Digitális kompetenciák a NAT-ban és a XXI. század elvárásai

Árgilán Viktor[1], Kelemen András[1]

[1] {gilan, kelemen}@jgypk.szte.hu
SZTE JGYPK

Absztrakt. Napjainkban széleskörűen elterjedtek a szabadon konfigurálható és programozható eszközök, ugyanakkor a Nemzeti Alaptantervben megfogalmazott digitális kompetenciák közül hiányzik az algoritmikus, a felmerülő probléma megoldására fokuszáló gondolkodás. Ennek következménye, hogy a közoktatásban főleg csak alkalmazói ismereteket tanítanak. A szerzők véleménye szerint a digitális kompetenciák közé mindenképpen fel kell venni az algoritmikus problémamegoldó gondolkodást, és az informatika órákon főként az ilyen gondolkodásra nevelő programozást, valamint az ehhez kapcsolódó ismereteket (architektúrák, operációs rendszerek, hálózatok) kell oktatni. Az általános felhasználói ismeretek oktatását pedig be kell integrálni más tantárgyak keretébe.

Kulcsszavak: digitális kompetenciák, problémamegoldó gondolkodás, algoritmus, programozás,

1. Bevezetés

A XXI. század technológiai fejlődése egyre inkább az általános műveltség részeként megköveteli az algoritmikus gondolkodást, a programozási és szoftverfejlesztési ismereteket. Magyarországon a közoktatást szabályzó törvényekben [1] a digitális kompetenciák hangsúlyosan szerepelnek. Ezek három részre oszthatóak:

- Ismeretek:

  Jogi/etikai szabályok, Információtárolás, kezelés szabályai

- Készségek:

  Szövegszerkesztés, Táblázatkezelés, Adatbázis-kezelés, Internet használat, Információkeresés; gyűjtés; feldolgozás

- Attitűdök:

  Kritikus gondolkodás, Kreativitás, Innováció

A digitális kompetenciák ilyen megfogalmazása miatt a középiskolák döntő többségében az informatika órákon a klasszikus irodai és multimédia szoftvereknek mélyen a részletekbe belemenő oktatása történik. Ez azt a hamis képzetet kelti a diákokban, hogy az informatika csak irodai célokra használható, s lényegében csak az irodai munkások számára fontos.

Ráadásul ma már szinte minden diák rendelkezik okostelefonnal, aktívan használja a közösségi hálózatokat, de az informatika órákon ezen eszközökről, valamint a biztonsági kérdésekről nem esik szó. Meg kell jegyezni, hogy néhány oktatási intézményben az irodai programcsomagok mellett szoftverkészítési és fejlesztési metódusokat is tanítanak, melyek az algoritmizálás témakörben szerepelnek, de oktatásuk elavult, manapság már nem használt technológiákra (Pascal) és elvekre (algoritmus programfejlesztés nélkül) korlátozódik. Hiányzik az alapvető adatábrázolási struktúrák (láncolt listák, fák, gráfok, stb.) használata, valamint az objektum orientált szemlélet. A programozási ismereteken kívül hiányoznak azon ismeretek oktatása, melyekkel a diák képes operációs rendszert telepíteni, beállítani. Biztonságosan otthoni hálózatot kiépíteni, üzemeltetni.

Tapasztalataink alapján elmondható, hogy a középiskolai tanulmányok befejezése után az informatika-specifikus felsőoktatási szakokra felvételt nyerő diákok számára a programozási kurzusok teljesítése gyakran nagyon nehéz, olykor lehetetlen feladatot jelent. A fenti alapvető problémakör ismertetése és elemzése alapján az előadásban bemutatunk egy koncepciót arra vonatkozólag, hogy miként lehetséges a kor célkitűzéseinek megfelelően oktatni a közoktatásban az informatikát.

2. Koncepció

A koncepció fő célkitűzésként megfogalmazható célja, hogy a jövő informatikai szakemberei mellett a mindennapi felhasználó is elsajátíthassa azon ismereteket, melyek a jövő szakmái esetében várhatólag szükségesek lesznek. Bár a dinamikusan változó gazdasági és társadalmi környezetben jelenleg nagyon nehezen meghatározhatóak a következő évtizedek munkaerőpiacán használható kompetenciák, azonban az infokommunikációs technológiák térhódítása egyértelművé teszi, hogy az átlagfelhasználó által szabadon konfigurálható és programozható eszközök használata elengedhetetlen lesz. Ezen eszközök használatához szükséges kompetencia azonban nincs benne az elvárt digitális kompetenciák között. Hiányzik a problémamegoldó gondolkodásra való nevelés. Ennek egyik következménye, hogy a PISA teszteken Magyarország most már rendszeresen az OECD átlag alatt teljesít. Találkoztunk már olyan feladattal, amelynél egy tetszőeges alakzat területének minél pontosabb becslése volt a cél egy adott terület egység felhasználásával. Ezzel szemben a közoktatásban a szabályos alakzatok (kör, paralelogramma, háromszögek, stb) területének a kiszámítására helyezik a hangsúlyt, s nem viszik tovább a gondolatmenetet, pedig a feladat megoldása meglehetősen egyszerű ötlettel megvalósítható: helyezzük az alakzatot az egységnyi területekből alkotott négyzethálóra és számoljuk össze az alakzaton belül levő egységnégyzeteket. Azoknak a területeknek a becslésére, amelyek az alakzaton belül vannak, de kisebbek, mint az egységnégyzet könnyen átlátható módszer adható. A feladat megoldása rávilágít arra, hogy ennél a problémánál az integrálszámítás ismerete nélkül elfogadható pontosságú becslést tudtunk adni az alakzat területére. Az informatika oktatásában ugyan ez a helyzet. Az alkalmazói ismeretek sulykolásával az informatikai problémamegoldó gondolkodás és képesség kerül háttérbe.

A koncepció kiinduló eleme az, hogy az általános felhasználói ismereteket (szövegszerkesztés, táblázatkezelés, prezentáció, képszerkesztés, stb.) integráltan más tantárgyak keretében lenne kívánatos megvalósítani. Ez persze feltételezi, hogy a nem informatika szakos tanárok is magabiztosan használják a különböző IKT eszközöket, ezek az eszközök szükség esetén rendelkezésre álljanak. Nyilvánvalóan lesznek olyan tanárok, akik többlet teherként fogják ezt érezni, azonban a tantervek korszerűsítésével, az óraszámok változtatásával és a tanárok megfelelő képzésével ezek a problémák áthidalhatóak. Természetesen a felhasználói ismeretek ilyen integrált oktatása esetén, már el lehet kerülni azt a – véleményünk szerint káros – gyakorlatot, hogy egy diák akkor jó informatikából, ha mélyen ismeri néhány szoftver használatát. Ugyanis pl. nyelvtan órán szövegalkotás nyelvtani és stilisztikai szabályainak gyakorlásakor a gyakorlást össze lehet kötni szövegformázási feladatokkal. Hasonlóképpen a képszerkesztési ismereteket a rajz és vizuális nevelés órákon is el lehet sajátítani. A felsorolást nyilvánvalóan lehet még tovább folytatni.

De akkor mit tanítsunk az informatika órán? A rendelkezésre álló időkereteken belül: hardver ismeretek, operációs rendszerek telepítése, beállítása, file és könyvtár műveletek, hálózatok működése, biztonsága, programozás. A következőkben felsorolásszerűen áttekintjük, hogy az egyes címszavaknál mit tartanánk fontos ismeretnek.

Hardver ismeretek [2]

Számítógépek általános felépítése (processzor, memória, beviteli-kiviteli eszközök, háttértárolók). Számrendszerek, logikai műveletek, kódtáblák.

Operációs rendszer [3]

Operációs rendszer telepítése tiszta gépre. Meghajtó programok telepítése, hálózati kapcsolatok beállítása, Felhasználói fiókok beállítása. Vírusvédelem, karbantartás.

Fájl és könyvtár műveletek [4]

File és könyvtár fogalma. Nevezési konvenciók, Fájlok és könyvtárak létrehozása, törlése, másolása mozgatása konzol parancsokkal és grafikus felületről is. Fájl és könyvtárműveleteket támogató szoftverek.

Hálózatok működése, biztonsága [5]

Számítógép hálózatok fogalma. Rétegzett felépítés (OSI modell, négyrétegű TCP/IP modell), IP cím fogalma, útvonalválasztás. Kliens-szerver architektúra, hálózati protokollok (http, ftp). Hálózati kommunikáció biztonsága, Gyakran használt hálózati szolgáltatások (Web, levelezés, közösségi hálózatok)

Programozás [6, 7]

Az adat és az algoritmus fogalma. Hogyan lesz az adatból és az algoritmusból program: programozási nyelvek. Adat és kódábrázolás a számítógépen: Neumann elv.

Az oktatásra kiválasztott programozási nyelven: egyszerű feladatok megoldása, összetettebb feladatok megoldása részfeladatokra történő bontással. Függvények, eljárások készítése. Alapvető algoritmusok (keresés, rendezés, rekurzió, file műveletek). Dinamikus memóriakezelés. Magas szintű adatábrázolási technikák: bináris fák, láncolt listák. Gráfok ábrázolása, gráf algoritmusok: bejárás, feszítő fa, stb.

Objektum orientált programozás: osztály és objektum, példányosítás (konstruktor, destruktor), adattagok, metódusok, láthatóság, származtatás, absztrakt osztályok.

Adatbázis fogalma, relációs adatbázis, az SQL alapjai

Vizuális programozás: Felhasználói interfész tervezése, kódolása. Felhasználói események kezelése (szövegmezők, nyomógombok, szelekciós vezérlők). Kapcsolódás relációs adatbázishoz (ODBC).

Webes alkalmazás fejlesztés alapjai: kliens oldali és szerver oldali programozás alapjai.

3. Oktatási módszer

Mint a fenti koncepcióból látszik a cél az, hogy aki informatika-specifikus felsőoktatási szakra jelentkezik az, az informatika órákon megszerezze azokat a kompetenciákat, melyek szükségesek az ilyen szakok sikeres elvégzéséhez. Természetesen azok a diákok, akik nem informatikából szeretnének továbbtanulni, azok úgy érezhetik, hogy ez számukra nehéz és elvont és sok. Igazuk van. Számukra az IKT eszközök működési logikájának a megértése és ez által ezen eszközök biztonságos használata a fontos. Azonban a működési logika megértése programozási ismeretek nélkül – véleményünk szerint – nagyon nehézkes. Gondolhatunk itt arra, hogy, pl. az Excel függvényeinek használata a jelenlegi általános tantervű osztályokban is kötelező tananyag. Azonban oktatásuk és megtanulásuk a legnehezebben tanítható és elsajátítható ismeretek közé tartozik pont a megfelelő programozási ismeretek hiánya miatt. Így ezen ismeretek gyökértelenül maradva hamar a feledés homályába kerülnek. Ugyanakkor manapság már az adminisztratív munkakörökben is elvárt az Excel függvényeinek készségszintű használata. Nem beszélve a VBA nyújtotta lehetőségekről. Felvetődik a kérdés, hogy hogyan motiváljuk a diákokat arra, hogy programozással foglalkozzanak. Első lépésként mindenképpen az algoritmikus gondolkodást fejlesztő játékos feladatokkal kell kezdeni. Ilyen feladatok pl:

- Igazmondók-hazudósok szigete,
- Farkas-kecske-káposzta,
- Kannibálok és misszionáriusok.

A sort lehet még folytatni. Fontosnak tartjuk, hogy a feladatok megoldásánál mindig algoritmust adjunk, s térjünk ki minden lehetséges esetre. Véleményünk szerint a programozás oktatásakor a kezdő lépések megtanulása után már ne öncélú feladatokat oldogassunk meg, hanem már a kezdet kezdetén mutassuk be, hogy a félév - vagy esetleg az év- végére a tanórák során milyen szoftvert fogunk elkészíteni. Így a diák számára a cél ismeretében világossá válik, hogy mit miért csinálunk. Természetesen a nagyon motiváló tud lenni, ha a megvalósítandó szoftver látványos, szórakoztató és viszonylag egyszerű adatszerkezetekkel és algoritmusokkal megvalósítható. Ilyen szoftverek pl. szókitaláló (akasztófa játék) program, torpedó játék, tic-tac-toe, egyszerűbb kártyajátékok. Magasabb szinten pl. chat szoftver, egyszerű rajzoló program. Fontosnak tartjuk még, hogy

- a programozás tanulása során a diákok két, háromfős kiscsoportokban dolgozzanak,
- minden segédeszközt szabadon használhassanak,
- a számonkérésnél a szomszéd kivételével is szabadon lehet használni minden segédeszközt. Nyilvánvalóan a számonkérés során itt nem a lexikális ismereteket kell bevasalni, hanem a programozási készségek fejlődését. Szélsőséges esetben ez persze egyéni feladatokat jelent.

Javasoljuk, hogy középiskolában programozási nyelvként a napjainkban mind a felsőoktatásban, mind az iparban széleskörűen elterjedt nyelvek (C/C++, C#, Java, a webes technológiák estén PHP) válasszunk. Természetesen az adott osztály érdeklődési körének függvényében a tanár szabadon súlyozhat. Általános iskolában programozásra a LOGO és a SCRATCH kiváló eszköz.

Irodalom

1. *Nemzeti Alaptanterv 2016* (2007) http://www.nefmi.gov.hu/letolt/kozokt/nat_070926.pdf
2. Andrew S. Tanenbaum: Számítógép-architektúrák - 2. átdolgozott, bővített kiadás, Panem kft., Budepest, (2006)
3. Andrew S. Tanenbaum, Albert S. Woodhull: Operációs rendszerek - Tervezés és implementáció CD melléklet, Panem kft., Budepest (2007)
4. James F. Kurose, Keith W. Ross: Számítógép-hálózatok működése - Alkalmazásorientált megközelítés, Panem kft., Budepest (2009)
5. Benkő Tiborné, Tóth Bertalan: Együtt könnyebb a programozás - Java - CD melléklettel, COMPUTERBOOKS, Budapest (2007)
6. Benkő Tiborné, Dr. Poppe András: Együtt könnyebb a programozás - C - CD melléklettel, COMPUTERBOOKS, Budapest (2009)