

Oktatási környezetek vizsgálata a programozás tanításához

Horváth Győző¹, Menyhárt László Gábor²

¹gyozke@inf.elte.hu

ELTE IK

²menyhart@inf.elte.hu

ELTE IK

Absztrakt. Az informatika tárgy területe a többi tantárgyhoz képest viszonylag gyorsan változik, így az informatika oktatásának céljait, eszközeit, módszereit időről időre felül kell vizsgálni annak érdekében, hogy a szükséges ismereteket a tanulók sikeresen megszerezzék. Cikkünket informatikaoktatással foglalkozó kutatóknak és olyan informatika tanároknak ajánljuk, akik szeretnének újszerű oktatási környezetekkel foglalkozni és ezek segítségével megszerettetni tanulóikkal az informatikát, azon belül is a programozásoktatást. Írásunkban először áttekintjük az sikeres ismeretátadás komponenseit és a programozásoktatás hagyományos formáit és eszközeit. Ezt követően összeszedjük a ma tanulókorban lévő új generáció igényeit és ezekre válaszként született egy-két megoldási lehetőséget. Az újszerű oktatási környezetek közül mutatunk be párat, végül elemezzük őket.

1. Bevezető gondolatok

Egy adott ismeret sikeres elsajátításának folyamatában a következő komponensek különíthetők el:

- az elsajátítandó **tananyag**;
- a tanulás folyamatának szereplői: az **oktató** és a **tanuló** személy;
- azok az **eszközök**, amelyek a tanulási folyamathoz elengedhetetlenül szükségesek, illetve azok, amelyek segítik ezt;
- az ismeretátadás módja, az oktatás **módszere**;
- valamint az oktatás célja, egy adott célterület **kompetenciái**.

Ezek a komponensek természetesen nem függetlenek egymástól, hiszen az elsajátítandó ismeret megkövetelheti bizonyos eszközök meglétét és rendelkezésre állását, a résztvevők jelenléte vagy annak hiánya, valamint előképzettségük vagy valamilyen szempontú tulajdonságaik (korosztály, tanulási nehézség stb.) az ismeretátadás módszereit befolyásolhatja. Ráadásul a komponensek az időben változhatnak, ami az egész ismeretsajátítási folyamatra kihatással lehet, hatékonyabbá vagy éppenséggel kevésbé hatékonyá teheti azt. Az egyes komponensek ugyanakkor nem egyszerre változnak. Az oktatás során elérendő kompetenciák viszonylag időtől függetlenül adottak, ezek kialakítása az oktatás célja. Az eszközök azonban gyakran változhatnak, és azt is tapasztaljuk, hogy nem kell sok időnek eltelnie, hogy a résztvevők minősége is megváltozzon. A módszereknek pedig jobb esetben követnie kell ezeket a változásokat.

Az informatika tantárgy ilyen szempontból nincsen szerencsés helyzetben. Bár az elérendő kompetenciákat megvizsgálva elmondhatjuk, hogy azok ennél a tárgynál is időtállóak, az informatika területének gyors változása azonban megköveteli a tantárgyhoz kapcsolódó ismeretek oktatásának is viszonylag sűrű felülvizsgálását. A következőkben röviden megnézzük, hogy a tanulási folyamat egyes komponenseiről mit mondhatunk az informatikában.

A kompetenciák. Az informatikában a következő kompetenciákat különböztetjük meg [1, 2]:

- Algoritmikus gondolkodás
- Adatmodellezés
- A valós világ modellezése
- Problémamegoldás
- Kommunikációs képesség
- Alkalmazói képesség
- Csoportmunka, együttműködő-képesség
- Alkotó képesség
- Információs tájékozódási és tájékoztatási képesség
- Rendszerszintű gondolkodás

A listát áttekintve láthatjuk, hogy ezek a képességek, illetve ezek kialakításának szükségessége kevésbé van kitéve az idők és a többi komponens változásainak. Ezek kialakítása olyan cél, amely eszköztől, konkrét tananyagtól függetlenül is elérendő. Másik irányból viszont egyfajta kontrollt ad a többi komponensnek, viszonylagos stabilitást kölcsönözve nekik.

A tananyag. Más tantárgyakhoz képest az informatika területe nagyon gyorsan változik (összevetve például a matematika, fizika és a többi tárgy törzsanyagával akár a felsőoktatásban is), és az ott elért újdonságok nagyon hamar beszivárognak és megjelennek egyrészt a hétköznapi élet számos területén (munka, szórakozás, életvitel), másrészt – sokszor az előbbi miatt – az oktatásban is. Gondolhatunk itt az internet megjelenésére vagy éppen a mobil eszközök robbanásszerű terjedésére, ami új lehetőségeket és kihívásokat jelent nemcsak az oktatásban, hanem társadalmi szinten is. A változások e komponens tekintetében 5-10 évben mérhető.

Az oktató. Az oktatási folyamat vezetője is sokféle formában jelenhet meg. A hagyományos kontaktórák mellett egyre gyakrabban jelennek meg a távoktatásos vagy kombinált tanulási formák (blended-learning). Ismereteket manapság már nemcsak tanórán vagy könyvekből, hanem az Internetről igényesebb blogbejegyzéseken, leírásokon vagy videókon keresztül is el lehet sajátítani.

A tanuló. A tanuló az, akinek az érdekében az összes többi komponens működik. Bár humán szereplőről van szó, azt tapasztalhatjuk, hogy az oktatás ezen szereplői is dinamikusan változhatnak [3], viszonylag rövid időn belül is. Az információszerezés és -feldolgozás módja ugyanis megváltoztatja a gondolkodási struktúrákat. Nem véletlen, hogy Y- és Z-generációkról beszélnek, akik már teljesen másképp dolgozzák fel az információkat, mint az előző generáció.

Az eszközök. Az oktatáshoz használt segédeszközök és az oktatás céleszközei is nagyon gyorsan változnak. Ez az a terület, amelyen a legszembetűnőbben tetten érhető az informatika gyors változása. Segédeszközök tekintetében nem is olyan régen még a tábla volt a kizárólagos lehetőség, azóta viszonylagos gyors ütemben jelentek meg a projektorok, aktív táblák, érintőkép-

ernyős és mobileszközök. Ezek használata sokszor technikai nehézséget okoz a tőlük idegenkedőknek, és az oktatásmódszertanba is viszonylag lassan szivárognak be. Ugyanakkor a tanulók szívesen használják ezeket. A céleszközök tekintetében is hasonló ütemet láthatunk, gondoljunk itt az egyes operációs rendszerek, alkalmazói rendszerek vagy programozási nyelvek fejlődési ütemére és változatosságára.

A módszerek. Ez a komponens a többi komponens mozgatórugója, általa juttatjuk el a tanulókat az elsajátítandó ismeretekig. Az oktatási módszerek elvileg nagyon érzékenyek a többi komponens változásaira, hiszen megváltozott körülmények között az ismeretátadás módjának is változnia kell, hogy hatékonyságát megőrizze. És változnak is, hiszen a pedagógia jól érzékeli ezeket a változásokat, és próbálja az átadás módját is ezeknek megfelelően módosítani. Így a hagyományos frontális oktatás mellett már számos alternatíva létezik, mint például a kooperatív tanulás. A gyakorlat azonban azt mutatja, hogy hagyományos tantermi keretek között az újabb oktatási módszerek nehezen találnak utat. Érdekes viszont, hogy a nem tantermi oktatási formák milyen jól igazodnak a célközönség igényeihez (ld. távoktatás, lépésről lépésre történő leírások, közösségi problémamegoldás (StackOverflow), MOOC, stb) [7].

Bár a változásoknak az informatika minden részterülete erőteljesen ki van téve, a következőkben elsősorban a programozásoktatásra és azon belül is a felhasznált eszközökre fókuszálunk. Először megnézzük azt, hogy hagyományosan milyen formában és milyen eszközökkel valósul meg az oktatás, majd felvázoljuk a tanulók újszerű igényeit, és azt, hogy ennek milyen hatása lehet a programozás tanításában. Ezt követően egy konkrét részterület, a JavaScript nyelv kapcsán nézzük meg, hogy milyen lehetőségek vannak. Végül vizsgálatainkat a cikk végén elemezzük és foglaljuk össze.

2. Hagományos formák és eszközök a programozásoktatásban

Ha a programozásoktatásra gondolunk, akkor nagyon gyakran elsőre kódsorok és fejlesztői környezetek jutnak eszünkbe, holott tudjuk, hogy a programozásoktatás ennél sokkal több. Először is a fent említett kompetenciák közül többfélélt is fejleszt, mint például az algoritmikus-gondolkodást, adatmodellézést, valós világ modellezését, rendszerszemléletet. Másodsor tantermi körülmények között a szinte kötelező számítógép és fejlesztői környezet mellett megjelenik hagyományosan a tábla vagy a projektor is. Az ismereteket elsősorban a tanár nyújtja, mellette további fő információforrás régebben a szakkönyv volt, újabban az internetes dokumentációk, szakmai fórumok veszik át ezek helyét. A programozásoktatás tantermi körülmények között sokszor frontális módszerrel történik, a tanár elmondja az ismereteket, amelyek alapján a tanulók önálló vagy közösen vezetett formában oldanak meg feladatokat.

Az alkotó munka nagyobbik része a fejlesztői környezetben történik, így ennek megfelelő kiválasztása elengedhetetlenül fontos. Heizlerné és Szlávi foglalta össze legutóbb, hogy milyen szempontokat érdemes figyelembe venni egy programozási nyelvénél [4] és egy fejlesztői környezet kiválasztása kapcsán [5]. Az alábbi szempontokat sorolták fel egy fejlesztői környezet értékeléséhez:

- áttekinthetőség és kezelhetőség tekintetében illeszkedjék a célkorosztály kívánalmaihoz
- több programozási nyelvtámogatása
- több alkalmazástípus támogatása
- ne igényeljen drága hardvert

- stabilan fusson
- több operációs rendszer alatt is elérhető legyen
- ne kelljen további szoftvereket telepíteni
- forráskód szerkesztéséhez kapcsolatos statikus támogatás (kódkiemelés, kódkiegészítés, kódrészletek, kódformázás, stb.)
- futás közbeni dinamikus támogatás, nyomkövetési, hibakeresési szolgáltatások

A programozásoktatás kezdetén a fejlesztői környezet és az oktatási környezet nagyjából egybeesett. A tanítási folyamatban kizárólag a tanáron múlt, hogyan tudja a fejlesztői környezetet oktatási környezetként felhasználni. A 80-as évek végétől egészen a 2000-es évek elejéig a modern professzionális fejlesztői eszközök és az oktatási eszközök ugyanazok voltak. Ugyanazt a TURBO Pascal szerkesztőt használta egy végzett programozó, mint amit egy programozással ismerkedő kezdő tanuló. Ez a filozófia akkor is megmaradt, amikor később megjelentek a komolyabb tudású, grafikus fejlesztői környezetek. Ezzel a megközelítéssel csupán annyi gond lehet az oktatás során, hogy ezeket az eszközöket, fejlesztői környezeteket azzal a céllal alakították ki, hogy a fejlesztési folyamatokat támogassák, azaz kifejezetten professzionális programozók igényei szerint vannak kialakítva, nem pedig oktatási szempontból. Mivel azonban az oktatás úgyis a tanár kezében van, így rajta áll vagy bukik, hogy ezek a fejlesztői eszközök mennyire válnak oktatási környezetté. Ehhez persze sokszor egyéb segédeszközökre van szükség. A fent vázolt jelenség azért nem probléma sokszor, mert általában a fejlesztői környezetek által nyújtott szolgáltatásoknak csak egy részhalmazával kell megismerkedni az oktatás során.

Érdekes, hogy ez a jelenség azoknál a nyelveknél és fejlesztői környezeteknél jelentkezik, amelyek professzionális felhasználást szem előtt tartva készültek. Az oktatási célú nyelveknél és környezeteknél ezek a problémák nem jelentkeznek. A főleg az általános iskolai korosztályt célzó bevezető nyelveknél, mint például a LOGO vagy a Scratch, mind a nyelv, mind a felület a korosztály igényeinek, és az elérendő célnak, az algoritmikus gondolkodás fejlesztésének van alávetve. Játékos formában, rajzos felületen keresztül keltik fel a korosztály érdeklődését, és így hozzák közelebb az elsajátítandó ismereteket.

A „komolyabb” nyelvek és koncepciók bevezetésekor miért nem alkalmazzuk ezt a filozófiát? Miért nem haladunk lépésről lépésre? Miért nem használunk olyan eszközöket, amelyek támogatják az oktatási folyamatot? Sokszor éppen az eszközök hiányoznak, hiszen sokkal egyszerűbb egy általános célú fejlesztői környezetet használni, ahol csak pár absztrakciós szintet kell hirtelen átugrani, mint egy specifikus eszközt kifejleszteni. Érdekesképpén jegyezhetjük meg, hogy valamelyik amerikai egyetemen a kezdő programozásoktatás Scratchben valósul meg. De több ilyen vizuális, blokkok helyére húzásával történő programozási felület is van, ami a Google Blockly fejlesztését használja alapul, mely háttérben XML.

3. Újszerű igények és módszerek

A bevezetésben említettük, hogy az informatikatanítást nemcsak az nehezíti, hogy maga az anyag és az eszközök gyorsan változnak, hanem az is, hogy a részt vevő diákok igényei is, motíváltsága is más. Ez természetesen nemcsak az informatikát érinti, hanem minden más tárgyat is, azonban az informatikában szembetűnő, hogy milyen informatikai eszközzel tudunk a megváltozott igényeknek megfelelni. A következőkben sorra vesszünk pár, a fiatalok által igényelt új, informatika-oktatási szempontot.

Játékosság. Mindenki szeret játszani, és ezt az igényt az oktatásban is kihasználhatjuk. Kétféleképpen is megtehető ez. Az egyik lehetőség, hogy a tanulókat érdeklő játék fejlesztése kapcsán ismerkednek meg a programozással és a kapcsolódó fogalmakkal. Ők maguk is érdekeltté válnak tehát az alkotási folyamatban, látják, hogyan fejlődik az alkalmazás, jó esetben azt is látják, hogy saját ötleteikkel hogyan lehet bővíteni azt, azaz a tanórán kívüli tanulásra is rávehetjük őket, végül pedig olyan végeredményt hozhatnak létre, amelyet szívesen osztanak meg másokkal.

A másik lehetőség pedig az, hogy az ismeretszerzési folyamatot játékos köntösbe bújtatjuk. Most már külön tudományterület foglalkozik azzal, hogyan lehet a játékokban használt motiváló eszközöket tetszőleges oktatási folyamat részévé tenni [6]. Az ötlet alapját az képezi, hogy a játékok általában úgy vannak felépítve, ne rettentsek el a velük játszókat. Ehhez több módszert is alkalmaznak. Folyamatosan építik fel a könnyebb pályáktól a nehezebbekig a játék ívét, a pozitív dolgokat, mint például gyorsaság, ügyesség, plusz pontokkal, kitüntetésekkel jutalmazzák, ha pedig valami balul sült el, akkor újra lehet kezdeni büntetés nélkül.

Ugyancsak a játékosság része az egészséges versenyzés, a legjobbak közé való bekerülés, minél több pontszám gyűjtése. Ez játékos formában megint csak kevésbé zavaró, sőt a legtöbb esetben motiváló lehet az adott játékkal vagy témával történő foglalkozásra.

Webes alkalmazások. Az internetpenetráció növekedtével egyre többféle alkalmazás érhető el a weben. Gyakorlatilag már nincs olyan alkalmazási terület, amelynek ne lenne webes vagy legalábbis böngészőbe épülő megfelelője. A mai fiatalok életéhez kapcsolódó tevékenységek szinte kizárólag webes környezetben valósulnak meg: kapcsolattartás (e-mail, szöveges, hangalapú és videóbeszélgetések), közösségi szolgáltatások (Facebook, Google+), játékok (minijátékok, de egyre inkább a nagyobb grafikus és számításigényesebb játékok is), információ-szerzés (keresők, wikik), ügyintézés (internetbank, szolgáltatók), stb.

Közösségi lét. Bár a kutatók nem tudják a következményeit egyértelműen meghatározni, mindenesetre kétségtelen tény, hogy a mai fiataloknak fontos a különböző közösségi oldalakon való jelenlét és kapcsolattartás. Élményeiket, gondolataikat ezen keresztül osztják meg egymással, események, csoportok szervezését ezeken a portálokon keresztül oldják meg, sokszor segítséget is itt kérnek.

Online lét. Ugyancsak az Y- és a Z-generációra jellemző a folyamatos online jelenlét. Erre ma már széles körben adott a lehetőség, hiszen valamilyen formában kézre esik egy számítógép (asztali vagy hordozható formában), és az internetre való csatlakozás költségei sem nagyok (wifi vagy mobilhálózati adatkapcsolaton keresztül). Jelen lenni, azonnal értesíteni és azonnal reagálni, kapcsolatban lenni, ez jellemzi a fiatalok egy részét.

Csapatmunka. Fontos elvárás kompetencia a csoportban történő munka, az együttműködésre való képesség kialakítása is. A tanulók nem feltétlenül igénylik a közös munkát, de a közösség iránti igényükre építve megtapasztalhatják a csoportban rejlő erőt. A pedagógia nagy lehetőséget lát a kooperatív tanulásban, amely megfelelő alternatívája lehet a kizárólagos frontális oktatásnak, és amely adekvát módon válaszolhat a megváltozott oktatási körülmények bizonyos kihívásaira is, új eszközöket adva a tanárok kezébe és új módon segítve a kompetenciák kialakulását.

Alkotás. A tanulók sokkal motiváltabbak, ha ők maguk is érdekeltek az elsajátítandó ismeretek megszerzésében, ha ennek közvetlen hasznosulását látják. A feladat-orientált megközelítés

eleve praktikummal tölti meg a tananyagot, de még erősebb késztetést érezhetnek a tanulók, ha az elkészítendő mű egyedi, akár a saját ötletük alapján.

Felfedezés. Ha érdekes a feladat, akkor szívesen foglalkoznak tovább a környezetet felfedezve, akár önálló módon utánajárva a szükséges ismereteknek. Az információért való sikeres küzdelem további motiváltságot adhat.

Siker-orientáltság. Az emberek szívesebben foglalkoznak olyan dolgokkal, amelyekben sikereket érnek el, mint olyanokkal, ahol folyton saját korlátaikkal találkoznak, és kudarcot szenvednek. Ha a befektetett energia jutalommal jár, akkor később is szívesebben fektet be az ember energiát a siker érdekében.

Ha összevetjük ezeket az igényeket vagy jellemzőket a hagyományos oktatási környezet által biztosított „szolgáltatásokkal”, akkor láthatjuk, hogy a hagyományos kontaktórák csak kis részben teszik a tanulókat motiválttá.

4. Néhány újszerű oktatási környezet áttekintése

Az interneten keresztül azonban már megjelent pár kezdeményezés, amely megpróbál az újfajta kihívásoknak eleget tenni, és újfajta módszerekkel átadni az ismereteket a diákoknak. Az alábbiakban áttekintünk pár ilyen portált, kifejezetten arra fókuszálva, hogy milyen módon próbálják a tanulókat érdekeltté tenni a tanulásban. A legtöbb portál ingyenes lehetőséget ad a tananyagok megtekintésére, így bátorítunk minden olvasót, hogy saját szemével is győződjön meg ezen portálok funkcionalitásáról. Gyűjtésünket elsősorban a programozásoktatásra fókuszáltuk, de még ezek között is vannak olyan oldalak, amelyek más tárgyakhoz tartozó ismereteket is tartalmaznak.

Khan Academy (<https://www.khanacademy.org/>). A sort nyomban egy olyan portállal kezdjük, amely nagyon sokféle tárgyhoz tartalmaz ingyen elérhető tartalmat, angol nyelven. Általában ez abban merül ki, hogy érdekes magyarázó videókat lehet megtekinteni, majd pedig ahhoz kapcsolódó tesztekkel lehet megoldani. Ezek váltakozásával mélyülnek el a tanulók az adott tárgy anyagában. A tantárgyak közül kiemelkedik a programozásoktatási rész (<https://www.khanacademy.org/computing/computer-programming>), ugyanis itt a videók mellett helyben megoldható feladatok is vannak. Egy webes rajzolókönyvtárra épülve grafikus elemek programozásán keresztül ismertetik meg a gyerekekkel a különböző programozási alapfogalmakat és koncepciókat. A kurzus végére egy játék megírása a cél. A programozásoktatás a web natív nyelven, JavaScriptben történik. Minden részfeladat megfelelően ellenőrzésre kerül, így lehet a következő feladatra lépni. Másik figyelemre méltó jellegzetessége az, hogy lehetőséget ad tanulóként, szülőként vagy tanárként megtekinteni az oldalt, így például képet kapva a gyermek haladásáról.

Code Avengers. (<http://www.codeavengers.com/>) A játékos formába öltöztetett programozástanítás egyik eklatáns példája. Alapvetően fiatalokat megszólító felületkialakításával, játékos formában játék készítésével, a feladat egyszerű lépésekre bontásával, pontszámok és jelvények megszerzésével próbálja célját elérni. A nyelv ebben az esetben is a JavaScript, az elérendő cél nagyon egyszerű HTML és algoritmikus szerkezetek implementálása.

CodeHunt. (<https://www.codehunt.com/>) Ez az oldal a C# és a Java nyelv oktatását tűzte ki játékos formában. Felületkialakítása és a megoldandó feladatok is a nagyobb korosztályt célozzák meg. Ezen az oldalon ugyanúgy lépésekre bontott feladatmegoldás található. Érdekessége,

hogy közlésezi az ellenőrizendő teszteseteket, és látható, hogy azok közül, melyik teljesül, melyik nem.

Codecademy. (<http://www.codecademy.com/>) Egy sokoldalú és jól felépített oldal, sokféle ingyenesen elérhető tananyaggal. Az ide látogató tanulók a webes nyelvek és szabványok világával ismerkedhetnek meg, többek között HTML, CSS, PHP, Ruby, Python és JavaScript tananyagok is vannak. A tananyag itt is lépésekre van bontva, minden lépésnél egy rövid kis ismertető található az adott lépés újdonságairól, majd egy rövid feladatleírás. A tanulónak a weboldalon megjelenő kódszerkesztőben kell a helyes megoldást megadnia. Ezt sikeresen elvégezve ugorhat a következő lépésre. Kiemelendő jellegzetessége az adott tananyagnak megfelelő élő kép, ami programozási nyelveknél tipikusan a konzolablak kimenetét mutatja, webes technológiáknál viszont a legenerált HTML oldalt.

Udacity. (<https://www.udacity.com/>) Míg az előző példák a teljesen önálló tanulásra lettek kialakítva, és alapvetően nem túl magas szintet határoztak meg végcélként, az Udacity egy platformot ad a közösségi programozástanuláshoz. Célként az ipari elvárásokat tűzi ki, konkrét alkalmazások elkészítése a cél. Ehhez a tanulókat csoportokba szervezik, és kijelölnek számukra egy iparban tapasztalatot szerzett vezetőt is. Az önálló tanulás mellett tehát itt kiemelt fontosságú a napi kapcsolattartás is. Ezek mellett a (sokszor fizetős) projektek mellett azonban vannak olyan tananyagok is, amelyek önálló tanulásra adnak lehetőséget a fenti példáknak megfelelően (lépésekre bontott feladatmegoldás oktatóvideó és ellenőrzött helyi programkészítéssel, pl. JavaScriptben).

Kódolás órája. (<http://csedweek.org>) Végezetül egy olyan portált szeretnénk bemutatni, amely a programozás népszerűsítését tűzte ki célul. Ehhez olyan, online elérhető oldalakat gyűjtötték össze, amelyeken keresztül játékos formában lehet a programozás alapjaival megismerkedni. A gyűjteményben többek között a fenti oldalak is megtalálhatóak, de számos egyéb oldalra is van hivatkozás (<http://code.org>, <http://codehs.com>, stb).

A fentebb ismertetett oldalak közös jellemzője, hogy üzenetük célba viteléhez megpróbálják figyelembe venni a célcsoport igényeit. Ha az újszerű igények fejezetben felsorolt jellegzetességek mentén elemezzük a fenti oldalakat, akkor láthatjuk, hogy ezeknek nagy részét próbálják figyelembe venni. Az oldalak többsége játékos formában vezeti végig a tanulót: a tananyag egyszerű lépésekre bontott, két lépés között minimális az újdonság. Bármelyik lépés akárhányszor újravégezhető, sokszor a lépések között is szabad az ugrálás előre és hátra, meghagyva a szabadságot az anyag feldolgozásához. Visszajelzés tekintetében a legegyszerűbb megoldás az, ha az anyag feldolgozottságának százalékát tünteti fel az oldal, de vannak olyan oldalak, amelyek pontszámokkal, gyűjthető kitüntetésekkel teszik azt még játékosabbá.

Mindegyik oldal korszerű webes környezetben kerül kialakításra, így nincs szükség további fejlesztői környezetek telepítésére. Az oldalak kialakítása a célkorosztálynak megfelelő, a fiatalabbak megcélzóak kellően színesek, erőteljesen vezetettek és maga a tananyag is általában valamiféle játék vagy játékos formába öltöztetett feladat, a nagyobb korosztálynál pedig ergonomikusan és modern design szerint kialakított felületen történik a tananyag feldolgozása. Az oldalak bármikor elérhetőek, nincsen időben korlátozva a tanulás. A feladatok érdekesek, az apró lépések és a mellé adott részletes leírások miatt szinte garantált a lépésenkénti siker.

Csapatmunkával csak pár professzionálisabb oldal foglalkozik, és érdekes az is, hogy a közösségi oldalakkal való integráció nem jelenik meg ezeken az oldalakon erőteljesebben.

5. A felmerült módszertani kérdések

Milyen tapasztalatokat vonhatunk le ezen felületek alapján a programozás tanítására? Először is azt kell látnunk, hogy a fenti oldalak nagy többsége azzal a feltételezéssel él, hogy a tanuló teljesen önállóan vesz részt a tanulási folyamatban a világ bármely részéről. Ennek megfelelően van az anyag lépésekre bontva, és ezért van a legtöbb lépésnél igen részletesen leírva az adott lépéshez tartozó új ismeret és maga a feladat is. A kód ellenőrzésekor kapott hibaüzeneteknek is kellő részletességgel kell megjelennie, hiszen annak értelmezése is a tanulóra van bízva. Másodsor, a legtöbb oldal azt feltételezi, hogy a tanulási folyamatban a tanuló egyedül vesz részt. Általában nincs lehetőség sem a csoportmunkára, sem az aktuális állapot közzétételére vagy segítségkérésre.

Egy kontaktórán megvalósuló oktatásban a tanár jelen van vezető és segítő szerepben is. Egyrészt a feladatmegoldáshoz szükséges ismereteket az egyszerű szöveges leírásnál változatosabb formában képes nyújtani (akár újabb informatikai eszközökkel), másrészt segítőként is aktívan lehet rá támaszkodni. Továbbá számos módszertani lehetősége van az anyag feldolgoztatásához, szemben a fenti oldalak viszonylag lineáris, egyszereplős megoldásával szemben.

Kétségtelen előny lehet egy online, webes környezetben dolgozni. Ez egyrészt korszerű, „modern”, másrészt lehetőséget teremt a közösségi oldalakkal való integrációra, a megosztásra, másrészt az azonnali kapcsolatteremtésre. Ez utóbbiban akár online felületen keresztül lehet csoportokat létrehozni, és azokat működtetni. Végül a web mindenkor elérhetősege biztosíthatja azt, hogy munkáját bármikor újra előveheti, folytathatja, anélkül, hogy különböző környezetek között kellene váltogatnia. A webes környezet hátránya lehet, hogy kevésbé szofisztikált fejlesztői eszközök vannak egy böngészőben, mint egy asztali alkalmazásban. Hiányozhatnak bizonyos kódszerkesztési funkcionálisok, ezeknek testre szabása, másrészt a böngészőn belüli nyomkövetési és hibakeresési stratégiák is különböznek a hagyományos fejlesztői környezetekben megismertektől.

A programozás alkotó folyamat, amelyben nagyobb önállóságot kaphatnak a tanulók. A fenti oldalak legnagyobb tanulsága, hogy a tanulási folyamat játékossá tehető, és így a tanulók motiváltsága fenntartható. Ehhez persze megfelelő környezet és megfelelően előkészített anyagok kellene, aminek előállítása persze sokszor idő és pénz kérdése is.

Az alkotó folyamat másik jellegzetessége, hogy olyan témákkal foglalkozzanak a tanulók, amelynek létrehozásában örömet szerezhetnek. A játékok programozása ilyen lehet. Megfelelő absztrakciós réteg előkészítésével és bevezetésével látványos alkotásokat lehet készíteni egyszerű ismeretekkel is. Ebben például a webes technológiák megint csak sokat segíthetnek.

Végül pedig bizonyos témaköröknél érdemes a frontális óravezetéstől elszakadva a diákoknak nagyobb kezdeményezést adni. Ez kiterjedhet az óra egy részére kooperatív tanulás keretében, de akár külön projekteket indíthatunk.

6. Összefoglalás

Az informatika és ezen belül is a programozásoktatás világa folyamatos változásnak van kitéve, így időről időre felül kell vizsgálnunk az oktatási folyamatokat. Ebben a cikkben áttekintettük,

hogyan az ismeretátadási folyamat egyes komponensei (az oktató tananyag, a tanuló és a tanár mint résztvevő, a használt és célzott eszközök, az oktatási módszerek és az eléendő kompetenciák) mennyire kitettek a változásoknak. Áttekintettük azt is, hogy a tanulók részéről a megváltozott technikai és szociális környezet milyen új igényeket szül, és ezek hogyan jelentkezhetnek az oktatási folyamatban. A programozásoktatásban eszközszinten számos új kezdeményezés látott napvilágot a világhálón, ezek közül tekintettünk át párat azzal a céllal, hogy egyrészt ezek mennyire felelnek meg az újszerű tanulói igényeknek, másrészt pedig milyen ötletekkel szolgálhatnak a hagyományosabb keretben felépülő tanórák számára. A sokféle jellegzetesség közül kiemelkedő volt a tanulási folyamat játékosá tétele a webes alapú megközelítés, amely bárhol, bárkikor és bárki által elérhetővé teheti az anyagokat, és utat nyithat különböző közösségi vagy csoportos megoldások felé. Megállapítottuk, hogy az oktatási környezet kialakítása elsősorban a tanár kezében volt és van, de megfelelő oktatási módszereket választva ezen új eszközök segítségét jelenthetnek a programozás tanításának tanulóhoz való közelebb vitelében.

Irodalom

1. Péter Szlávi, László Zsakó: Key Concepts in Informatics: Algorithm, *Acta Didactica Napocensia*, ISSN 2065-1430, Volume 7, Number 1, 2014
2. Gyöző Horváth, Péter Szlávi, László Zsakó: Informatics competences, ICAI2010
3. Sabine Graf: AN APPROACH FOR DYNAMIC STUDENT MODELLING OF LEARNING STYLES, *IADIS International Conference on Cognition and Exploratory Learning in Digital Age (CELDA 2009)*
4. Szlávi Péter, Heizlerné Bakonyi Viktória: Evaluation of programming languages in education, *ICAI (International Conference of Applied Informatics) 2014*. Eger
5. Heizlerné B Viktória, Szlávi Péter: Evaluation of development environments from the viewpoint of education, *ICAI (International Conference of Applied Informatics) 2014*. Eger
6. Sebastian Deterding, Dan Dixon, Rilla Khaled, and Lennart Nacke: From game design elements to gamefulness: defining "gamification". In *Proceedings of the 15th International Academic MindTrek Conference: Envisioning Future Media Environments* (MindTrek '11). 2011. ACM, New York, NY, USA.
7. Kovács Miklós: Tömeges, nyitott online kurzusok (MOOC) módszertani vizsgálata, *VI. Oktatás-Informatikai Konferencia*, Budapest, 2014. február 7-8.