

SZAKDOLGOZAT

Podlovics Imre

Debrecen

2008

Debreceni Egyetem
Informatikai Kar

ANGOL NYELVTANI OKTATÓPROGRAM KÉSZÍTÉSE

Témavezető:

Dr. Rutkovszky Edéné

Egyetemi tanársegéd

Készítette:

Podlovics Imre

Programtervező matematikus

Debrecen

2008

Tartalomjegyzék.....	3
Bevezetés.....	4
A programozott oktatásról.....	5
Programtípusok.....	7
A programozási típusok szerinti szekvenciák egyes sajátosságai.....	11
Számítógépes oktatási programok tervezése.....	13
Oktatási programok készítésének szempontjai.....	14
Számítógépek az oktatásban.....	17
Az angol igeidők.....	18
Simple present.....	22
Present Continuous.....	24
Present Perfect.....	26
Present Perfect Continuous.....	27
Simple Past.....	28
Past Continuous.....	29
Past Perfect.....	30
Past Perfect Continuous.....	31
Simple Future.....	32
Future Continuous.....	33
Future Perfect.....	34
Future Perfect Continuous.....	35
A program megvalósítása.....	36
A program elkészítésekor jelentkező sajátosságok.....	37
A program működése.....	39
Programozói dokumentáció.....	40
További lehetőségek a program hasznosítására.....	42
Összefoglalás.....	44
Felhasznált irodalom.....	46

Bevezetés

Szakedolgozatom témájaként egy angol nyelvtani oktatóprogram készítését választottam. Ezt több dolog is indokolta esetemben. Eredeti képesítesem szerint tanár vagyok, ami miatt mindenképpen szerettem volna olyan témát választani, ahol előzetes ismereteimet hasznosítani tudom. A másik oldalt tekintve a programtervező matematikus szakhoz nagyon jól illeszkedik egy program tervezése és elkészítése.

Napjainkban a számítógépet, mint általánosan használt eszközt, az élet szinte egyik területén sem lehet megkerülni. Beszélhetünk a legbonyolultabb számításokat igénylő pl. aerodinamikai fejlesztésekről vagy akár csak egy egyszerű bevásárlásról. A számítógépek mindenhol jelen vannak, nem kivétel ez alól az iskolapad sem. Nagyon sok munkát és időt lehet megtakarítani a tanítás – tanulás folyamatban előre elkészített programok használatával. Ezeket a programokat bárhol és bármikor elő lehet venni, nem igényelnek speciális feltételeket, magát a számítógépet leszámítva. Napjainkban azonban már maguk a számítógépek sem kötelezően az asztali gépeket, ún. desktop-okat jelentik, hanem egyre többször már az olcsó és egyszerű, de oktatási célokra tökéletesen megfelelő hordozható számítógépeket, laptop-okat is ide sorolhatjuk. Ha ezt figyelembe vesszük, kijelenthetjük, hogy használati szempontból az oktatóprogramoknak nincsenek jelentős korlátai. Rohanó világunkban pontosan erre van szükség, és ez teszi az oktatóprogramokat oly jól használhatóvá.

A programozott oktatásról

A tanítási-tanulási folyamat egy bonyolult rendszer, melynek működését és eredményességét igen sok tényező befolyásolja összetett módon. Új eszközök alkalmazása magával hozza a tanulás és tanítás módszereinek, a tanítás technológiájának állandó változását is.

A számítógép oktatási eszközként való felhasználása nagy változást jelent a tanításban és tanulásban egyaránt. A tanulót és a tanárt felmenti mechanikus tevékenységek elvégzése alól, és olyan információforrásként szolgálhat, amely a tárolt ismereteket rendezett és rendszerezett formában nyújtja. Az új eszközrendszer lehetővé teszi a tanítási-tanulási folyamat jobb szervezését, ellenőrzését, gyakorlófeladatok generálását, és a folyamat - legalábbis részleges - irányítását.

A számítógép oktatástechnikai alkalmazására 20 - 25 évvel ezelőtt kezdődtek a kísérletek. A számítástechnika rohamos fejlődése lehetővé tette, hogy mind kisebb méretű, ugyanakkor egyre olcsóbb berendezéseket alkalmazzanak, amelyek kapacitása azonban rendszerint sokszorosán felülmúlja a konkrét oktatási igényt. Döntő változást jelentett a mikroprocesszorok elterjedése, ami a következő időszakban egyre nagyobb szerepet kap az általános iskolai oktatásban is.

A számítógépes oktatási programok történelmi előzménye a programozott oktatás.

A Pedagógiai Lexikon a következőképpen definiálja a programozott oktatást:

"Programozott oktatás: az oktatásnak az a formája, amely lehetővé teszi, hogy a tanuló a tanulásra előkészített tananyagból a program segítségével, közvetlen tanári irányítás nélkül egyéni munkával sajátítson el ismereteket, illetve szerezzen jártasságot, készséget az ismeretekre épülő feladatok megoldásához. A programozott oktatás tehát feltételezi egy oktatási eszköznek, a programnak az alkalmazását..."

A hagyományos tankönyvtől főleg abban különbözik, hogy nem csupán közli a tananyagot, hanem lépésről lépésre megtervezi az anyag feldolgozásának menetét, így lényegében a tanulás módszerét is. A program, mint az egyéni tanulás munkaeszköze, általában a következő jellemzőkkel rendelkezik:

- kis egységekre (lépésekre, szekvenciákra) tagolja a tanulmányi anyagot
- valamennyi egységben kiváltja a tanuló aktív közreműködését (a tanuló kérdésekre válaszol, kijelölt feladatokat old meg)
- lehetővé teszi, hogy a tanuló lépésről lépésre ön-ellenőrzéssel győződjön meg válasza, feladatmegoldása helyességéről.

A program készítésénél figyelembe kell venni, hogy milyen előzetes ismeretekkel rendelkeznek a tanulók a tanulás meg-kezdésekor. Ügyelni kell arra, hogy az egymást követő lépések se túl nehezek, se túl könnyűek ne legyenek egyetlen tanuló számára sem, de mindenki egyéni ütemében haladhasson a végső cél eléréséig.

A tanulás programozása abban különbözik a hagyományos oktatástól, hogy a pedagógus a tanulóval egy eszköz segítségével érintkezik. A kimondott szó és az azt kísérő sokféle inger helyett (mimika, taglejtések, az ezekben kifejeződő érzelmek, értékelés, stb.) az *írott szó* útján létesít kapcsolatot tanár és tanuló között.

Programtípusok

A programozott oktatás módszere esetében a tanítás-tanulás irányításának eszköze a **program**. Annak ellenére, hogy jelen pillanatban rengeteg program létezik, ezek lényegében két fő típusra korlátozódnak:

a/ olyan programok, amelyekben a tanuló maga fogalmazza meg a választ;

b/ feleletválasztásos programok

Az első a B. F. Skinner - féle, a második pedig a N. Crowder - féle programozást, felfogást jelenti.

a/ Önállóan fogalmazott válasz programja (lineáris programok)

A programozott tanulás első formáját B. F. Skinner dolgozta ki pszichológiai alapon (1954). Szerinte "Tanítani valójában nem egyéb, mint a megerősítés olyan körülményeinek a megszervezése, amelyben a tanulók tanulni fognak." A programozás és a programozott oktatás a megerősítés feltételeinek meghatározott célok elérése érdekében való megszervezését jelenti.

A programozott oktatás Skinner felfogásán alapuló elvei:

a/ *a kis lépések elve*: az anyagot kis egységekre (lépésekre) bontják, hogy ez által helyes válasz formájában minél több megerősítés váljon lehetővé.

b/ *az aktív részvétel elve*: a tanulónak minden információs egységgel foglalkoznia kell. Ebből a célból minden egyes esetben egy - egy gyakorlatot kell elvégeznie.

c/ *a siker elve*: az információkat és a feladatot úgy kell kidolgozni és fokozni, hogy a siker biztosítható és a hiba elkerülhető legyen.

d/ *az azonnali ellenőrzés*: a tanulónak minden művelet után meg kell ismernie az eredményt és a megelégedés érzését, ami úgy érhető el, hogy a tanuló lehetőséget kap saját válaszána a helyes válasszal való összehasonlítására.

e/ *a fokozatos előrehaladás*: a nehézségek egymásutánját úgy kell megállapítani, hogy a tanulónak lehetősége legyen az egyszerűtől a bonyolult, a könnyűtől a nehéz felé haladni.

f/ *egyéni ritmus*: a tanulónak saját ritmusában kell dolgoznia, időbeni korlátozás nélkül. Ezzel biztosítható a tanulás egyéni jellege.

Skinner módszerét azért nevezik lineárisnak, mert a tananyagot folyamatosan és egymásra építve fejti ki, adja át a tanulóknak. Minden tanuló azonos utat követ kitérő nélkül, különbség csupán a program lebonyolításának ritmusában jelentkezik. A lépések sorrendjét nem a tanuló, hanem a programot készítő programozó határozza meg.

Az önállóan megfogalmazott választ magában foglaló programtípus elemeinek sorrendje sematikusán így ábrázolható:

- a tanuló informálása ("A" szekvencia)



- megoldandó feladata (kérdés, gyakorlat, stb.)



- a válasz megfogalmazása (megalkotása)



- ennek összehasonlítása a helyes válasszal az eredmény megismerése céljából



- "B" szekvencia

A lineáris programok összeállításánál a következő szabályokat kell követni:

- minden választ azonnal meg kell erősíteni; ez a pszichológiából ismert "megismerési elmélet" reflexe. Úgy vélik, hogy a megerősítés olyan jutalom, amely egyben ösztönzést is jelent a tevékenység folytatására.

- a hibalehetőséget csökkenteni kell, mert ez negatívan befolyásolja a tanulási folyamatot; a hibák számának lecsökkentése érdekében kis lépésekben kell előrehaladni, ugyanakkor ösztönzést kell nyújtania a tanulónak, amely a helyes válaszok megalkotásához vezet el

b/ Feleletválasztásos programok (elágazásos programok)

Ezekben a programokban az információval, és az illető feladattal egy időben több válasz is szerepel (rendszerint 3, 4) melyek közül a tanuló az általa helyesnek vélt feleletet választhatja ki. A helyes válaszon kívül a többit úgy szerkesztik meg, hogy azok fedjék a leggyakoribb hibákat. A választott felelettől függően a tanuló megkeresi a program megjelölt részét és ez vagy megerősíti a válasz helyességét, vagy kiegészítő információt nyújt a téves felelet okáról. Helyes válasz esetén a tanuló tovább olvassa a soron következő anyagrészt, helytelen feleletnél pedig utasítást kap a meg nem értett részlet ismételt áttanulmányozására, vagy egyéb kiegészítő információk átvétele után folytathatja tovább a program feldolgozását.

Az elágazásos programok, a lineáris programokkal ellentétben a tanulók által elkövetett hibákra építenek. A tanuló hibás feleletválasztása az alapja a különböző mellékutaknak. A mellékút legegyszerűbb esetben a hibára reagáló kiegészítő információt nyújtó programlépés, amelynek feldolgozása után a tanuló visszatér az alaplépésre.

Vázlatosan ez a programtípus így néz ki:

a tanuló informálása (A szekvencia)



feladat (kérdés, gyakorlás)



a válasz kiválasztása



Hibás válasz



Kiegészítő információ a hiba kijavítása



Helyes válasz



A következő információ (B szekvencia)

céljából



(A szekvencia)

Az ilyen típusú program szerint összeállított tankönyvekben a válaszokhoz mutatót csatoltak, amely jelzi, hogy az illető válasz kiválasztása esetén a könyv melyik oldalán kell keresni a folytatást.

A programozási típusok szerinti szekvenciák egyes sajátosságai

A/ A lineáris programok szekvenciái rendszerint három elemet foglalnak magukban; az információt, a válasz kérését és megerősítést.

1. Az információ prezentálása

- a/ A szekvencia keretében nyújtott információnak nem kell kiadósnak lennie. Egy szekvencia nem kell, hogy több legyen, mint egy esetleg megismételt szabály
- b/ Az információkat pontosan és érthetően kell szerkeszteni. Általában az információt a lehető legszűkebbre kell korlátozni, a tanuló számára megszokott példákat kell alkalmazni.
- c/ A szekvenciának elégséges elemet kell tartalmaznia a tanuló számára az információ megértésére. Ebből a célból grafikonokat, sémákat és ábrákat kell magába foglalnia.

2. A válasz megkövetelése

A megkövetelt válasz rendszerint egy szó, amelyet a tanulónak az erre a célra fenntartott helyre kell beírnia. Fontos, hogy csupán EGY szót, vagy kifejezést kérjünk és ne többet.

3. A válasz megerősítése

Amint láttuk, a lineáris programban a válasz megerősítése a tanulási folyamat szerves részét képezi. Növeli a motivációt, és a program lebonyolítása során a tanulóra serkentőleg hat.

A válasz megerősítését nagyon érthetően kell megfogalmazni. Itt azonban olyan helyzetek is adódhatnak, amikor a tanuló nem a programozó által kigondolt szóval, hanem szinonimával válaszol. Ebben az esetben a megerősítésnek nemcsak az eredetileg kigondolt szót, hanem a lehetséges szinonimákat is tartalmaznia kell.

B/ Az elágazásos program szekvenciáinak két kategóriáját különböztethetjük meg.

1. főszekvenciák, amelyeken akkor halad át a tanuló, ha válaszai minden esetben helyesek

2. korrekciós szekvenciák, amelyeken hibás válasz esetén megy végig a tanuló.

A főszekvenciában ugyanazok az elemek találhatók, mint a lineáris program szekvenciáiban, de más megvalósításban és elrendezésben. Az első helyen az előző szekvenciában választott *válasz megerősítése* áll, a másodikon az *új információ* a harmadikon a megoldandó *feladat* (kérdés).

A korrekciós szekvenciákban az első helyen az előző szekvenciában választott válasz cáfolata szerepel. Ezt követik a tanuló által elkövetett hiba kijavítására szükséges magyarázatok, és néha, a főszekvenciához képest, eggyel több információ. A szekvencia a főszekvenciához, vagy a korrekciót szolgáló szekvenciához való visszairányítással zárul.

A tanulás folyamatát vizsgálva és a magatartásra utalva Skinner megállapítja: "A nem ismeréstől a megismeréshez vezető, bármely területen végbemenő egész folyamatot nagyszámú kis lépésre kell felosztani, ezek bármelyikének a megerősítését pedig az illető lépés elsajátítása felé kell irányítani." Majd így folytatja: "Miután a lépéseket igen kicsikre méreteztük, a megerősítések gyakoriságát a maximumra növelhetjük; ilyen módon a minimumra csökkenthetjük az esetleges hibák esetleges negatív következményeit."

Egy anyag elsajátítását akkor valósíthatjuk meg maximális hatékonysággal, ha ebben a folyamatban a tanuló olyan kis lépésekkel halad előre, hogy az el nem sajátítás lehetősége minden egyes lépésnél rendkívül csekély legyen.

Az egyik kísérlet szerint egy matematikai programban a szekvenciák számát variálták, és megállapították, hogy a hibák száma annál alacsonyabb, minél jobban csökken a lépések nagysága (minél több szekvencia van a programban). Ezzel egy időben nő az ismeretek alapossága is.

II.Számítógépes oktatási programok tervezése

A számítógépes oktatási programok módszeres tervezésének történelmi előzménye a programozott oktatás. Mint ahogy az már a korábban leírt definícióból kiderült, a programozott oktatás feltételezi egy oktatási eszköznek, a programnak az alkalmazását. A program közvetítésének eszközei közül korábban leginkább a programozott tankönyv és az oktatógép terjedt el.

Programozott tankönyvek és oktatógépek

A programozott oktatás nemzetközi térnyerése nyomán hazánkban a 60-as években indultak meg a vizsgálatok. Magyarországon programozott tananyagokkal az első nagyobb szervezett kísérletet az 1966/67-es tanévben bonyolították le. A kísérlet eredményei alapján készültek el 1969-ben az első programozott tankönyvek 3000 példányban. Majd megjelentek a Pressey által készített mechanikus oktatógépek, melyeket a mai gépek "őseinek" tekinthetünk.

Az oktatógépekkel korábban megvalósított funkciókat napjainkra átvették a számítógépek. A személyi számítógép - ha oktatási célra használjuk - minőségileg más, mint a Pressey-gép volt. Kis túlzással azt is mondhatjuk, hogy annyira más, amekkora különbség egy Forma-1-es versenyautó és egy ekhós szekér között van, habár mind a kettő közlekedési eszköz.

Oktatási programok készítésének szempontjai

A számítógépes oktatóprogram-tervezés speciális jellemzőinek elemzésekor célszerű S. Mayer-Markle: A programozók tizenöt parancsolatában megfogalmazott szempontokat sorra venni.

1. Minden tételt világosan és hibátlan nyelven kell megfogalmazni
2. A tételeknek tartalmilag helyesnek kell lenniük, tévedéseket tehát nem tartalmazhatnak.
3. A tételek sorrendjét logikusan, kapcsolataik figyelembevételével kell megállapítani.
4. A tanulótól megkövetelt feleletből tükröződjön a tétel szükségessége és célja. Ha a tétel valaminek az elkészítését kívánja meg, akkor azt inkább csináltassuk meg a tanulóval, mintsem arról beszéljünk, vagy beszéltessek.
5. Ha a tétel többféle szempontból megválaszolható és a cél a tanuló felfogásának és helyes mérlegelésének vizsgálata is, akkor lehetőleg mellőzzük az elcsépett válaszadási lehetőségeket.
6. Teljes mértékben használjuk ki a tanuló azon képességét, amellyel a demonstrációs anyagot kezelni tudja, s így kerüljük el azoknak a dolgoknak az elmondását, amiket magától is meg tud, vagy meg tudhat találni. Ugyancsak kerüljük a másoltatással, vagy az "ismételje utánam" módszerrel való foglalkoztatást.
7. A tételeket úgy kell felfogni, mint információs egységeket, nem pedig mint rövid, pár soros szövegeket. Azt, és csakis azt tartalmazzák, ami szükséges.
8. El kell kerülni a teljesen magyarázó tételeket, ahol a tanulótól választ nem várunk. Az ismertetést kívánó anyagot inkább több tételre bontsuk fel.
9. A nem egyértelmű anyagrészeket, a kevésbé elfogadható adatokat merészen el kell hagyni.

10. Helyes egy sajátos témára olyan jellemző mintapéldákat adni, amelyek az eset minden változatára kiterjednek, a megértést elősegítik, sőt, ha a téma nem nyújt egyértelmű magyarázatra szolgáló példalehetőséget, ellenpéldák alkalmazása is célszerű.
11. A témák egymásutánja, programozása állandóan éreztesse és érzékeltesse a tanulóval azt, hogy azok egymásutánján keresztül elérkezik a kívánt célhoz, a tananyag megértéséhez, megtanulásához.
12. A téma kifejtése során a tanuló befolyásolásának csak olyan mértékűnek kell lennie, amennyire feltétlenül szükség van. Határozott befolyásolása csak több megoldási lehetőség esetén indokolt, s annak helyét és mértékét is csak az előző kísérleti anyag alkalmazása során szerzett tapasztalatok határozzák meg.
13. Amikor a tanulót helytelen vagy kevésbé helytelen válasza során "visszalépésekre", illetőleg a témát más oldalról is magyarázó tételek áttanulmányozására utasítjuk, ez ne váltson ki a tanulóban elkeseredést, ne sugározzon belőle személyi "leértékelés" vagy lenézés.
14. Vigyázni kell arra, hogy az egyes tételeken belül elsajátítható információk mennyisége arányos legyen, ne tartalmazzanak a megértéshez túl kevés, vagy túl sok információt. A lépések ésszerű és célszerű nagyságát a tapasztalati próbák mutatják meg. A tételek ne kívánjanak meg a tanulótól hosszú számolásokat, vagy feleleteket.
15. Ne a szerkesztő impressziója döntse el a program helyességét, hanem a kísérleti, tapasztalati próbák. Viszont a kezdeti felelősséget azért is vállalni kell és lehet, mert változtatásokra még a késznek elfogadott programnál is szükség lehet, akár a tananyag korszerűsítése, akár a tanulók átlagképességeinek változása miatt.

A számítógépes oktatóprogramoknak néhány speciális jellemzőjük:

Szakszerű, egyértelmű stílus, a definíciók pontos megadása.

A fokozatosság elvének betartása fontos szempont kell, hogy legyen. Logikus rávezetéssel segíteni kell a helyes megoldást, de nem túl hamar.

Szükség van az időnkénti összegzésre is. Ügyelni kell arra, hogy meglegyen az állandó visszacsatolás. Folyamatos eredménymérés mellett a tanuló azonnal értesüljön tudásszintjéről.

Nem elkerülhető szempont a program "felhasználóbarát" volta sem. Ne csak a főcím legyen mutató, hanem a szövegben is célszerű alkalmazni a színeket. Ennek a figyelemfelkeltés és a figyelem fenntartása szempontjából egyaránt nagy jelentősége van. Mindenki sokkal szívesebben dolgozik színes, ízlésesen megtervezett környezetben.

A program jellegét feltétlenül előre meg kell határozni:

- új anyagot közlő
- információt közlő
- gyakoroltató
- ellenőrző
- szimulációs
- problémamegoldó
- játék
- adminisztrációs

Számítógépek az oktatásban

A számítógépek alkalmazása az oktatásban több mint húsz éves múltra tekinthet vissza. Csoma Gyula A számítógép az iskolában című könyvéből idézve: " Évek óta tudjuk, hogy a számítógép milyen bámulatos teljesítményekre képes, s hogy ezek milyen hatással lehetnek az ember életére. Alkalmazási lehetőségeit az oktatás területén megvizsgálták és kipróbálták már. De a számítógépek még a közelmúltban is terjedelmes és drága készülékek voltak; felhasználásuk az oktatásban csak lassan terjedt, és csaknem teljes egészében a világ leggazdagabb országainak elit intézményeire korlátozódott."

A nyolcvanas évek elejére ez a helyzet gyökeresen megváltozott. Az 1981-ben rendezett III. Oktatási - Informatikai Világkonferencián Ershov professzor megállapította: "A számítógép-programozás néhány éven belül olyan lesz, mint egy második ábécé, s ugyanolyan nélkülözhetetlen lesz, hogy tudjunk számítógéppel dolgozni, mint amilyen ma az, hogy tudjunk olvasni és írni. A számítógépek újjáteremtik az emberi szellem harmóniáját, ami jelentősen fokozza majd az emberiség intellektuális erejét". Ha ezek a jóslatok valóra válnak, az iskolák - olyan mértékben, ahogy számítógépek birtokába jutnak - először is azzal a feladattal kerülnek majd szembe, hogy kidolgozzák a számítógép alkalmazásába való bevezetés programjait.

A számítógépek felhasználásának és hozzáférhetőségének exponenciális növekedésével változás történt az informatikai ábécé elsajátításának adott jelentésében. Korábban ehhez tudni kellett, hogy a számítógép mire képes, hogyan működik, melyek a különböző felhasználási területei. Ma ennek az ábécének az ismerete egyszerűen annyiból áll, hogy a számítógépet "tudjuk használni".

Az angol igeidők

Első lépésként azt érdemes tisztázni, hogy az idő és az igeidő nem ugyanazt jelentik. Vegyük példának először a magyar nyelvet: a "bemegyek" szó jelen idejű alak. Ha viszont úgy mondjuk a mondatot, hogy "Holnap bemegyek az üzletbe és megveszem az ajándékot, amit kinéztem." már a jövőre utal. Lehet azonban múlt időre utaló használata is: "Képzeld el, tegnap mi történt velem! Szomjas voltam, a közelben pedig volt egy bolt, bemegyek – hát kit láttam benn?" Végezetül lehet jelen idejű jelentése is, mint a következő mondatban: "Én minden reggel bemegyek a munkahelyemre."

Az igeidő az események valamilyen sajátos szempontból való vizsgálata. Ezek közé a szempontok közé akár az idő is tartozhat, azaz, hogy múlt, jelen vagy jövőbeli eseményről van-e szó. Az angol logika úgy gondolja, hogy ezeket meg kell különböztetni, tehát kialakult benne a három különböző fajta igeidő:

Múlt (Past)	Jelen (Present)	Jövő (Future)
----------------	--------------------	------------------

Az angol nyelv alapvetően úgy is használja ezeket az igeidőket, ahogy várnánk: a múlt igeidőket múltbeli események leírására, a jelen igeidőket a jelenre és a jövőt jövőre. Persze itt sem ennyire egyszerű a helyzet. Ahogy a magyar nyelvben, egyes esetekben az angolnál is ugyanúgy felborul a rend, és előfordul, hogy jelen igeidővel fejez ki pl. múltbeli eseményt.

Ez akár már elég is lehetne, gondolhatnánk. Van egy igeidő-rendszer, amivel mindent ki lehet fejezni, különbséget lehet tenni múlt, jelen és jövő között. Felmerült azonban az igény, hogy egyéb különbségeket is kifejezzenek egyszerűen csak az idő meghatározásán kívül. Ez nem az angol nyelv sajátossága, természetesen a legtöbb nyelvben megvan ez. Ha egy kicsit utánanézzünk, hogy mi a helyzet a magyar nyelvvel, azt találjuk, hogy az sem kivétel. Ha a következő mondatokat szemügyre vesszük, kiderül, hogy nagyon nem mindegy, hogy éppen melyiket használjuk.:

- 1a. Tanultam vezetni.
- 1b. Megtanultam vezetni.
- 2a. Festette a körmét.
- 2b. Kifestette a körmét.
- 3a. Mosogatok majd holnap.
- 3b. Elmosogatok majd holnap.

Az (a) jelű mondatok azt jelzik, hogy volt/lesz/van olyan cselekvés, amely a mondatban szerepel, míg a (b) jelűek azt hangsúlyozzák ki, hogy a cselekvések már befejeződtek, valamilyen eredménnyel lezárultak. A magyar nyelv az igeekötőket használja ehhez. Ám az angolban nincsenek igeekötők, ráadásul az sem várható el, hogy ugyanúgy fejezze ki a gondolatait, mint más nyelv. Az angolban ezeket a különbségeket más-más igeidő használatával érik el. Kialakultak erre az úgynevezett befejezett (perfect) igeidők. Ezek, többek között, azt fejezik ki, hogy az adott esemény már lezajlott. Léteznek még emellett azok az igeidők, ahol ez a jelentés nincsen meg. Ezek az egyszerű igeidők. Ennek megfelelően az igeidők táblázata a következőképpen alakul.:

	Past	Present	Future
Simple (Egyszerű)	Simple Past	Simple Present	Future Simple
Perfect (Befejezett)	Past Perfect	Present Perfect	Future Perfect

Ez a rendszer már megfelelően bonyolultnak tűnik, hihetnénk, de ez továbbra sincsen így. Az angol nyelvben még ehhez a táblázathoz képest is vannak olyan jelentéskülönbségek, amelyeket nem lehet kifejezni csak azokkal a szerkezetekkel, amelyek itt vannak felsorolva. Nagyon hasonlít ez a helyzet a magyar nyelvhez, hiszen ezek a különbségek ott is megvannak. Ha szemügyre vesszük a következő mondatokat, azt látjuk, hogy itt sem mindegy, hogy melyikről beszélünk, azaz milyen különbségeket fejezünk ki bennük:

- 4a. Mentem át az utcán, amikor észrevettem, hogy engem szólít.
 4b. Átmentem az utcán, amikor észrevettem, hogy engem szólít.
 5a. Amikor mentünk fel az emeletre, megkérdeztem, van-e lift.
 5b. Amikor felmentünk az emeletre, megkérdeztem, van-e lift.

Az (a) jelű mondatokban a későbbi esemény mindig az előbbi esemény közben történik, míg a (b) jelű mondatok esetében először megtörténik az első, aztán lezajlik a második esemény. A magyar nyelv esetében a jelentéskülönbségeket a szórend megváltoztatásával érjük el. Nem mindegy, hogy az igekötő az ige előtt vagy után van. Amint megváltozik a helyzete, rögtön megváltoztatja a mondat jelentését is. Az angol nyelv felépítéséből adódóan nem rendelkezik igekötőkkel, ráadásul igen kötött szórendje is van. Talán nem is meglepő, hogy egy ilyen probléma áthidalására az angol nyelvben kialakult egy új igeidő. A dolog valamelyest úgy tűnik, hogy az (a) jelű mondatokban a cselekvést folyamatában tekintjük, tehát az újonnan megismert igeidő is folyamatos (Continuous) igeidő lesz. Ezek a különbségek minden időben megjelenhetnek, ráadásul a befejezett és nem befejezett szemlélet mellett is előállnak, ezért praktikusán megduplázza az igeidők számát. A következőképpen alakul tehát a táblázat, amelyik az igeidőkkel foglalkozik:

		Past	Present	Future
Simple	Simple	Simple Past	Simple Present	Simple Future
	Continuous	Past Continuous	Present Continuous	Future Continuous
Perfect	Simple	Past Perfect	Present Perfect	Future Perfect
	Continuous	Past Perfect Continuous	Present Perfect Continuous	Future Perfect Continuous

Az előbb tárgyaltakból következik, hogy...

Ha az előző gondolatmenetet jobban megnézzük, arra kell következtetnünk, hogy az angol igeidők közötti hasonlóság igen nagy. Ha azt vesszük például, hogy a perfect igeidők miért is alakultak ki, akkor azt látjuk minden perfect igeidő hasonló dolgot fejez ki. Ugyanazt az aspektust tudjuk kifejezni ezen elgondolás alapján az összes continuous (folyamatos) és szintén egyezőt az összes past (múlt) igeidővel. A folyamatos igeidőknél az a fontos, hogy a cselekményt, mint folyamatot szemléli, míg a múlt időknél az a lényeg, hogy a cselekmény a múltban történik, ott zajlódott le.

Természetesen, mint ahogy más nyelvekben is, maguk az igeidők használata az angol nyelvben nem korlátozódik kizárólag csak az előbb tárgyalt esetekre. Sokkal többet is ki lehet fejezni az igeidőkkel pusztán a cselekedetek leírásánál. Erről még hosszasan lehetne értekezni, de jelen dolgozatom nem csak erre hivatott létrejönni, ezért itt most megelégszünk ennyivel.

Simple Present

(Egyszerű jelen idő)

Képzése:

+ A + ige 1. alak (E/3. sz.: ige + s) .

pl.: I usually get up at 7:30.

— A + do/does + not + ige 1. alak .

pl.: He doesn't get up at 7:30.

? Do/Does + A + ige 1. alak ?

pl.: Does he usually get up at 7:30?

Használata:

1. Szokásos ismétlődő cselekvések kifejezésére. (I usually get up early in the morning.)
2. Általános igazságok kifejezésére. (Sun goes down on the West.)
3. Utazással kapcsolatos, megtervezett jövőbeli cselekvés kifejezésére. (I arrive in Prague at 3 p.m. and then I go to Budapest.)
4. Elbeszélő szövegekben, narrációkban /regények, filmek leírásánál/
5. Here/There szavakkal a beszéd pillanatában lezajló cselekvés kifejezésére. (Here they come/There he goes)

Időhatározók:

usually, never, every day, often, sometimes, always, rarely, on Wednesday, once a week, occasionally, generally

Helyesírási tudnivalók:

1. Ha az ige o, x, sh, ch, s-re végződik E/3. személyben –es ragot kap.

2. Ha az ige y-ra végződik, és mássalhangzó előtt áll, E/3. személyben i-re változik (pl.: study ➔ studies)

Present Continuous

(Folyamatos jelen idő)

Képzése:

+ A + be + ige + -ing.

pl.: I am standing.

— A + be + not + ige + -ing.

pl.: He is not sitting.

? Be + A + ige + -ing ?

pl.: Are you listening to the radio at the moment?

Használata:

1. A beszéd pillanatában történő cselekvés kifejezésére. (He is sitting now.)
2. Nem a beszéd pillanatában történő, de mostanában zajó cselekedet kifejezésére. (I am creating a new statue.)
3. Megtervezett szándékos jövő idejű cselekedet kifejezésére. Időhatározóval együtt használjuk. (I am listening to somebody tomorrow.)
4. Az always szó használatával rendszeresen ismétlődő, állandó cselekvés kifejezésére, amely a beszélőt kellemetlenül érinti. (He is always watching TV.)

Időhatározók:

now, at the moment, at present, nowadays

Helyesírási tudnivalók:

1. Az -ing előtt a ki nem ejtett *e* betű kiesik.
2. A szó végi *l*, *p*, *m* megduplázódik.
3. Ha a szó *oe*, *ye*, vagy *nge* végű, akkor megmarad az *e* betű.

4. Ha a szó *ie*-re végződik, akkor az *ie* *y*-ra változik.
5. Vannak igék, amelyeket soha nem használunk folyamatos alakban:
 - érzelmeket kifejező igék (pl.: like, love, hate, prefer...)
 - érzékelést kifejező igék (pl.: smell, see, hear...)
 - birtoklást kifejező igék (pl.: have, owe)
 - gondolati tevékenységekkel kapcsolatos igék (pl.: think, believe, want)
6. Ha a szó egy szótagból áll, mássalhangzóra végződik és a mássalhangzó előtt magánhangzó áll, a szó végi mássalhangzó megkettőződik. (pl.: sitt→itting)

Present Perfect

(Befejezett jelen idő)

Képzése:

+ A + have/has + ige 3. alak .

pl.: He has been to Paris.

— A + have/has + not + ige 3. alak .

pl.: She hasn't been to Paris yet.

? Have/Has + A + ige 3. alak ?

pl.: Have you ever been to Paris?

Használata:

1. A múltban megkezdődött és ott befejeződött cselekvésnek jelenbeli eredménye van. (I have broken the glass.)
2. A cselekedet a múltban kezdődött, a jelenben is tart és a jövőben is tartani fog. (I have lived here for 5 years.)
3. A múltban kezdődött és a jelen pillanatig tartott a cselekvés. (I haven't seen you for ages.)
4. A múltban kezdődött, és a nagyon közeli múltban fejeződött be a cselekvés. /Általában az "Éppen most" kifejezéssel fordítjuk le/ (I have just met her.)
5. Amikor a cselekvés megtörténtét akarjuk kifejezni. (I have never eaten mango. Have you ever eaten kiwi?)

Időhatározók:

yet, already, never, recently, ever, lately, since, for (2 years), just

Kérdőszavak:

How long? Since when?

Present Perfect Continuous

(Folyamatos-befejezett jelen idő)

Képzése:

+ A + have/has + been + ige + -ing .

pl.: I have been waiting for two hours, but the he hasn't come yet.

— A + have/has + not + been + ige + -ing .

pl.: She hasn't been living here since 2002.

? Have/Has + A + been + ige + -ing ?

pl.: Have you been driving since you were 17?

Használata:

1. A múltban elkezdődött és a jelen pillanatban is folyamatban lévő, a jövőben is tovább folytatódó cselekvés kifejezése. (I have been waiting for the taxi for 20 minutes.)
2. A múltban kezdődött és ott be is fejeződött cselekvés kifejezése, melynek a jelenben is eredménye van és a múltbeli folyamatossága a fontos.
 - Lehet időhatározóval: I'm tired, because I have been walking all day.
 - Lehet időhatározó nélkül: I'm tired, because I have been walking.
3. Amikor az eredmény mellett a felháborodáson is hangsúly van. (Who has been eating with my fork?)

Időhatározók:

for (2 years), since

Kérdőszavak:

How long? Since when?

Simple Past

(Egyszerű múlt idő)

Képzése:

+ A + ige 2. alak .

pl.: I went to the cinema yesterday.

— A + did + not + ige 1. alak .

pl.: He didn't write the letter last week.

? Did + A + ige 1. alak ?

pl.: Did you go to work when you were ill?

Használata:

1. A cselekvés a múltban kezdődött, ott be is fejeződött és nincs köze a jelenhez. (I ate a cake yesterday.)
2. Múltbeli szokásos ismétlődő cselekvés kifejezésére. (I went school, when I was 6 years old.)
3. Narratív, elbeszélő szövegekben, történet elmesélésekor.
4. A múltban egy időszakot kitöltő cselekvés, ami lezáródott.
5. Múltbeli cselekvés, ami a jövőben nem ismétlődhet meg. (I met John Lennon.)
6. Múltbeli egyszeri cselekvés. (I watched TV yesterday.)

Időhatározók:

yesterday, yesterday evening, last night, last week, last year, last month, in 1952, in September, in Spring, on Monday, for (5 years), 2 hours ago, 2 years ago

Kérdőszó:

when?

Past Continuous

(Folyamatos múlt idő)

Képzése:

+ A + was/were + ige + -ing .

pl.: She was playing yesterday afternoon.

— A + was/were + not + ige + -ing .

pl.: I wasn't running yesterday in the morning.

? Was/Were + A + ige + -ing ?

pl.: Were you playing the piano while your mother was doing the cleaning?

Használata:

1. A múlt egy adott időpontjában zajló cselekvés kifejezésére, mely folyamatos. (I was eating at 8 o'clock yesterday.)
2. A múltban egy cselekvés kifejezésére, melynek időtartama adott. (I was reading all afternoon.)
3. A múltban párhuzamosan zajló két cselekvés kifejezésére. (While my father was reading the newspaper, Joe was cleaning window.)
4. A múltban folyamatosan zajló *cselekvés* közben egy *esemény* történik. (Az esemény ➔ Simple Past; a cselekvés ➔ Past Continuous) (I was having a bath when my sister came home.)

Időhatározók:

at 4 o'clock, all day, when..., as..., while..., from...to, between...and...

Past Perfect

(Befejezett múlt idő)

Képzése:

+ A + had + ige 3. alak .

pl.: She had done the washing up, when her son came home.

— A + had + not + ige 3. alak .

pl.: He hadn't made the tea, when his husband got up.

? Had + A + ige 3. alak ?

pl.: Had you done your homework by that time?

Használata:

1. Egy múlt idejű cselekvés kifejezésére, amely a másik, szintén múlt idejű, cselekvéshez viszonyítva:
 - már befejeződött. (I had finished my picture when Joe arrived home.)
 - még folyamatban van. (I had been in the choir for 10 years when we met.)
2. Az időben közelebbi cselekvést helyettesíthetjük időhatározós szerkezettel is. (By that time, by then, by 1 o'clock)
3. Múltbeli cselekvés, amelynek a múltban volt eredménye is. (I had worked on the ground that's why I was tired two days before.)

Kötőszavak:

when, before, by the time, after, by that time, by...

Past Perfect Continuous

(Folyamatos-befejezett múlt idő)

Képzése:

+ A + had + been + ige + -ing .

pl.: I had been living here for 1 year, when we met.

— A + had + not + been + ige + -ing .

pl.: He hadn't been working here, when the new coach came.

? Had + A + been + ige + -ing ?

pl.: Had you been living here, when your neighbour died?

Használata:

1. Egy múlt idejű cselekvéshez képest egy másik múlt idejű cselekvés már elkezdődött és azután is folytatódott. (I had been working for the company for 2 years when we met.)
2. A múlt idejű cselekvésnek a múltban eredménye volt. (I was tired, because I had been working in the factory /all day/.)

Időhatározók:

for (3 hours), since

Simple Future

(Egyszerű jövő idő)

Képzése:

+ A + will + ige 1. alak .

pl.: I will go to eat.

— A + will + not + ige 1. alak .

pl.: I won't drink the juice.

? Will + A + ige 1. alak ?

pl.: Will you do the cleaning?

Használata:

1. Jövő idejű cselekvés kifejezésére, amit a beszélő ebben a pillanatában dönt el. (What will you do now?)
2. Jóslásnál. (Your test will be successful.)
3. Ígéretnél. (I will do my homework.)
4. Szándéknál. (I will help you.)
5. Fenyegetésnél. (I will kill you.)

Időhatározók:

next Saturday, tomorrow, in 4 weeks, by tomorrow

Future Continuous

(Folyamatos jövő idő)

Képzése:

+ A + will + be + ige + -ing .

pl.: I will be listening to the radio from 5pm to 6pm.

— A + will + not + be + ige + -ing .

pl.: She will not be reading, when her brother goes away.

? Will + A + be + ige + -ing ?

pl.: Will you be playing football tomorrow afternoon?

Használata:

1. A jövő egy adott időpontjában vagy tartamában folyamatosan zajló cselekvés kifejezésére. (I will be running at 5 o'clock tomorrow.)
2. A jövőben két cselekvés összehasonlításakor, amikor az egyik egyszeri cselekvés, a másik pedig folyamatosan történik. (I will be singing when he comes home)
3. Olyan jövő idejű cselekvés kifejezésére, amely rendszeresen ismétlődik, azaz biztosan meg fog történni.

Future Perfect

(Befejezett jövő idő)

Képzése:

+ A + will + have + ige 3. alak .

pl.: By the end of the month, my brother will have been 22 years old.

— A + will + have + not + ige 3. alak .

pl.: I will haven't finished this work by tomorrow.

? Will + A + have + ige 3. alak ?

pl.: Will you have read the book by the next month?

Használata:

1. A jövő egy adott időpontjában az adott cselekvés már múlt lesz, vagy be lesz fejezve. (By the end of the week, he will have been here for 5 years.)

Future Perfect Continuous

(Folyamatos-befejezett jövő idő)

Képzése:

+ A + will + have + been + ige + -ing .

pl.: By the end of March we will have been living here for 2 years.

— A + will + not + have + been + ige + -ing .

pl.: He won't have been living here for 2 years by the end of the year.

? Will + A + have + been + ige + -ing ?

pl.: Will you have been going in this school for 3 years by the end of the year?

Használata:

1. A jövő egy időpontjából visszatekintve a cselekvés már egy ideje folyamatban van, még nem fejeződött be, és nem is tudjuk, hogy mikor fog befejeződni. (By the end of the next month, he will have been working here for 2 years.)

A program megvalósítása

Általános szempontok, melyeket figyelembe kellett venni a megvalósításkor

Az alkalmazás megvalósításakor több szempontot is figyelembe kellett vennem. Ismerve az iskolák helyzetét, helyes iránynak tartom, hogy az alkalmazást egy olyan környezetben készítsem el, ahol megglehetősen szabadsággal lehet mozogni, ezentúl mindenki számára elérhető, azaz ingyenes. Napjainkban szerencsére nem nehéz ilyen keretrendszereket találni, hiszen az a tendencia figyelhető meg, hogy egy-egy programozási nyelvhez rengeteg ún. fordítót készítenek szerte a világban. Természetesen ezek nagy része ingyenes, melyekből könnyen kiválasztható az, amelyik jó referenciákkal rendelkezik.

Egy alkalmazás elkészítése igen sok munkát ró a program elkészítőjére. Rengeteget kell terveznie, és igen sok szempontot kell figyelembe vennie. Továbbá akármennyi energiát is fektet valaki a gondos tervezésbe, még mindig szinte biztosra veheti, hogy előre nem látott problémákkal fog szembesülni. Ahogy az előző bekezdésben említettem is, mindenképpen szükséges egy olyan környezet kiválasztása, amelynek legalább a futtatási környezete ingyenes és mindenféle platformra elérhető, bár manapság ezzel nem tapasztalható igazán probléma, hiszen mindenkinek az az érdeke, hogy a keretrendszerét minél többen használják, minél többen ismerjék, ezért magát a futtatási környezetet ingyen és bárholonnan elérhetően biztosítják a nagyközönség számára. Bár hardver oldalról a legtöbb iskola aránylag fejlettnak mondható, nem lehet teljes bizonyossággal kijelenteni, hogy az elkészült program minden környezetben és minden iskola géptermeiben tudna majd futni. Azonban szerencsés helyzetben vagyok, hiszen szüleim révén betekintésem van egy igen kis falu szintén nem túl nagy iskolájának számítógépparkjának felépítésébe. Ez alapján kijelenhetem, hogy az általam használt keretrendszer és annak futtatási környezete eléggé rugalmas ahhoz, hogy a legtöbb helyen futni tudjon Magyarországon.

A program elkészítésekor jelentkező sajátosságok

A gyakorlati megvalósításkor magát az alkalmazást három főbb egységre bontottam. Az első egységben, mint egy hagyományos iskolai órán, az információ közlése volt a cél. A második egységben egyfajta számon kérő ismétléssel igyekszik a program az információs részben ismertetett anyag bevézésére, míg a harmadik részben egy teljes mértékben számon kérő és értékelő folyamat zajlik le.

Az információs panel

A program felépítése jól illeszkedik ahhoz az alapelvhez, hogy magát az információt nem az alkalmazáson belül tároljuk, hanem külső adatforrásból, esetemben gyakorlatilag egy HTML fájlból olvasom ki. Ez tulajdonképpen azt jelenti, hogy a program szerves részét képezi az a szöveg, amit meg kell jelentenie. Ennél a panelnél azt az elvet követtem, hogy mivel sokkal könnyebb egy HTML dokumentumot előállítani külön, mint egy hosszabb szöveget egy programon belül megjeleníteni, ezért magát a szöveg formázását a programon kívül oldom meg. A program ezen része lényegében beolvassa és megjeleníti a szöveget.

A gyakorló panel

A tanulási folyamatban rendkívül fontos, a tanulók az új ismeretet azonnal jól értsék meg. Ha valami helytelenül rögzül a folyamat során, azt később igen nehéz helyretenni. Ez független attól, hogy milyen korosztályról beszélünk. Az oktatóprogramok egyik nagy hátránya, hogy az ehhez hasonló helyzeteket, melyek sokszor előre nem látott események, nehézkesen vagy egyáltalán nem képesek kezelni. Ezt a problémát hivatott enyhíteni a gyakorló panel a programon belül, melyben ellenőrző kérdésekkel megpróbálom feltérképezni a tanulóknál jelentkező esetleges hiányosságokat. Programozási szempontból ez azt jelenti, hogy szintén kívülről vezérlem a programot. Azonban itt már nem elég csak egy szöveget a képernyőre tenni. Az egyszerűség kedvéért ebben az esetben teljesen hétköznapi szöveges állományt használok. Az állományban azonban speciális jelentése van a sorok elhelyezkedésének. Ilyen módon tudom megvalósítani azt, például hogy az egyébként a fájlban mindig jelen lévő extra információt, magyarázatot csak akkor jeleníti meg a program, ha ez éppen szükséges, azaz a tanuló nem tudta a kérdésre a választ, tehát ezt a bizonyos részt újra kell magyarázni. Ez természetesen azt is jelenti egyben, hogy az ehhez a részhez tartozó szöveges állományoknak jóval rögzítettebb

formátuma van, mint az információs panel HTML részének. Itt azt az elvet követem, hogy a soroknak jelentése van. Egy kérdés úgy áll össze az állományban, hogy az első sor maga a kérdés. A következő négy sor a négy lehetséges válasz. Az ezután következő sor az esetleges magyarázat a rossz válasz esetére. Így tehát minden 6 sor felel meg egy kérdésnek a `kerdesek.txt` nevű állományban. Itt kell még megjegyezni, hogy magában a fájlban a válaszok sorrendje nem változik sosem. Azonban ha változtatás nélkül írnám ki ezeket, akkor a tanulók igen hamar rájönnének, hogy mindig az első lehetőség a jó válasz és minden bizonnyal ki is használnák ezt. Ezért találtam ki azt, hogy a válaszokat véletlenszerűen helyezem ki a képernyőre, ezáltal biztosítva a változatosságot. Magában az állományban mindig az első válaszlehetőség a helyes, ezzel tudom elérni azt, hogy a program tudja, mely választ értékelje helyesnek. Mindez addig tart, amíg a program egy olyan sor be nem olvas, amelynek tartalma 'END'. Számára ez azt jelenti, hogy a gyakorlás véget ért. Ekkor egy kis összegzést ad a tanuló felé, melyben jónak értékeli minden olyan választ, melyre akár azonnal vagy akár a segítség után adott jó választ a tanuló.

A teszt panel

Ez a része a programnak nagyban hasonlít a gyakorló részhez, hiszen funkcióját tekintve is hasonló. Ebben a részben egy felmérést végez a program arról, hogy milyen mélységben sikerült az átadott anyag elsajátítása a tanulónak. Hasonló alapelvet követve halad végig a `kerdesek.txt` állományon, mint a gyakorló részben. Itt azonban már nem ad visszajelzést a tanuló számára, hogy az hibázott-e vagy sem. Sorban halad végig a kérdéseken, egészen addig, amíg az előző részben már említett 'END' tartalmú sorhoz nem ér, ahol befejezi a szöveges állomány feldolgozását, és értékeli az addigi eredményeket. Annak érdekében, hogy az értékelés minél inkább testre szabható legyen, a program nem ad érdemjegyet, hanem inkább a jó válaszok százalékos arányát adja meg. Egy esetleges éles használat esetén a tanár, aki a programmal szeretne felmérést csinálni meghatározhatja, hogy mely százaléktartományhoz milyen érdemjegy tartozzon.

A program működése

Az elkészített alkalmazás – illeszkedve a mai kor követelményeihez – egy különálló ablakban fut, melynek több ún. panelje van. A felhasználók a panelek között a panelek tetején található fül részre való kattintással válthatnak szabadon. A fülek felépítése, s ennél fogva a használatuk is eltérő. Míg az első panelben főleg az információ prezentálásán van a hangsúly, addig a második két panel már inkább a számonkérés irányába tolódik el.

Az ismeretközlő panel működése nem szorul különösebb magyarázatra, mindössze annyit érdemes megjegyezni, hogy a benne található szöveget egy szövegdobozban lehet elolvasni és természetesen görgetni is. A szöveg formázása illik a tananyag jellegéhez, és jól tagoltan prezentálja az átadandó ismereteket.

A gyakorló panel részhez kicsit bővebb magyarázat illik. Ebben a részben a felhasználó kérdéseket kap sorra. Ezek a kérdések a következőképpen helyezkednek el a panelen. Az első sorok mindig maguk a kérdések. Ez nagyban követi az előző panelen felsorolt tananyagot. Minden kérdésre a program ad négy különböző válaszlehetőséget. Ezekből a lehetőségekből mindig csakis egy válasz a helyes. A felhasználó által helyesnek vélt válasz melletti körbe (vagy magára a válaszra) kattintva lehet ezt kijelölni. Ha a felhasználó meggondolja magát, még mielőtt a program számára is tudatná, hogy melyik választ szeretné helyesként megjelölni, megteheti, mindössze egy másik válaszra kell kattintania. Ekkor az előző kijelölés helyett az új kijelölés lesz az aktív. Egyszerre csak egy választ lehet helyesként megjelölni. Amennyiben a felhasználó elégedett a választásával, rákattint a 'Következő' feliratú gombra. Ekkor a program vagy egyszerűen beadja a következő kérdést, jelezve ezáltal, hogy a válasz helyes, vagy jelzi a felhasználó számára, hogy a válasza helytelen volt. Egy felugró ablak tájékoztatja a felhasználót, hogy helytelenül válaszolt. Miután ezt az ablakot nyugtázta az OK gomb segítségével, a felhasználó megkapja az előző kérdést, de ezúttal kap hozzá némi magyarázatot is, ami alapján már minden bizonnyal könnyebb lesz megadni a helyes választ. Amennyiben a válasz most sem helyes, újabb javítási lehetőség már nincs, a következő kérdést fogja a tanuló megkapni. A következő kérdéssel az előbb leírtak kezdődnek újra. Az utolsó kérdés után egy kisebb összegzést kap a felhasználó, de mivel ez csak gyakorlás volt, ezért semmilyen jellegű értékelést nem készít a program a teljesítményről.

Programozói dokumentáció

A programmal szemben támasztott követelmények, valamint a ma elvárt jellemzők alapján a következő eszközöket és technológiákat választottam a kitűzött feladat megoldására, elkészítésére.

A felhasznált nyelv a C++ ami a legtöbb operációs rendszeren-és platformon megtalálható. Gépközei adottságai miatt rendkívül gyors kód fordítható vele valamint a fejlesztés menetét a nyelvi eszközei rugalmassá, robusztussá teszik. Megfelelő körültekintéssel a programkód akár forrásszinten portolható lesz más architektúrákra és platformokra, ami ma már alapkövetelmény. Az elkészült bináris file mérete töredéke magasabb szintű nyelveken megírt változatoknak valamint nem szükséges hozzá külön futtatókörnyezet.

A kiválasztott fordítóprogram a GNU alapítvány GCC projectjében szereplő G++ nevű eszköz, az alábbiak miatt:

- ingyenes, tehát bárki hozzájuthat,
- szinte minden platformra létezik,
- kiforrott,
- nyílt forrású, emiatt azonnali-és rugalmas támogatást élvez a háttérben lévő igen méretes saját társadalma által.

Windows platformon a cygwin project valósít meg egy teljes POSIX környezetet, aminek része a G++ program.

A felhasznált grafikai könyvtárgyűjteményt szintén platformfüggetlensége és könnyű használata, sokoldalúsága miatt választottam ki. A WxWidgets project teljesen platform független grafikus felület eszköztár készítését tűzte ki céljának valamint C++ nyelven készült és elsődlegesen ezt a nyelvet is támogatja.

A WxWidgets rendszer sok olyan terhet vesz le a fejlesztő válláról, amelyek megvalósítása sok időbe telhet. Emellett igen nagyszámú hibalehetőséget is rejt magában. Továbbá a platformok közötti különbségeket és hibás vagy szokatlan jellegzetességeket is lekezei, ezzel is segítve a

fejlesztés menetét. Ugyancsak egy rendkívül jó ok a rendszer használata mellett a széleskörű elterjedtsége, hiszen már jelenleg is igen sok sikeres project használja alapjául. Közel teljes absztrakciót nyújt az alkalmazás számára forrásszinten is. Ezáltal a legtöbb platformon szinte azonnal lefordítható lesz maga a kód. A belső felépítéséből adódó eseményvezérelt működését forrásnyelvi makrókkal oldja meg, ami egyértelmű hozzárendeléseket eredményez, így az egy-egy operációsrendszerre jellemző, specifikus tévedések, helytelen függvényhasználatok elkerülhetők.

Támogatja a nemzetközi karakterszabványokat is, amit szintén platformoktól függetlenül igyekszik azonos szinten tartani.

A rendszer könyvtárai használhatóak közvetlen a bináris mellett elhelyezve is. Ez természetesen operációs rendszerenként változhat. Emiatt Windows operációs rendszer esetén is a program telepítése megegyezik a program bináris-és a használt függvénykönyvtárak valamint a használt adatállományok azonos könyvtárba való másolásával.

További lehetőségek a program hasznosítására

Mivel a keretprogram elkészült, a további hasznosítása már csak a felhasználón múlik. A lehetőségek adottak, azonban természetesen figyelni kell néhány kötöttségre, melyek a kis keretrendszer sajátjai.

Elsőként meg kell említeni, hogy az átadandó információ egy hagyományos HTML állományban található. Ennek a szövegnek a cseréje könnyen és egyszerűen megoldható, hiszen semmi más sem kell tenni, mint egy új 'leiras.html' állományt létrehozni, mely tartalmazza az új információt. Ennek az állománynak ráadásul még nem is kell kötelezően az angol nyelvről szólnia ezáltal a program még többet is nyújt, mint amennyit e szakdolgozat címe sugall. Akármely témáról szólhat egy újabb leírás – kérdések kombináció.

A kérdéseket szintén egyszerűen és könnyen újra lehet kreálni. Mindösszesen annyit kell tenni, hogy egy új 'kerdesek.txt' állományt kell létrehozni. Azonban itt figyelni kell arra, hogy ennek az állománynak igen kötött formátuma van. Amennyiben ez nincsen betartva, borulhat az egész program működése. A program készítése közben akadt néhány igen makacsul jelenlévő probléma melyet, mint később kiderült, egyszerűen az okozott, hogy a 'kerdesek.txt' állomány szerkezete nem volt megfelelő. Lényegében itt arról van szó, hogy egy tényleges kérdéshez hat sor tartozik a 'kerdesek.txt' állományban. Az első sor tartalmazza magát a kérdést, a következő négy sor pedig a négy lehetséges választ. Itt mindig az első válasz a helyes, de csak magában az állományban. A program futása során a válaszok véletlenszerűen kerülnek a képernyőre, ezáltal biztosítva azt, hogy a program ne legyen könnyen kiismerhető. A hatodik sor a rossz válasz esetén megjelenő magyarázat, ami alapján a jó válasz könnyebben kitalálható. A kérdések sorát egy 'END' tartalmú sor zárja. Ezen útmutató alapján bármikor és bármilyen témában készíthető újabb kérdéssor.

Az utolsó panelen található számonkérés alkalmassá teszi a programot arra is, hogy egy feleltetést is le lehessen bonyolítani a segítségével. Mivel az osztályzás igen eltérő lehet iskolánként, ezért sem szöveges, sem jegyben kifejezett értékelés nem került a programba. Ehelyett a helyes és helytelen válaszok százalékos aránya kerül megjelenítésre a kérdéssor kitöltése után. Innen kezdve az osztályzás már a felhasználó pedagógus feladata, de

természetesen a nagyobb munkát jelentő dolgozat vagy felelés minőségének kijelzése segítségére van az értékelés elkészítésében.

Mindenképpen szeretném megemlíteni, hogy egy lehetséges továbbfejlesztési változatként a programot használhatóvá lehetne tenni némi változtatással arra, hogy egyfajta on-line működési módú legyen. Ezalatt azt értem, hogy a leírást és a kérdéseket tartalmazó állományt elérhetővé lehetne tenni az Interneten, ami által maga a program igen sok tanulót tudna kiszolgálni egy időben. Ehhez valójában nem is kellene igazán nagy átalakításokat eszközölni a program szerkezetében.

Összefoglalás

A szakdolgozatom megírásával célom az volt, hogy egy oktatóprogramot hozzak létre, mely alapvetően az angol nyelvvel, mint oktatási anyaggal foglalkozik. A program megírásnál igen sok szempontot kellett figyelembe vennem. A hasonló megközelítéssel készült programok igen széles témakört ölelnek fel, ezért egy kisebb áttekintést adtam erről a dolgozatom elején. Részletesen szólok a programozott oktatásról és annak fajtáiról.

Részletesen foglalkozom az oktatási programok tervezésével, hiszen ez igen közel áll mindahhoz, amivel tanulmányaim során foglalkoztam. Egy hasonló programot nem lehet csak légből kapott ötletként hirtelen leprogramozni, mert ez a probléma annál jóval bonyolultabb. Gondos és hosszadalmas tervezést igényel, melynek később meg is látszik a gyümölcse, mert a tervezésre rászánt és ráfordított idő a megvalósításnál bőségesen megtérül. Persze fordítva is meg lehet fogalmazni ezt a jelenséget, hiszen a tervezésnél megspórolt időt viszont kamatostól kell megfizetni a programozás során.

Az oktatási programok a programokon belül egy igen jól meghatározható csoportot takar. Ennek következtében az ilyen programok elkészítésekor igen jól meghatározható szempontokat kell figyelembe venni. A dolgozatomban erről is szólok részletesebben, mert az itt meghatározott szempontrendszer követése igen hatékonyá tud tenni egy készülő oktatóprogramot. Mivel igen sokan tanulmányozták már az oktatóprogramok sajátosságait, ezért meglehetősen nagy tapasztalattal is rendelkezünk ezen programokkal kapcsolatban. Ezen tapasztalatokat igyekeztem felhasználni a saját témakörömben készülő program készítése során.

Választott témaköröm az angol nyelvtan. Ennek egy részletét választottam ki, mely mintegy példaként szerepel a dolgozatomban. Ez a részlet az angol igeidők, melyek az angol nyelv alapvető pilléreiként szolgálnak. A nyelvtani szabályokról és szabályosságokról részletesen leírtam azt, amit egy ilyen léptékű dolgozat megkíván.

Végezetül a program megírásának részleteiről írtam le gondolataimat, tapasztalataimat. Itt is részletesen kitérek a technikai megvalósításra, a megvalósítás közben fellépő problémákra, és azok megoldására is. A program működését is részletesen leírom, mert bár azok számára, akik ezt a dolgozatot olvassák sok dolog egyértelműnek tűnik, akadnak sokan a potenciális felhasználók

közül olyanok, akik nem rendelkeznek olyan mély ismeretekkel a számítógép használatáról, mint az előbb említett csoportba tartozók. Az ő számukra viszont készítettem egy sokkal inkább technikai leírást is, mely mintegy programozói dokumentáció szerepel dolgozatomban.

Felsorolok végezetül néhány olyan további felhasználási módot, melyek megvalósítása inkább tartozna egy másik dolgozat témájába vagy csak egyszerűen túlságosan csapongóvá tette volna a dolgozatom megjelenését.

Végezetül, de nem utolsó sorban szeretném megköszönni azt a mérhetetlen segítséget, türelmet és időt, mellyel feleségem és lányom e mű létrejöttéhez hozzájárult!

Felhasznált irodalom

Dohár Péter: Kis angol nyelvtan, Dover, 2007.

Pedagógiai Lexikon, Kereban Kiadó Budapest, 1997.

Takács E.: Programozott oktatás Bp. 1978.

Dr. Hátori Miklós : Tanulás és tanítás számítógéppel, Tanönyvkiadó, Budapest, 1983.

Vágner Gyula : Számítógépes oktatás, Egyetemi Számítóközpont, Budapest, 1974