véleményükről is tettünk fel kérdéseket. A tesztben szereplő összesen 112 kérdést az áttekinthetőség kedvéért hét blokkra osztottuk. Ez nem lényegi, strukturális felosztás; a blokkok határait inkább a feladattípusok eltérései adják. A teszt kérdéseit függelékben közöljük.

### *1. blokk*

Az első blokkban (1–11) azt kérdeztük meg, hogy a diákok szoktak-e – és ha igen, akkor hol, hogyan, és milyen céllal – programozni, algoritmust írni, és saját megítélésük szerint kitől tanultak a legtöbbet.

*Írtam már programot saját használatra, ismerősömmek*

*Az otthoni gépen több különböző programozási nyelven írhatnék programot*

Feltételeztük, hogy a programozáshoz való viszonyt jól jellemzi, hogy az otthoni gépen telepítve vannak-e különféle nyelvek. A pilot vizsgálat azonban kimutatta, hogy az kevésbé jelent bármit is, hogy az otthoni gépen vannak-e vagy nincsenek programozási nyelvek.

### 2. blokk

A második blokkban (12–31) a megkérdezetteknek 13 programozási, illetve „ál-programozási” nyelvről kellett nyilatkozniuk: melyikről hallottak már, s melyiket milyen gyakorisággal használják.

### 3. blokk

A harmadik blokkban (32–46) a diákoknak számszerű adatokkal kellett válaszolniuk a kérdésekre. A programozásra vonatkozóan meg kellett adniuk, hogy mióta, milyen gyakran és legfeljebb mennyi ideig írtak programokat, illetve az iskolában hány héten keresztül tanulták a programozás témakörét. Számszerűen kérdeztük az informatika tárgyból és magatartásból kapott jegyüket, valamint a bizonyítványuk átlagát. Ez a kérdés kicsit meglepő, a korrelációkat vizsgálva azonban belátható, hogy fontos volt feltenni. Ugyanennek a blokknak a második részében az iskolában használt programokat kellett kiválasztaniuk a diákoknak, megnevezve, hogy a felsoroltak közül melyiket használták elsőként, gyakran, illetve a legtöbbit, és melyikkel nem találkoztak addig.

### 4. blokk

A negyedik blokk (47–82) első kérdése a diákoknak a programozás tanulásához fűződő viszonyára vonatkozott. Megkérdeztük azt is, hogy éreztek-e ebben valamilyen változást.

*A programozás tanulása kezdetben elég reménytelennek tűnt, és az idő múlásával ez csak egyre rosszabb lett ... nem, ez nem igaz, nem volt reménytelen*

Ezután a kérdés után a blokk két nagyobb részre osztható. Az első részben a „Miután megírtam életem első néhány programját...” kijelentéssel kezdődő és különféleképpen befejezett mondatokkal kapcsolatban kellett kifejezniük a véleményüket, egyetértésük szintjét pontozással jelezve.

A blokk második nagy egységében egyrészt az informatikához (s ezen belül elsősorban a programozáshoz) való érzelmi viszonyukat kifejező, másrészt nem mindennapi gyakorlati jellegű, konkrét informatikai kijelentésekről kellett véleményt nyilvánítaniuk, s ezek mellé még beillesztettünk 27 egyéb, személyes tulajdonságaikat vizsgáló kérdést. Szerepeltek olyan kérdések is, amelyeknek az elsődleges célja a személyiségjellemzők felmérése volt, s csupán megfogalmazásukban kapcsolódtak a számítástechnikához. A pilot vizsgálat megmutatta, hogy az a vélemény, miszerint azok programoznak jól és könnyedén, akiknek a matematikához több közük van, gyenge lábakon áll.

### *Rajongok a szerepjátékokért (Mágus, ADND,...)*

Az a csoport, melynek tagjai rajonganak a szerepjátékokért, nehezen körülhatárolható. A rajongás annyira erős, hogy nehezen mondható meg, vajon csupán a véletlenül múlik-e, hogy ki szereti meg az ilyen fajta játékokat. A felmérés erre a kérdésre nem ad választ, csupán ennek a csoportnak az informatikai társadalomban való elhelyezkedését vizsgálja.

### *A CD könnyen törik*

Kevesen tudják, hogy ha egy átlagos CD-t annyira meggörbítünk, hogy két

szemközti oldala összeér, az esetek nagy százalékában még mindig lejátszható marad. Hétköznapi körülmények között egyszerűen kizárt, hogy egy CD eltörjön. Ha frizbiként használják és éles, kemény tárgynak ütközik, akkor szét tud robbanni, de a táska alján nehéz könyvek alatt sem törik össze. Az emberek igen nagy része mégis azt gondolja, hogy a CD-vel óvatosan kell bánni. A CD törékenységének érzése feltételezésünk szerint egyfajta távolságtartást, bizalmatlanságot jelent, a gyakorlat hiányára utal, és elárulja azt is, hogy a vizsgált személy kitől tanult a legtöbbet. Az, aki felnőttektől tanulta az informatikát, nem pedig a kortársaitól, nyilvánvalóan egyszer sem fogja hallani, hogy a CD-re egyáltalán nem kell vigyázni.

*A lyukkártyás számítógépek képtelenek voltak az adatfeldolgozásra*

A lyukkártyás gépek természetesen képesek voltak adatfeldolgozásra, sőt eleinte kizárólag arra voltak alkalmasak. Megítélésünk szerint azok válaszolnak inkább nemmel erre a kérdésre, akik az informatikából leginkább az újat, a modernséget értékelik.

*Szerintem a kisebb gyerekeknek nem tesz jót, ha sokat számítógépeznek*

*Szüleim pont fordítva gondolják az előző kérdésre adott választ, mint én*

*Jobbnak tartom, ha egy gyerek a számítógép előtt ül, mint ha a televízió előtt ülne*

Ez utóbbi három kérdés a család hozzáállását vizsgálja. A megfogalmazás lehetővé teszi, hogy a gyermek–szülő viszonyra is következtessünk.

#### 5. blokk

Az ötödik blokkhoz (83–89) négy tisztán logikai kérdés mellett két olyan logikai feladat is tartozik, amelyeknek a megoldásához már informatikai tudás is szükséges. A logikai kérdéseknél a Csapó Benő „Iskolai tudás” című könyvében szereplő teszteket adaptáltuk a témához.

#### 6. blokk

A hatodik blokkban (90–109) eldöntendő kérdésekre kellett választ adniuk a diákoknak. Ezekkel a kérdésekkel azt akartuk megállapítani, hogy milyen viszonyban állnak az informatikai kultúrával, mennyire élnek együtt a számítógéppel, mire emlékeznek a programozás témaköréből és mennyire tartják ezt az eszközt „mindenhatónak”.

*Veszélyes-e belenyúlni egy bekapcsolt számítógépbe?*

*A számítógépes kísérletek világítottak rá, hogy nem létezik örökmozgó?*

*Shoktad állítani a monitor fényességét?*

*El lehet indítani egy számítógépet hajlékonylemezzel?*

#### 7. blokk

A teszt utolsó blokkja (110–112) három kérdésből áll. A legutolsó kérdés talán furcsának és szokatlannak tűnik, de már a pilot vizsgálat is megmutatta, hogy létezik némi gyenge korreláció a programozás és a – ezzel a nem éppen „informatikai” tesztbe illő kérdéssel vizsgált – jelenség között, ami eléggé izgalmas adat ahhoz, hogy érdemes legyen rajta elgondolkodni.

## AZ EREDMÉNYEK ELEMZÉSE

### Alapstatisztikák, összefüggések

A programozásról szóló tesztünk kérdéseire adott válaszok alapstatisztikáinak meghatározásakor az ellentmondásos adatokat korrigáltuk, vagyis például, ha azt állította valaki, hogy még nem írt programot, akkor a programozási szokásokra vonatkozó kérdésekre adott esetleges válaszait töröltük. Azokat, akik írtak már programot, a továbbiakban az egyszerűség kedvéért „programozóknak” vagy „programozó diákoknak” nevezzük. Az eredményeket kérdés-blokkonként közöljük.

#### *1. blokk*

A megkérdezett diákok 40%-a (1206 tanuló) írt már programot, és majdnem minden programozó diák kapott már önálló programozási feladatot. A programozó diákok 60%-a nyilatkozott úgy, hogy írt már algoritmust. Ez a megdöbbentően alacsony arány – az egyéb kérdésekre adott válaszok tükrében – arra enged következtetni, hogy sokan esetleg nem is tudják, mi az algoritmus, vagyis a tanárunk esetleg nem elég sokszor említette ezen a néven azt, amivel foglalkoztak.

A programozó diákoknak több mint a fele nem az iskolában készítette el első algoritmusát. Blokkdiagramot 28%-uk, egyszerű szöveges leírást pedig 30%-uk készített már. 23%-uk vallotta azt, hogy programírás előtt mindig készít algoritmust is, bár az előbbi megjegyzés tükrében ez nem jelent sokat. Igen magas arányban (32%) állították azt, hogy az algoritmus már keveredik a kód szövegével. 40%-uk írt saját magának vagy egyéb célra olyan programot, ami nem házi feladat volt. A programozó diákok 30%-a nem az iskolában írta élete első programját. 37%-uk tudna az otthoni gépen több különböző programozási nyelvet alkalmazni.

A diákok – saját bevallásuk szerint, tanárukat nem számítva – a következő forrásokból tanultak meg programozni: barátaiktól 32%, semmi egyéb forrás felhasználásával 30%, könyvekből 22%, a program helpjéből, autodidakta módon 13%, az Internetről 3%. Akik azt mondták, hogy írtak már algoritmust, nagyobb gyakorisággal tanultak könyvekből programozni, de lehet, hogy ez csak azt jelenti, hogy ők jobban tudják, mi az algoritmus. Aki írt már saját célra programot, az is inkább a könyvet, az Internetet vagy a program helpjét használta fel programozás-tanulási céllal, a többiek (a „nem hobby-programozók”) inkább azt állították, hogy a barátaikon és tanáraikon kívül senkitől sem tanultak. Kiderült, hogy aki csak az iskolában kezd el programozni, az nem fog utánanézni a programozás rejtjelmeinek, és az a legvalószínűbb, hogy tanárán kívül semmi más forrásból nem fog tanulni programozástechnikai fortélyokat.

#### *2. blokk*

Nyelvhasználat, nyelvismeret

*Programozók*

## 1. táblázat

## A diákoknak a különféle nyelvekben való jártassága

<i>Nyelv</i>	<i>Hallottak róla</i>	<i>Használták</i>	<i>Többször is használták</i>
Pascal, Turbo Pascal	93%	74%	40%
Basic	90%	35%	11%
Visual Basic	85%	29%	9%
Java	78%	18%	7%
Delphi	65%	22%	11%
Comenius Logo	60%	31%	11%
C, C++	60%	22%	13%
Visual C	55%	17%	9%
Assembly	53%	14%	8%
Prolog	36%	12%	9%
<i>Állyelvek</i>	<i>Hallottak róla</i>	<i>Használták</i>	<i>Többször is</i>
Macintosh	48%	13%	5%
Visual HTML	40%	15%	8%
Constructor	34%	12%	9%

Jól látszik, hogy a Pascal mennyire uralkodó szerepet játszik, mind az ismertség, mind a használat, mind pedig a rendszeresség terén: a diákok 14%-a kedvencének vallja ezt a nyelvet.

A különböző programozási nyelvek ismeretét természetesen nem ilyen sorrendben kérdeztük meg a diákoktól. A fenti táblázatot itt a „hallottam már róla” oszlop szerint rendeztük. Az eredeti testben már a harmadikként említett „nyelv” (Macintosh) nem valódi; sőt, közismerten nem valódi nyelv volt, mivel arra szeretnénk volna ösztönözni a diákokat, hogy tényleg csak olyan nyelveket jelöljenek meg, amelyeket ismernek. Annak a sejtésnek az alapján tettük ezt, hogy vannak olyan számítógép-használó diákok, akik bizonyos mértékig tájékozottak, ismernek egyes fogalmakat és a számítástechnikával kapcsolatos idegen szavakat (pl. Macintosh), de szeretnék ennél többet mutatni magukból. Ezzel a módszerrel világossá vált, hogy hány százalék az, amit például az Assembly vagy a Prolog esetében biztosan el lehet hinni. A C, ill. C++ sorában a *többször is használták* címszónál látható egy ugrás, s ez a fenti táblázat tükrében már tényleg elhithető.

*Nem programozók*

Érdekes az alábbi adatsor, amely azt mutatja, hogy mennyire ismerik az igazi programnyelveket azok, akik még nem írtak programot: Basic 73%, Pascal, Turbo Pascal 62%, Visual Basic 58%, Java 53%, Comenius Logo 44%, Visual C 31%, Delphi 30%, Prolog 25%, C, C++ 25%, Assembly 23%. Az álnyelvek esetében kapott arányok: Macintosh 37%, Visual HTML 29%, Constructor 28%. A két adatsor összehasonlításából kiderül, hogy az *álnyelvek* a nem programozó diákoknál nem különbülnek el annyira a többtől. A kérdéssor értelmezésében megint az álnyelvek lehetnek segítségünkre. Megfigyelhető, hogy a nem programozó diákokra inkább jellemző a „nagyotmondás”, mint programozó társaikra, mivel a három álnyelv közül majdnem mind-

egyiket nagyobb valószínűséggel ismerték fel. Az alábbi százalékok azt mutatják, hogy azok, akik már írtak programot, milyen arányban kezdték a tanulást az egyes programozási nyelveken: Pascal 48%, Basic 33%, Comenius Logo 9%, Vizuál nyelvek 4,4%, C 3%, Delphi 1,6%, Assembly 0,9%.

### *3. blokk*

#### *Programozó diákok*

Ebből a blokkból kiderül, hogy a programozó diákok 28%-a egy évvel, 16%-a két évvel, 14%-a pedig három évvel korábban írta élete első programját. A megkérdezettek 10%-a a felmérés idején tanult programozni, a fennmaradó 32% pedig még korábban kezdte a tanulást. A diákok 12%-a már több mint 5 évvel korábban megírta élete első programját. A tanulók a felmérés előtt eltelt egy év alatt átlagosan hat programot írtak. 24%-uk egyet sem írt, 50%-uk viszont több mint hetet. 20%-uk több mint 20-at, 10%-uk pedig 40-nél is többet írt. 58%-uk még nem írt olyan programot, amit ne fejezett volna be aznap, amikor elkezdte, és a diákok 90%-a nem írt 20 napnál tovább programot. A tesztet kitöltő diákok fele ötöst szokott kapni informatikából. Ebből a tárgyból kapott osztályzataik átlaga szintén elég magas (4,38), a bizonyítványuk átlaga viszont csak 3,8 volt, vagyis informatikából az átlagnál könnyebben tudtak jegyet szerezni. A diákok magatartását a korreláció-analízis, illetve a faktoranalízis során érdemes figyelembe venni.

#### *A teljes minta adatai*

A megkérdezett diákok átlagosan 20 héten keresztül tanultak programozni, 17%-uk egyáltalán nem tanulta ezt a témakört az iskolában. Érdekes, hogy azok, akik korábban nem tanultak programozást, rosszabb jegyeket szereztek informatikából. Ezt a megfigyelést kétféleképpen lehet értelmezni. Az egyik lehetőség szerint – figyelembe véve, hogy bizonyos gyerekek vagy tanulnak programozni, vagy nem egy osztályban – azt mondhatjuk, hogy azokban az iskolákban, ahol az informatika oktatásához nem tartozik hozzá a programozás témaköre, ebben a tárgybán nehezebb jó jegyet szerezni. A másik értelmezés szerint nem elhanyagolható számban vannak olyan diákok, akik még (vagy már) azt sem tudják, hogy tanultak-e valaha programozni, és ezt figyelembe véve nem is meglepő, hogy az informatika tárgyban szerzett osztályzataik rosszabbak.

Az összes – köztük a nem programozó – diákot számításba véve az informatika jegyek átlaga 3,8, szemben a fent említett, 4 egész feletti átlaggal. A bizonyítványok átlaga 3,67, vagyis, ha csak egy picivel is, de rosszabb eredményt kaptunk, mintha csak azoknak a diákoknak az osztályzatait vettük volna figyelembe, akik tanultak programozni. Ezekre az összefüggésekre nem lehet könnyen magyarázatot adni. Sokféle értelmezés elképzelhető, például az is lehetséges, hogy a tanár azokat tanítja programozni, akik jobbak informatikából, és az is elképzelhető, hogy a programozás tanulása után jobban fog menni nekik az informatika, és egyéb területeken is sikeresebb dolgozatokat fognak írni.

A harmadik blokk második része az iskolában tanult programozási nyelvekről szól. Az iskolában a diákok 12%-a a Comenius Logo, 37%-uk a Pascal, 18%-uk pedig a Basic nyelvel kezdte a programozás tanulását. Ha összehasonlítjuk ezeket az adatokat az előző blokkban feltett kérdésre adott válaszokkal, amelyekből kiderül, hogy melyik nyelvet tanulták először, akkor arra a következtetésre juthatunk, hogy a

Comenius Logo nyelvvel otthon, magánszorgalomból nem próbálkoznak a gyerekek, és a Basic népszerűbb „házi” programozási nyelv, mint a Pascal, annak ellenére, hogy épp az ellenkezőjét váránk. A diákok 16%-a kezdett valamilyen vizuális nyelven tanulni az iskolában, ami messze meghaladja az első nyelvnél megjelölt százalékokat. (Ennek talán az az oka, hogy ezeket a modern vizuális nyelveket nem olyan „egyszerű” beszerezni, mint például a Basic vagy a Turbo Pascal nyelvet.)

Aki írt már programot, az nagyobb valószínűséggel ismeri fel a különböző programozási nyelveket, illetve a Macintosh-t és az egyéb álnyelveket, bár ez a korreláció szerencsére nem szignifikáns.

A korrelációs táblázatból kiolvasható, hogy minél jobban ismeri és minél többször használja is valaki a Pascalt, annál valószínűbb, hogy írt már algoritmust. Az adatokból jól kiolvasható, hogy ezek a diákok először az iskolában írtak algoritmust, s azt minden egyes programozás előtt megírják. Természetesen a Pascalon kívül van még néhány nyelv (Basic, Delphi, Comenius Logo), amelyeknél szintén találunk korrelációt a fenti kijelentésekkel, de ezek nem szignifikánsak, szemben a Pascal 0,2-0,3-0,4 körüli értékeivel. Azok is gyakrabban írnak algoritmust, akik Pascallal kezdték a programozás tanulását. (Hasonló korreláció mutatható ki azzal a kijelentéssel, hogy „az iskolában az első nyelvem a Pascal volt”.) Ugyanezekkel az állításokkal szignifikáns negatív korrelációt találunk azoknak a diákoknak az esetében, akik a programozás tanulását valamelyik vizuális nyelvvel kezdték.

Meg gondolandó persze, hogy megfelelő cél-e egyáltalán algoritmikus elven megközelíteni a modern számítógépeket, nem kellene-e inkább géprendszer-tani és objektum-orientált alapokon gondolkodtatni a gyerekeket. Az *OECD* tesztjében az egyik feladat megoldásakor a diákoknak egy blokkdiagramot kellett kiegészíteniük. A korrelációk alapján azt állapíthatjuk meg, hogy *akik legelőször a Delphi nyelven kezdtek el programozni, azok sokkal kisebb valószínűséggel tudták megoldani a blokkdiagramos problémát, mint azok, akik egyéb nyelveken kezdték a programozást.* Hasonló megállapításra juthatunk, ha a blokkdiagramos feladat megoldása és az egyéb vizuális nyelveken való programozás közötti összefüggést vizsgáljuk meg. Az informatikai tudást is megkívánó logikai feladatokkal szintén pozitív korrelációt találunk. Érdeemes megjegyezni, hogy a kizárólag logikai képességeket vizsgáló tesztkérdéssel viszont nincs szignifikáns korreláció.

#### 4. blokk

Ebben a blokkban a diákoknak a programozáshoz fűződő viszonyát vizsgáltuk. A tanulóknak „*A programozás tanulása kezdetben elég reménytelennek tűnt...*” kezdetű mondatot kellett – öt lehetőség közül választva – kiegészíteniük. Alig 8%-uk vallotta azt, hogy ez a későbbiekben csak rosszabbodott, 20%-uk úgy nyilatkozott, hogy mostanra sem lett jobb a helyzet, 30%-uk számára pedig közömbös volt ez a témakör. A megkérdezettek 15%-át tették ki azok, akik kezdetben tartózkodóak voltak, de később megszerették a programozást. A lelkesek táborához, azokhoz, akik kezdetben sem tartották reménytelennek a dolgot, 27% tartozott.

A blokkok közötti összefüggések

Azzal a kijelentéssel, miszerint „minél többet foglalkozunk informatikával, annál reménytelenebbnek tűnik átlátni a számítógép működését”, inkább azok értettek egyet (0,12), akik nem írtak még programot. Az is érdekes tény, hogy aki írt már prog-

ramot, az jobban szereti a matematikát (0,10). Ennek sok oka lehet, de a legvalószínűbb magyarázat az, hogy főleg azokban az osztályokban tanítanak programozást, ahol a diákok inkább „reál-érdeklődésűek”. Várható volt, hogy aki írt már programot, az nem mondja magáról azt, hogy inkább tévézik, mint számítógépezik (0,15). Lehet, hogy a programozás közelebb hozza a diákokhoz a számítógépet? Azt is meg lehet állapítani, hogy a programozó gyerekek régebben sem voltak rossz viszonyban a számítógépekkel (0,18), persze ez – annak tükrében, hogy sokan vannak, akik az iskolai tanulás előtt már programoztak – esetleg éppen azt jelenti, hogy azok kezdtek el korábban programozni, akik jobb viszonyban voltak a számítógépekkel.

A programozó diákok nagyobb gyakorisággal oldották meg jól a logikai feladatokat (ez bármelyikre igaz a hat ilyen feladat közül), s az iskolában az ő informatikai tudásukat többre értékelték, de nem szabad elfelejtenünk, hogy legtöbbször az egész osztály tanult programozni, és nem túlságosan valószínű, hogy az osztály egésze a diákok egymáshoz viszonyított helyzetéhez képest előrébb került a skálán. Ezek a diákok úgy gondolják, hogy ki tudná használni, ha lenne laptopjuk (0,11), az általuk használt szoftverekben több programhibát találnak, mint társaik (0,23), továbbá többször állítják be a monitor fényességét (0,12). Fontos fájl veszett már el a gépükből (0,19), természetesen szokott náluk kislemez lenni (0,16), sőt – társaikkal ellentétben – számukra nem kérdés, hogy a számítógép elindul kislemezről, és ők azok, akik ismerik a programokban használt standard gyorsbillentyűket is (0,11). Érdekes adat, hogy közülük többen gondoltak már hajfestésre (0,08), és nekik szignifikánsan kevésbé jelent gondot a csirkebél-lapátolás, jó fizetség esetén (0,14).

Ha összefüggést keresünk a különféle programozási nyelvek ismerete és a logikai feladatok (83-88) megoldása között, szintén érdekes megfigyeléseket tehetünk. Minél jobban ismeri egy diák a Delphi nyelvet, annál kevésbé tudja helyesen kitölteni a logikai jellegű kérdéseket tartalmazó tesztet. Bár a korreláció szignifikánsnak nem nevezhető, hat esetből ötször negatív értékű. A Basic esetében pontosan fordított a helyzet: negatív korrelációt egyszer, pozitív korrelációt pedig öt esetben találtunk. A Pascal nyelv használata egy kicsit jobb hatást gyakorolhat a logikai készségekre, mint a Basic nyelv. A tesztünk eredményeként kapott korrelációs táblázatból azt olvashatjuk ki, hogy a Comenius Logo használata csak negatív hatással volt a diákok logikai készségére (hat negatív korreláció).

Ugyanezeknek a logikai feladatoknak és az először használt nyelvnek az összefüggését vizsgálva a következő kijelentéseket tehetjük: a Comenius Logo használata esetén a logikai tesztkérdésekben nem kaptunk sem jobb, sem rosszabb eredményeket, a korrelációk értéke nulla körül volt. A Pascal esetében mind a hat korreláció pozitív volt, s ezek közül egyről állapíthattuk meg, hogy szignifikáns. A Delphi mint első nyelv és a logikai feladatok kapcsolata negatív korrelációkat eredményezett, két esetben szignifikáns jelleggel. A C és a Basic nyelv elsőként való tanulása és a logikai feladatok között nem volt szignifikáns korreláció. Meg kell jegyeznünk, hogy az első nyelvként alkalmazott „egyéb vizuális nyelveknek” a logikai teszt-eredményekkel való korrelációja egy esetben szignifikánsan negatív volt. A korrelációs táblából az is kiderül, hogy minél régebben írt valaki először programot, annál jobban tudta kitölteni a logikai tesztet. Egyik nyelvvel sem találtunk ilyen erős pozitív korrelációt. Mind pozitív, és a hatból kettő gyengén szignifikáns (0,12 ill. 0,18). Az utolsó egy év során írt programok számával is pozitív korrelációt lehetett kimutatni, de az előbbinél még

gyengébbeket. Az is megállapítható, hogy a logikai tesztek kitöltése független attól, hogy hány napon át írta valaki élete leghosszabb programját. A bizonyítvány átlagával, az informatika és a magatartás osztályzatokkal is enyhén pozitív korrelációt fedeztünk fel. Ez a korreláció – az informatika jegyek átlagával való összefüggést vizsgálva – egyedül az informatikai elméleti tudást kívánó logikai kérdéseknél (84, 85) mutat magasabb értéket. A logikai megoldások és az iskolában tanult első nyelv kapcsolatát tekintve szignifikáns korrelációt nem találtunk, annyit viszont meg tudtunk állapítani, hogy a korreláció a Pascal esetében inkább pozitív, a Delphinél pedig inkább negatív előjelű.

Aki írt már programot...

A programozó diák az iskolában is többet számítógépezik a nem programozóknál, sőt a barátainál is, de legfőképpen otthon használja a gépet (0,22). Az iskolában gyakrabban készít web-oldalt, több információt keres és talál az Interneten és többet e-mailezik, valamint többször használ grafikus kalkulátort és szimulációs programot. Ugyanezeket a tevékenységeket az iskolán kívül szintén többször végzi, mint azok, akik még nem programoztak, sőt, a táblázatkezelőt és a szövegszerkesztőt is gyakrabban használja. Ezek közül a diákok közül lényegesen többen telepítettek már programokat, nyúltak bele a számítógépbe, készítettek web-oldalt és használtak adatbázis-kezelő programot. Jártasabbnak érzik magukat a különféle programok használatában (0,24; 0,20; 0,20;) és kiváltképpen a programozásban (0,42).

A fenti adatok után talán meglepően hangzik, hogy ezek a diákok a teszt kitöltését megelőző napon saját bevallásuk szerint a többieknél jóval kevesebbet ültek a számítógép előtt. Az is furcsának tűnhet, hogy az *OECD* tárgyi tudást mérő tesztjén – a főként amerikaiak számára készült kérdésekre válaszolva – közülük kevesebben tudták megmondani, *hogy mi a vírus, a RAM, a CPU vagy a böngésző program és mit jelent a „mega” szó, s a fájlkiterjesztések jelentésével sem voltak annyira tisztában, mint a többiek. A szövegszerkesztésre, táblázatkezelésre és adatbázis-kezelésre vonatkozó kérdésekre – gyengén szignifikánsan – rosszabbul válaszoltak.*

Ez a csoport zártnak tűnik, ám a faktoranalízisben ez sem jelenik meg, s ennek alapján azt lehet sejteni, hogy mégsem a programozás tanítása vagy tanulása a meghatározó tényező.

Aki írt már programot, az 0,34-es korrelációval nem találja reménytelennek a programozás tanulását. Ezek a diákok 0,11-es korrelációval mondták azt, hogy miután tanultak, jobban el tudták képzelni, hogy hogyan működik a számítógép, és rájöttek arra is, hogy milyen bonyolult dolog megírni egy programot (0,30). Néhányan közülük szívesen lennének programozók (0,14). A „megértőbb lettem, ha lefagyott a gépem”, „gyökeresen megváltozott a számítógépekről alkotott véleményem”, „megérttem, hogy sokba kerülnek a jogtiszta programok”, valamint az „én is igyekszem, hogy otthon jogtiszta programjaim legyenek” kijelentésekkel viszont nem lehet szoros összefüggést találni. A programozó diákok többsége nem gondolja úgy, hogy „hasznosabb lett volna az informatika egy másik témakörével foglalkozni” (0,13).

## A tanulók informatikai attitűdjei

### A faktoranalízis eredményei

A 2983 darab 204 részből álló valid teszt faktoranalízise során – miközben tudatában voltunk annak, hogy az egyes faktorok nem feltétlenül jelentenek különálló csoportokat – a sajátértékek szerinti sorrendben (a világosan érthető fogalmazás kedvéért némileg egyszerűsítve a helyzetet) a következő faktorokat azonosítottuk:

#### Az elutasítók

Az első, kiemelkedően erős faktor a *számítógépet nem használók* és a géppel *ellenséges viszonyban levők* köre. Ők saját bevallásuk szerint rossz viszonyban vannak a gépekkel, szeptember óta csak néhányszor használták egyáltalán, és inkább néznének TV-t, mint hogy bármi olyat csináljanak, ami a számítógéphez kapcsolódik. Ahol a mondatokat előre megadott részek összepárosításával kellett megalkotni, ők rakták össze a legtöbbször azt a változatot, hogy „a gépekkel kezdetben rossz viszonyban voltam, és ez a helyzet csak romlani látszik”. A számítógépet *szükségtelennek*, vagy jobb esetben *szórakozási eszköznek* tekintik, és három óra után már könnyezik tőle a szemük. Alig várták a programozás témakörének a végét, alig emlékeznek valamire belőle, a programozás tanulása számukra egészen biztosan vagy valószínűleg szükségtelen volt, és nem gondolják, hogy az hozzátartozna az alpműveltséghez. Ami számukra fontos, azt szerintük az Interneten biztosan nem lehet megtalálni, veszélyesnek tartják a beletyúlást a bekapcsolt számítógépbe, és természetesen sohasem tettek ilyet. Meggyőződésük, hogy a CD könnyen törik. Úgy vélik, a gép nem indul el floppyról, és számukra nem lenne hasznos, ha laptoppal rendelkeznének. Szerintük a gyermekekre káros hatással van a számítógép, és a szülei is így gondolják. Nemcsak az informatika, hanem a magatartásjegyük és a tanulmányi átlaguk is rosszabb az átlagnál.

A többségnél jobban tudják, hogy az örökmozgó lehetetlenségére nem számítógépes kísérletek mutattak rá, de azzal már nincsenek tisztában, hogy nyeresi esélyük a lottón programok segítségével sem nő meg. Nem tudhatjuk, hogy mindez csupán az elutasító attitűd vagy reális megfontolás következménye. Ezek a diákok az informatikai tudástól független logikai kérdésekben is rosszabb eredményt mutattak.

#### *A faktor és a programozás kapcsolata*

Az ebbe a faktorba tartozó válaszadók főként vizuális nyelveken kezdtek el programozni, de többségük egyáltalán *nem írt* algoritmust és programot. Abból kiindulva, hogy ők kirándulni is kevésbé szeretnek, esetleg egy – nem csupán az informatikára vonatkozó – általános negatív attitűdöt is feltételezhetünk.

#### A logikusak

A második, „logikus” faktor főként a *logikai kérdések* megválaszolásának, illetve az általános *teszt-gyakorlottságnak* a faktora. Az ide tartozó diákok olyan kérdésekre is jobban tudtak válaszolni, amelyek csak tárgyi tudás alapján tölthetők ki. Ők tudták a legbiztosabban, hogy a Ctrl+G nem szokott semmit sem jelenteni, nem ismerték a fan-

tom programnyelveket, például a Constructort, a VisualPascalt vagy a Macintosh-t. Nagyobb gyakorisággal programoztak már, és a szemük három óra monitornézés után sem könnyezik.

Ennek ellenére az átlagnál kevesebbet ülnek gép előtt, kevesebb időt töltenek szövegszerkesztéssel, táblázatkezeléssel és egyéb számítógépes tevékenységekkel, viszont otthon az átlagnál többet játszanak. Talán ezért telepítenek gyakrabban programokat. Mindezek alapján azt gondolhatnánk, hogy ezek a tanulók jobban szeretik a matematikát és az informatika jegyük is jobb az átlagnál, ami nem bizonyult igaznak, de a bizonyítványukról már elmondható, hogy jobb az átlagosnál.<sup>1</sup>

### *A faktor és a programozás kapcsolata*

Ezekre a diákokra jellemző, hogy programot is és algoritmust is írtak már, és ezt jellemzően az iskolában tették meg először. Többször kaptak önálló programozási feladatot, és ha nem is kizárólag, de leginkább Pascal nyelven kezdtek el programozni. A faktoranalízis szerint a különféle programozási nyelveket az átlagosnál sokkal kevésbé ismerik, viszont nem dőltek be az „álnyelveknek”.

### **A „nagy mellényű” érdeklődők**

Fő ismertetőjegyük, hogy *alig használnak gépet*, viszont az anonim tesztben azt állították, hogy bizony jártasak nemcsak a valódi, hanem a számunkra is ismeretlen fantom programnyelvekben is. Természetesen írtak már saját használatra programot, és tekintet nélkül arra, hogy létezik-e az a nyelv, különböző környezetekben fejlesztették. A faktornak azonban mégis van értelme, mert bár validitás-mutatóként használtuk, mégis kifejezi az ide tartozó diákok viszonyát a számítástechnikához.

### *A faktor és a programozás kapcsolata*

Erről a faktorról talán elég annyit mondani, hogy ezek a diákok állításuk szerint nagyon jól ismerik a Turbo Pascal nyelvet, de azt nem sikerült eldönteniük, hogy a sorok végén van-e kettőspont.

### **A kihíváskereső, maximalisták**

Érdekes, hogy ez a személyiségjeggy egy elsősorban informatikai jellegű kérdés-sorban ilyen markánsan jelenik meg. Viselői azok, akik nehéznek, de fontosnak és elkerülhetetlennek tartják az informatika tanulását; nem játszanak, hanem szöveget szerkesztenek, táblázatot kezelnek és kommunikálnak. Szerintük az IKT veszélyekkel terhes, a gyerekeknek biztosan árt, a számítógépbe belenyúlni veszélyes; a CD számukra a legtörékenyebb, és a monitort is ők tartják a legnehezebbnek.

<sup>1</sup> Lehet, hogy a középiskolai matematikát – a közfelfogással ellentétben – nem a logikusan gondolkodó emberek szeretik?

Úgy gondolják, hogy az Internet bizonyos mértékig ki fogja szorítani a könyveket. Nem örülnek a programozás tanulásának, amit bonyolultnak, nehéznek és áttekinthetetlennek tartanak. Felnőtt korukban nem szívesen lennének programozók, a programozást szerintük elég lett volna a programozóknak tanítani, mégis azt gondolják, hogy bizony ez is hozzátartozik az alpműveltséghez, és jobb lett volna már korábban tanulni. Ez a már ellentmondásosnak tekinthető véleménycsoport arra utal, hogy szerintük az élet nehéz, a világ tele van küzdelemmel, tanulni és dolgozni kell. Egy kicsit rossz hír ennek a faktornak, hogy ők tudták a legkevésbé, hogy melyik Napóleon vesztette el a waterloo-i csatát. Mindenki másnál jobban félnek a határidőktől, könnyezik a szemük a monitortól, és ők szeretnek a legjobban kirándulni. A jogtisztaságot fontosnak tartják, egyet is értenek vele, de a jogi jellegű kérdésekre már nem biztos, hogy jó választ adnak. Teljesen megértők a géppel szemben, ha az lefagy, nekik egyébként ritkábban fagy le. A magatartásuk lényegesen jobb az átlagnál.

#### *A faktor és a programozás kapcsolata*

Ebben a faktorban az egész mintára jellemző átlagnál többen írtak programot és algoritmust, és ezt jellemzően az iskolában tették először, legnagyobb részben Pascal nyelven, de semmi esetre sem valamilyen vizuális nyelven. Programírás előtt az átlagnál jobban igyekeznek megírni az algoritmust. A felmérés előtti év során is több programot írtak az átlagnál.

### **A hedonisták**

Ehhez a faktorhoz sok jellemző vonás között leginkább az adja meg a kulcsot, hogy ők az Internetet, a számítógépet és az informatikát *szórakozásnak* tartják. Sokat játszanak, és nem zavarja a szemüket a monitor. Az átlagnál rosszabb a bizonyítványuk átlaga és a magatartásjegyük, és nem tudnának nekik annyit fizetni, hogy csirkebelet lapátoljanak. Szívesen ülnek gép elé, és beállítják a monitor fényerejét. Szeretnek szerepjátékot játszani, bulizni, és jó helyre sorolják magukat az osztályban informatikából, annak ellenére, hogy az informatikai tárgyi tudást mérő kérdésekben nem jobbak az átlagnál, és a jegyük sem ezt mutatja. Meglepően sokat csetelnek és e-maileznek otthonról, ami napjainkban egy kicsit a család anyagi helyzetére is enged következtetni.

#### *A faktor és a programozás kapcsolata*

Jellemző erre a faktorra, hogy tagjai nagyon sok nyelvet „ismernek”, de közel sem mindet. Amiről hallottak már (pl. Macintosh), azt „bevállalják”. Nagyon jellemző rájuk, hogy nem írtak sem algoritmust, sem programot, vagy a felmérés előtti évben nagyon kevés programot írtak, ha egyáltalán írtak, és az iskolában is csak nagyon rövid ideig tanultak programozni.

## A „mama kedvence” faktor

Nekik a legjobb a magatartásuk, gépbe még nem nyúltak, nem szeretnek szerepjátékokat játszani, és a szüleikkel teljesen egyetértenek abban, hogy a kisgyerekeket nem szabad számítógéphez ültetni. Nem szeretik a kólát, ellentétben a túrórudival, és csak az iskolában játszanak a géppel, otthon *szöveget szerkesztenek* vagy más *hasznos dolgokat csinálnak*. Kifejezetten jó a bizonyítványuk, és sok segítséget is kapnak mind otthon, mind az iskolában, de magukat nem tartják az átlagnál jobbnak informatikából, annak ellenére, hogy osztályzatuk lényegesen jobb az átlagnál.

### *A faktor és a programozás kapcsolata*

Erre a faktorra nem jellemző a programozó típusú diák, de aki írt már programot, az az iskolában kezdte, és ezt Comenius Logo vagy C nyelven tette. Kevés programot írt a felmérést megelőző egy év alatt, és eszébe sem jut programozónak menni.

## A „játékosak”

A felmérést megelőző napon ők ültek a legtöbbit a számítógép előtt, és játszottak. A programozást nagyon nem szeretik. Hardverépítésben, telepítésben teljesen otthon vannak, és egyáltalán nem tartják reménytelennek, hogy valaha is átlássák az egész rendszer működését. Mind informatika, matematika és magatartás jegyük, mind bizonyítványuk átlaga rossz.

Szívesen befesteneék valami „meredek” színűre a hajukat, kikapcsolódást, szóraozást keresnek a weben, illetve az egész informatikában. Sokat játszanak a barátaiknál is, szeretnek buliba járni, a számítógép-használattal kapcsolatban nem értenek egyet a szüleikkel, és szenvednek a programhibáktól. Nem gondolják, hogy a programozási ismeretekre szükségük volna, nem lennének programozók, már alig emlékeznek valamire a programozás témaköréből, és különben is jobb lett volna szerintük valami mást tanulni helyette. A programozás egész témaköre reménytelen volt és az is maradt számukra, nem tartják jártasnak magukat a programozásban, s véleményük szerint a számítógép a legkevésbé sem munkaeszköz, egyesek szükséges rossznak is mondják. A lyukkártya szerintük – talán éppen ezért – semmire sem volt alkalmas. Nagyon izgalmas eredmény, hogy szerintük számítógépes kísérletek világitottak rá az örökmozgó lehetetlenségére. Ebből messzire vezető gondolatsort lehetne levezetni, hiszen éppen az ehhez a faktornhoz tartozó diákok töltenek nagyon sok időt a virtuális valóságban, de arra gondolva, hogy nekik programok használatával a lottón is megnőne az esélyük, lehet, hogy egyszerűen csak *a számítógépbe vetett hit* a mozgatórugójuk.

### *A faktor és a programozás kapcsolata*

Ez a faktor nincs igazán kapcsolatban a programírással, de tagjai – amennyiben már programoztak – legelőször túlnyomórészt Pascal nyelven írtak programot.

## A Net-nemzedék

Ők kapcsolják a legtöbbször az Internetre a gépet, ők leveleznek, neteznek, csetelnek a legtöbbet. Megtalálják a Neten, amit keresnek, tudják mi a Java, szeretik a matematikát, és általában egyedül számítógépeznek. Saját maguktól tanultak a legtöbbet informatikából, jól válaszolnak a logikai jellegű kérdésekre, jók a jegyeik és a magatartásuk. A túrórudit sokkal jobban szeretik, mint a kólát. Ennek a faktornak a markáns megjelenése egyértelműen mutatja, hogy a *Net-nemzedék* már jelen van, érdekes azonban, hogy sok olyan dologtól nem független, amit nem sejtettünk.

### *A faktor és a programozás kapcsolata*

Az átlagosnál kevesebben tanultak programozni, de ezt Comenius Logo nyelven kezdték, az iskolában. Erre a faktorra egyáltalán nem jellemző az a vélemény, hogy „jobb lett volna mást tanulni programozás helyett”, sem pedig az a kijelentés, hogy „alig vártam a témakör végét”. Nem találtunk kapcsolatot a „reménytelennek tartottam” állítással sem.

## A „csavarhúzó programozók”

Ennek a faktornak a tagjait akkor tudjuk a legkönnyebben elképzelni, ha egy csavarhúzóval felszerelkezett „programozót” próbálunk megjeleníteni magunk előtt. Ők azok, akik a legközelebb állnak a *hardverhez* és a programozáshoz is. Nagy biztonnággal szűrik ki a nem létező nyelveket, és biztosan van náluk éppen most is egy kislemez. Nem nagyon ismerik a programozási nyelveket, de tudják, hogy floppyról is lehet „bútolni”. Szeretik a matematikát, a sportot inkább csak nézik, megírnák akár a lottósegítő programot is, és tudják, hogy az örökmozgó problémáját egyértelműen számítógéppel oldották meg. Beállítják a monitor fényességét is. Ami izgalmas, az az, hogy információkeresésre, e-mailezésre és csetelésre ők használják legkevésbé a gépet; ha valami fontosat keresnek a weben, akkor azt nem találják meg, és általában is távol tartják magukat a html-től. A jogi kérdésekben nem túl jártasak. A kóla magasan vezet náluk a túrórudi és a csoki előtt. A programozást hamarabb és alaposabban szerették volna tanulni, sőt tanulták is. Az ő asztalukon a gép valószínűleg *nyitott dobozzal, gombjaival a fal felé fordítva áll*, hogy jól hozzá lehessen férní a gép „érdemi részéhez”. Mindezekkel együtt nem tartják magukat jártasnak a programozásban.

### *A faktor és a programozás kapcsolata*

Ennek a faktornak a programozással való kapcsolatára nem derült fény. A faktor leg-erősebb jellemzője, hogy tagjai azt állítják magukról, szívesen lennének programozók.

## „Delphi-emberek”

Az tartozik közéjük, aki *oda kattint a jobboldali egér-gombbal, ahova átlagember sohasem fog*. Nagyon érdekes eredmény, hogy ez a faktor megjelenik. Tagjai *grafikus környezetben* tanultak programozni, ezt élvezték is, sokat tanultak és szívesen lennének *progra-*

*mozók.* Az előbbtől két dologban különül el élesen: sokkal inkább általában is a vizuális nyelveket ismerik, és míg az előző csoport az átlagnál lényegesen kevesebbet használja a Netet, ez a csoport lényegesen többet. Talán fontos lenne elgondolkodni azon, hogy miért áll ez ilyen erős kapcsolatban a kezdetben tanult nyelvvel. Esetleg a vizuális nyelvek hatására érzik úgy, hogy fél óránál kevesebb időre nem érdemes leülni a számítógép elé. Azzal is egyetértenek, hogy fontos a jogtisztá programok használata. Érdekes, hogy ez a csoport is a chips-et és a kólát részesíti előnyben az édessegekkel szemben, és különösen érdekes, hogy míg a csavarhúzó programozóknak az átlagosnál jobb, ennek a csoportnak rosszabb a magatartása. Az előző csoport tagjai mindig hordanak magukkal kislemezt, az ebbe a faktorba tartozókra azonban ez nem jellemző. Szintén szeretik a szerepjátékokat, a számítógépezést természetesen jobb dolognak tartják a tévé nézésnél, és boldogok, ha bekapcsolhatják a számítógépet. Érdekes továbbá, hogy a monitort az előbbi csoport az átlagnál gyakrabban, ez a csoport viszont az átlagnál sokkal ritkábban állítja be. Ezt a faktort talán „*hardver-távoli*” programozóknak is nevezhetnénk. Azt sem mellékes megjegyezni, hogy míg egy csavarhúzó programozó nem szívesen másol le senkinek olyan programot, amelyen sokat dolgozott, addig a delphi-ember ezt még szívesen is teszi.

## A „jogtiszták”

Ezt a faktort a jogtisztaság jellemzi. Tagjai szeretik bekapcsolni a számítógépet és azt jobbnak tartják a tévénél. Szerintük a legkevésbé káros a gyerekekre a számítógép, utálnak kirándulni és tévézni. Mindenben a jogtisztaságot támogatják, ugyanakkor – ezzel párhuzamba állítható módon – megértők, ha programhiba miatt lefagy a gépük, és szívesen másolják le az általuk készített programokat bárkinek. A számítógépet szórakozásnak, illetve használati tárgynak tartják. Kiemelkedően sok tapasztalatuk van a könyvtári adatkeresésben.

### *A faktor és a programozás kapcsolata*

A faktor tagjai jellemzően olyan programozók, akik először az iskolában írtak programot, Pascal vagy Delphi nyelven.

## A vizuális kultúra gyermekei

Ez a faktor azt bizonyítja, hogy sokan vannak azok, akiknek a számítógéphez fűződő kapcsolatában fontos szerepet játszik a *vizualitás*. Ők használnak a legtöbbször *grafikai programokat*, szeretnek *web-oldalakat* csinálni, bizonyára ízlésesen megtervezett saját oldaluk is van, és ők ismerik a legjobban a *meppelés* technikáit. Leginkább ők gondolják úgy, hogy a kisgyerekeknek ártalmas a gép, és érdekes, hogy csak az első faktorban szereplő elutasítók használják kevesebbet a gépet otthon, mint ők. Ez a faktor tanúsítja, hogy az Internettel és a számítástechnikával kapcsolatban a legmeglepőbb megközelítésekkel találkozhat az ember. Talán a nagy tárgényű grafikus programok és fájlok használatának tulajdonítható, hogy az összes közül ez a csoport tudta a legjobban, hogy a „mega” szó mit jelent.

### *A faktor és a programozás kapcsolata*

Az első nyelv, amit ezek a diákok tanultak, a Pascal volt, és ezt egészen tűrhető szinten sikerült is elsajátítaniuk, amint azt egy másik kérdés igazolta. Úgy tűnik, hogy a programozás ebben a faktorban nem játszott fontos szerepet.

### **Az „ember- és könyv-közeliek”**

Ez a faktor egyfajta *realista konzervativizmust* mutat. Tagjai hisznek a legkevésbé abban, hogy az Internet kiszorítja a könyveket, és ők értik a legkevésbé a grafikókat, diagramokat és folyamatábrákat. Ők hajlanak a leginkább arra a véleményre, hogy a számítógép szükséges rossz, ám ugyanakkor – szüleikkel egyetértésben – úgy vélik, hogy nem árt a kisgyerekeknek sem. A kólát egyenesen utálják, a csokoládé viszont a kedvencük. Nem szeretnek buliba járni, és nem lennének szívesen programozók.

Mint várható volt, nem hordanak maguknál kislemezt, nem szeretik a szerepjátékokat, és úgy gondolják, hogy a szoftverekben sok a programhiba.

### *A faktor és a programozás kapcsolata*

Érdekes a csoport, mivel fő jellemzője az, hogy tagjai Basic nyelven kezdtek el programozni. A Macintosh „álnyelvet” ugyanúgy programozási nyelvnek tekintik, mint a Comenius Logo, a Constructor, és a Turbo Pascal nyelveket. Nem gondolják azt, hogy a programozás tanulása borzalmas dolog volt, és azt sem, hogy programozást csak a programozóknak kellene tanulniuk.

## **Összegzés**

A programozással kapcsolatos kérdések és a számítógép-használati teszt kérdései között fennálló számos korrelációból többféle következtetést is le tudunk vonni. Egyrészt azt mondhatjuk, hogy akit megtanítanak programozni, az nagyobb kedvvel és lelkesedéssel lát hozzá új eszközök, programok felfedezéséhez, jobban fogja szeretni a számítógépet, és több időt fog eltölteni vele. Másrészt viszont nem szabad elhanyagolni a tanár szerepét sem. Sokszor ez a legmeghatározóbb. Motivációt jelenthet egy lelkes tanár, aki a különféle programok használata mellett nemcsak a kettős számrendszert tanítja, hanem programozást is. A diákok kedvet kapnak a tantárgyhoz, szívesen nyúlhatnak bele a számítógépbe, és lelkesen készítik el saját web-oldalukat. Ezek a diákok jártasabbnak érzik magukat a különféle informatikai műveletek elvégzésében, és még ha ez esetleg nem felel meg a valóságnak, akkor is jobban érzik magukat a számítógép előtt, mert úgy érzik, hogy „jártasak” ebben a világban. Nem titkolható el az a tény, hogy ezeknek a gyerekeknek a tárgyi tudása rosszabb, mint a többieké, viszont ha ennek a tárgyi tudásnak a megszerzése azzal jár, hogy a gyerekek frusztráltak lesznek és a számítógéppel rossz viszonyba kerülnek, akkor lehet, hogy erről a nem közvetlenül szükséges elméleti tudásról le kell mondanunk, és talán hasznosabb, ha gondolkodni tanítjuk diákjainkat.