SZAKDOLGOZAT

Zay László

Debrecen
2010

Debreceni Egyetem
Informatika Kar

KÖZÉPISKOLÁS GEOMETRIAI TÉTELEK BEMUTATÁSA OKTATÓPROGRAMMAL

Témavezető:
Nyakóné dr. Juhász Katalin
tudományos főmunkatárs

Készítette:
Zay László
informatikatanár

Debrecen
2010

TARTALOMJEGYZÉK

1. BEVEZETÉS

Az informatikatanári szak elkezdése előtt is mindig magával ragadott a középiskolás matematikai anyag és azon belül a geometria. Ez a terület az, ahol mindent törvények szabályoznak és az egész rendszer axiómák, definíciók és egzakt elvek alapján működik. A programozási nyelvek (kezdetben Pascal) megismerése ugyanilyen érzéseket kelt az emberben. Minden, pontosan és egyértelműen meghatározott törvények alapján van felépítve. A szintaktika mindent meghatároz és eldönt. Kérdés, hogyan lehet ilyen merev rendszerben dolgozni, ahol mindig a törvények által létrehozott falakba ütközik az ember? A helyzet az, hogy nemcsak falak vannak, hanem utak, lépcsők, folyosók ahol kényelmesen lehet mozogni és nagyon sok eszköz, amivel új utakat, folyosókat és épületeket lehet létrehozni. Az embert magával ragadja az új, a látványos, az eddig nem ismert valami létrehozásának a vágya, és a cél elérése után a sikerélmény.

A szak második évfolyamán megjelent a Programozás [8] című tantárgy, ami a Pascalon kívüli más nyelvek megismerését is lehetővé tette (Java, C). Sajnos a tantárgy gyakorlati oldala (időhiány) nem adott annyi lehetőséget, ami jobban megmozgatta volna az ember fantáziáját, és egészen alacsony szinten ösztönözte volna arra, hogy valami újat hozzon létre. E nyelveket már alapszinten használva is érdekes dolgokat lehet készíteni. Szintén a második évfolyamon jelent meg a Számítógépes grafika [10] című tárgy, ami a két területet, a geometriát és programozást összekapcsolta a C programozási nyelvet használva eszközként.

Ezek után nem sokat kellett gondolkodni azon, hogy milyen témát dolgozzon fel a szakdolgozat, ami nem lehetett más, mint valamilyen egyszerű geometriai probléma(-ák) látványos feldolgozása és bemutatása a középiskolai tananyag felhasználásával. A dolgozathoz elkészített program természetesen sok elemet felhasznált a Számítógépes grafika című tárgyban használt algoritmusokból, de azok itt teljesen más geometriai célra lettek felhasználva.

A téma nagyon sok új elméleti problémát hozott felszínre akkor, amikor az oktatóprogram készült. A programozási nyelv eljárásai, függvényei, kettő vagy többirányú elágazásai, ciklusai, kifejezései, sztring és grafikus felület kezelő eljárásai, hogyan használhatók ezen egyszerű geometria probléma bemutatására.

Elméleti jelentősége mellett talán nagyobb a munka gyakorlati jelentősége, a nevében benne van, hogy oktatóprogram. A program készítésének célja, hogy a középiskolában, egyszerű geometria tételeket minél könnyebben el lehessen sajátíttatni, úgy, hogy az ne legyen unalmas a programot használó tanuló számára. A geometria látványossá tehető, ha a számítástechnika lehetőségeit jól ki tudjuk használni. A pontokból, egyenesekből, szakaszokból, görbékből felépített ábrák részei akár a felhasználó által is vezérelhetők, ha az egér használatát is hozzákapcsoljuk a tételek bemutatásához. Az egérrel elmozdított objektumrészekhez tartozó paraméterek változása jól látható módon szövegesen vagy numerikusan jelenjen meg az ábra egy másik részén. Így a tanuló azonnal értelmezheti az ábrához kapcsolódó tétel törvényszerűségeit.

A középiskolás geometria tananyagból - amit az utóbbi időben jelentősen csökkentettek - a legfontosabb témák vannak kiválasztva. A munka lépcsőzetesen készül: a feladat vagy programrész leírásával, a felület megtervezésével, az algoritmus felvázolásával, az algoritmizálással, a teszteléssel és dokumentálással a szakdolgozatban. A programrészek többnyire a menürendszer struktúrájának megfelelően alakulnak ki.

2. A PROGRAM SZERKEZETE ÉS TARTALMA

(Az algoritmusok a mellékletekben találhatók!)

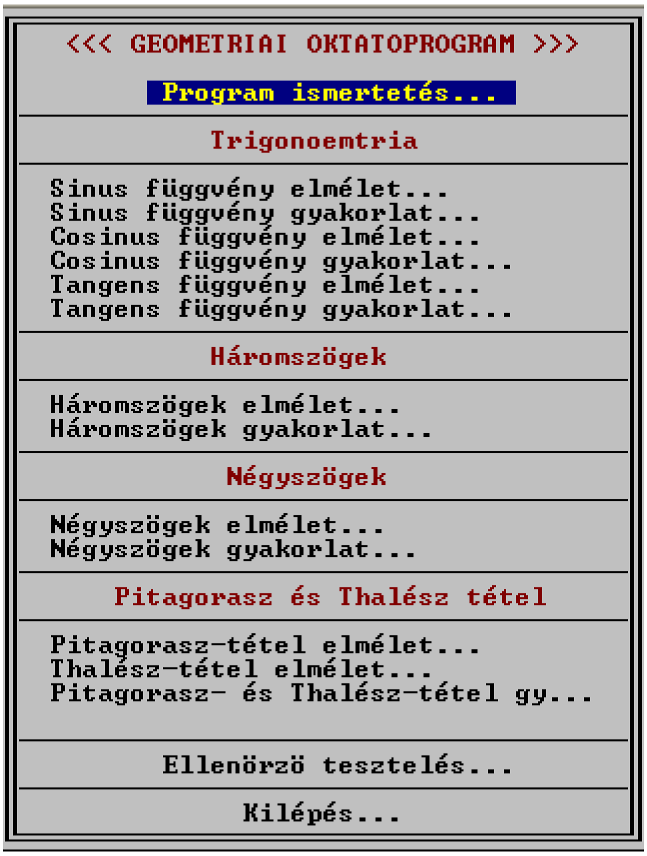
2.1. A szerkezet

A program indításkor (geomokt.exe) egy elérési utat kér, amiben el tudja érni a működéshez szükséges állományokat. Ha nem kap elérési utat, akkor a C:\GEOMOKT mappában keresi az állományokat, amit ki is ír tájékoztatásul. Ennek megfelelően célszerű egy ilyen mappát létrehozni a C: meghajtón, és minden állományt ide kell elhelyezni. Természetesen használható más elérési út is, de azt meg kell adni!

A kezdő felület egy egyszerű menüvezérelt rendszer, aminek használatát ugyanezen az alfanumerikus felületen megjelenő Navigáció nevű ablakban elhelyezett információ mutatja (1. ábra). Nevezetesen azt, hogy hogyan tudunk a menüpontokon mozogni, azokba belépni, esetleg a programot elhagyni.

1. ábra: Navigáció

A kezdő felületen a program nem nagy terjedelmét is mutatatva, minden menüpont megtalálható, az almenüket is beleértve. Az első és utolsó előtti kivételével a főmenük témaköröket tartalmaznak, a hozzájuk tartozó almenük pedig a témakörök grafikus szemléltetését. Az első főmenü a program használatának rövid ismertetése, az utolsó előtti pedig az elsajátítás tesztjellegű ellenőrzésére ad lehetőséget. Az utolsó menüpont a programból való kilépést szolgálja (2. ábra).



2. ábra: Menüszerkezet

Az almenükbe való belépéskor egy grafikus felületre lehet jutni, ahol szövegek és ábrák láthatók. A témakörökön belül az almenükben kétféle grafikus felület létezik. Ezek a statikus és grafikus felületek.

2.1.1. Statikus felület:

Itt a nevéből is sejthetően nem, mozgatható ábrák és szöveges információk találhatók, amik a tananyag elsajátíttatását segítik. Csak a tananyag kellő megismerése után célszerű a dinamikus felületre tovább lépni.

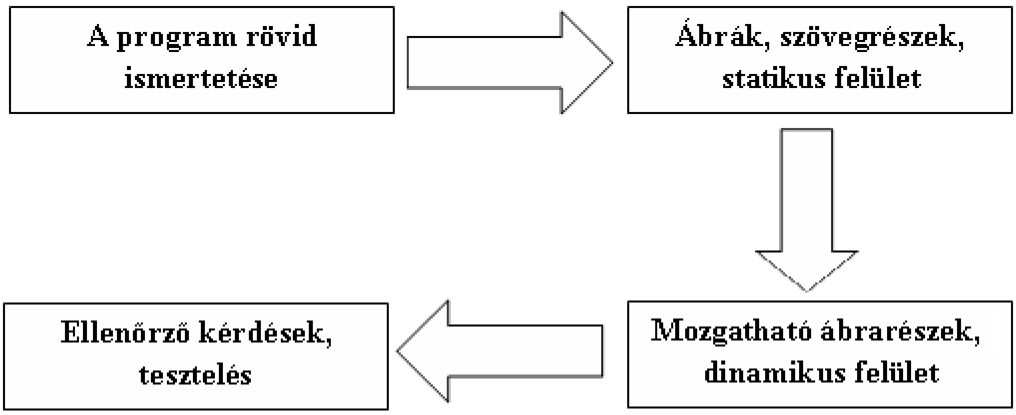
2.1.2. Dinamikus felület:

A dinamikus felületen az előző felület ábráihoz hasonló ábrák találhatók, de itt már nem található szöveges információ a tananyag megértéséhez. Itt az ábrák egérrel és billentyűzettel is mozgathatók. Mivel az ábra tartalmának változtatása annak paramétereit is változtatja, úgy ezek a numerikus és szöveges paraméterek a felület egy másik részén megjelennek, természetesen a mozgatásnak megfelelően változva. Az ábrán lévő egyes részek automatikusan is mozognak, mutatva a hozzátartozó paraméterek változását.

2.1.3. Ellenőrző kérdések, tesztelés

A főmenük egyike egy grafikusfelület, ami egy tesztjellegű kérdez-felelek rendszer, ahol a felhasználó ellenőrizheti, hogy az anyagot mennyire sajátította el. A válaszok közül a megfelelő kapcsoló használatával lehet választani és véglegesíteni a választ. A megoldás alatt is, és a befejezés után is a program mutatja az elért pontszámokat és a százalékos eredményt. (Bemutatás részletesen a 7. fejezetben.)

Összefoglalva: a program szerkezeti vázlata a 3. ábrának megfelelően néz ki.



3. ábra: A program szerkezeti vázlata [4]

2.2. A tartalmi rész

A tartalmi rész magába foglalja a középiskolás elemi geometria néhány fontosabb témakörét és tételeit [7]. Ezek a következők:

1. Legfontosabb trigonometria függvények:
   a. sinus függvény;
   b. cosinus függvény;
   c. tangens függvény;
2. A háromszögek ismertetése, ahol a terület és kerület számítása, valamint a háromszögek fajtái jelennek meg (általános, derékszögű, egyenlőszárú és egyenlő oldalú).

3. A négyszögek ismertetése, ahol a terület és kerület számítása, valamint a négyszögek fajtái jelennek meg (általános négyszög, paralelogramma, trapéz, derékszögű trapéz, deltoid, rombusz, téglalap és négyzet).
4. Pitagorasz-tétel.
5. Thalész-tétel.

2.3. A fejlesztői környezet(ek)

2.3.1. A Turbo Pascal nyelv a menürendszerhez [1, 6]

Indításkor a szoftver (az indító főprogram neve geomokt.exe) kér egy elérési utat, ahol a program összes állományát megtalálja (pl.: *c:\nem\map*). Ha csak <Enter>-t nyomunk, és nem adunk meg elérési utat, akkor a c:\geomokt elérési úton keresi a szükséges állományokat, így célszerű már telepítéskor ezt a mappát (főkönyvtárat) létrehozni.

A menürendszer elkészítésében a Turbo Pascal alfanumerikus fejlesztői környezetét választottam, ahol a menürendszer (későbbiekben főprogramnak is nevezem) felépítése egy hátultesztelős ciklusba épített eljárások segítségével történik. Minden menüpont egy eljárás, ahol az eljárást a nevével hívjuk meg a menüpont sorszámának ismeretében. Az eljárásnak egy szöveges típusú paramétere van, ami az indítani kívánt grafikus program elérési útját és nevét adja meg. A grafikus programok a C nyelvben készültek, amit a következő alfejezet tárgyal.

A hívást az <Enter> (013) billentyű megnyomása jelzi, ahol az algoritmus egy többirányú elágazásba fut bele. A többirányú elágazás kiértékelendő kifejezése a menüpont sorszámát tartalmazó változó. A következő oldal felső részén található algoritmusban ez látható. Ebből az is látható, hogy mik a nevei a C-ben készült grafikus programoknak.

A hátultesztelős ciklus feltétele az <Esc> (027) billentyű használatát vizsgálja. Használatakor a program kilép a ciklusból és így a programból is.

```
if ch=chr(013) then begin
 menu;
 case m of
   1: futtat(ut+'\ism.exe');
   2: futtat(ut+'\sinusst.exe');
   3: futtat(ut+'\sinusd.exe');
   4: futtat(ut+'\cosinst.exe');
   5: futtat(ut+'\cosind.exe');
   6: futtat(ut+'\tangst.exe');
   7: futtat(ut+'\tangd.exe');
   8: futtat(ut+'\hszogst.exe');
   9: futtat(ut+'\hszogd.exe');
  10: futtat(ut+'\nszogst.exe');
  11: futtat(ut+'\nszogd.exe');
  12: futtat(ut+'\pitst.exe');
  13: futtat(ut+'\thast.exe');
  14: futtat(ut+'\pitthad.exe');
  15: futtat(ut+'\teszt.exe');
  16: ch:=chr(027);
 end;
 TextMode(tmod); menu; segit;
end;
```

2.3.2. A C nyelv használata a grafikus részben [2, 3]

A grafikus rész elkészítéséhez a C nyelvet választottam, ami remek lehetőséget biztosít arra, hogy programozó a saját igényei szerint tudja megmutatn, a felhasználónak az alapvető geometriai ismereteket. Minden egyes téma vagy függvény (menüpont) önálló kis programként létezik (neveit az előző algoritmus mutatja) az oktatóprogram főkönyvtárában, amit a Turbo Pascal-ban megírt főprogram menürendszere futtat az Exec eljárás segítségével. A Pascal futtató eljárása, ami az „f” sztring típusú globális változóban található elérési úton lévő C nyelvben megírt állományt futtatja, a következőképpen néz ki:

```
{ ******************* Exe file futtatás! ******************** }
procedure futtat(f:string);
begin
  clrscr;
  {$M $4000,0,0 }
  SwapVectors;
  Exec(f,'');
  SwapVectors;
end;
```

Futtatáskor a kis grafikus program átveszi a vezérlést a főprogramtól, és ha onnan a felhasználó az <Esc> billentyűvel vagy az egérrel kilép, akkor visszaadja a vezérlést a főprogramnak.

A kis grafikus programok csak megtekinthetőek, vagy egérrel. esetleg klaviatúrával vezéreltek, ha a felhasználó már nem kívánja használni, úgy az <Esc> billentyűvel visszaléphet a főprogram menürendszerébe. A grafikus felületen használható egér- billentyűfunkciókat ugyanezen a felületen található információk segítik. Például: <Esc>: kilép, vagy <Space>: irányváltás, vagy H: jobbra lép, vagy G: léptetés, vagy a kis körök egérrel mozgathatók.

A billentyűzet aktiválását egy feltételes utasítás figyeli, amibe egy többirányú elágazás lett elhelyezve, minek a kifejezése nem más, mint a megnyomott billentyű karakterkódja. Így a folyamatosan működő grafikus felület a megnyomott billentyűhöz kapcsolt utasításoknak megfelelően frissül a következő, egy a programból kiemelt algoritmusnak megfelelően [10].

```
if (kbhit()) {                          // billentyűzet kezelés
  bill=getch();
  switch (bill) {
    case 13: {if (a==0) a=b; else a*=-1; break;}     // enter
    case 32: {if (a==0) a=b; else a*=-1; break;}     // space
    case 103: {a=0;b=-2;sz+=-2; break;}              // g betű
    case 104: {a=0;b=2;sz+=2; break;}                // h betű
  }
}
```

Az egér balgombjának aktiválását is egy feltételes utasítás figyeli, amibe olyan feltételes utasítások vannak beágyazva, amik az egér kurzor koordinátáit határértékek közé helyezik, majd azt átadják a program egy másik paraméterének (a következő algoritmus részletben például az „sz” változónak). Az „sz” változó a továbbiakban majd befolyásolja a grafikus felület frissítését [10].

```
if (balgomb) {
  if (egerx<300) egerx=300;     // egérkezelés
  if (egerx>1020) egerx=1020;
   sz=integer(egerx-300);
}
```

Ezek után következzen a program tartalmi része, és minél célszerűbb használatának a bemutatása a geometria oktatásában.

3. TRIGONOMETRIA

A trigonometria kifejezés a görög trigōnon (háromszög) és metron (intézkedés) kifejezésekből származik, a témakör a matematikai egyik ága. Kezdetben a háromszögek szögei és oldalai közötti összefüggések bemutatására szolgált, ezek az úgynevezett trigonometriai függvények [14].

A későbbiekben a megismert hullámtermészetű fizikai mozgások leírására remekül lehetett használni, ahol a független változó (α) értéke már a 360° vagy a 2Π egész számú többszöröseként jelent meg. Ilyen fizikai jelenségek a hang- és a fényhullám is. Ezek a függvények más tudományágakban is nagyon fontosak lettek. A trigonometriát tanítják a középiskolában a matematika részeként a geometrián belül.

3.1. A trigonometria története

Ókori egyiptomi, majd babiloni matematikusoknak hiányzott egy összefüggés, ami a hasonló háromszögek közötti kapcsolatot megmutatta volna. Felfedezték, hogy a hasonló háromszögek esetében az egymásnak megfelelő oldalak aránya azonos. Ókori görög matematikusok, mint például Euklidész és Arkhimédész tanulmányozták a hasonló háromszögek tulajdonságait, és algebrai képletekkel bizonyították a geometria trigonometriai összefüggéseit. A modern sinus függvény tulajdonságait először Surya Siddhanta határozta meg, majd tovább dokumentálta az 5. századi matematikus és csillagász Aryabhata. Mindketten indiai tudósok voltak [12].

Ezeket az indiai műveket tovább bővítette a középkori iszlám tudomány. A 10. században az iszlám matematikusok már használták a trigonometriai függvények táblázatos értékeit, és azokat alkalmazták a szférikus geometria problémáinak megoldására. Ugyanabban az időben az előzőektől függetlenül kínai matematikusok fejlesztették tovább a trigonometriát.

A trigonometriai függvények ismerete és használatának módszerei Al Battani (perzsa) és Nasir al-Din al-Tusi (arab) csillagászok munkáinak latin fordításán keresztül érte el Európát. A legkorábbi trigonometriai művek egyike a „De Triangulis” a 15. századi német matematikus Regiomontanus nevéhez fűződik.

A még oly kevéssé ismert trigonometriának a 16. századi Európában, Nicolaus Copernicus „De revolutionibus orbium coelestium” című műve szentelt két fejezetet, hogy elmagyarázza az alapvető fogalmakat

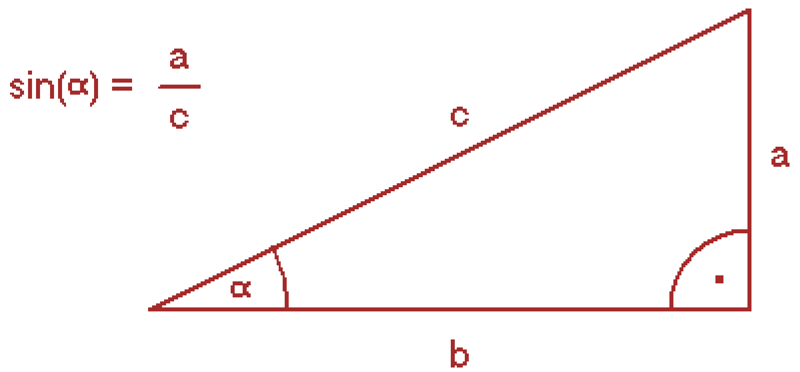
A 17. században, Isaac Newton és James Stirling fejlesztette tovább a trigonometrikus funkciókat. A 18. században, Leonhard Euler volt az, aki Európában a trigonometrikus függvények segítségével meghatározta a végtelen sorozatokat bemutató Euler-képletet. Ugyancsak a 18. században, Brook Taylor határozta meg az általános Taylor-sort, minek segítségével hatványozás révén előállítható mind a hat trigonometrikus funkció.

3.2. A három leghasználatosabb trigonometriai függvény bemutatása

3.2.1. A sinus függvény elméleti része

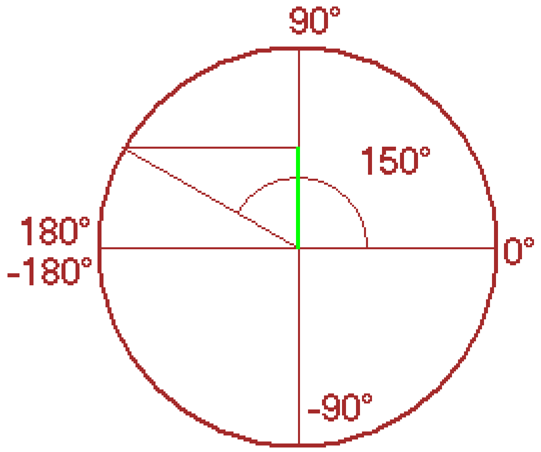
Középiskolás tananyag miatt fontos, hogy a sinus függvény értelmezési tartományát először csak (háromszög) 0° és 90° közé vegyük. A menüpont elméleti részének első fele így vezeti be a függvényt, (4. ábra) utalva a szöggel szemközti befogó és átfogó hányadosára. Az értékkészlet ebben az esetben még csak nulla és egy egész közé esik, mivel az átfogó mindig nagyobb, mint bármelyik befogó [7].

Nagyon fontos a jó ábra, ami kiemeli, hogy derékszögű háromszögről van szó, és rögzíti az átfogó és a befogók helyzetét. A definíciót is kapcsolni kell az ábrához: ***derékszögű háromszög estében egy szög sinusát a szöggel szemközti befogó és az átfogó hányadosa adja*** [13].



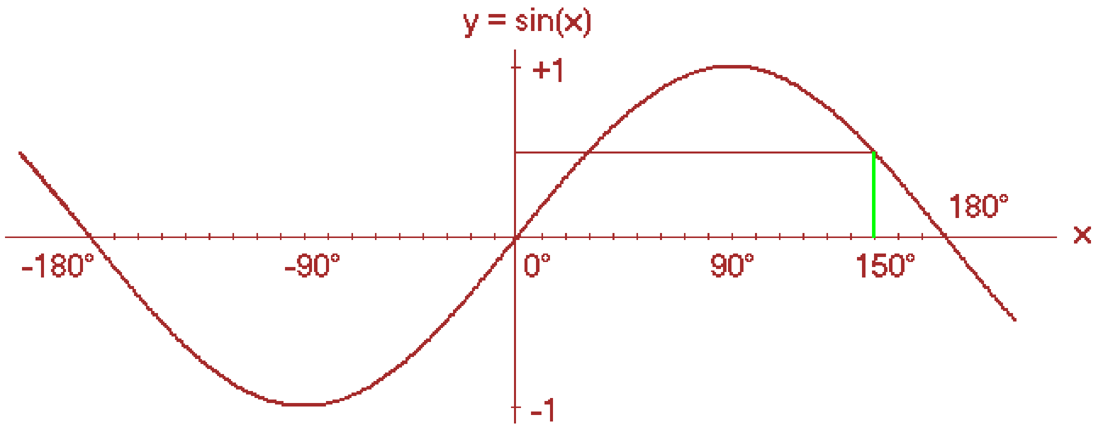
4. ábra: A sinus függvény bevezetése

Az elméleti rész második felében a program kibővíti az értelmezési tartományt 360°-ra, majd annak egész számú többszörösévé negatív és pozitív irányban. Ennek megfelelően az értelmezési tartomány mínusz és plusz végtelen közé esik. A 0° és 360° közötti értelmezési tartományhoz tartozó értékkészlet megismeréséhez sok segítséget nyújt az egységnyi sugarú kör (5. ábra.), amiről leolvasható, hogy az -1 egy és +1 közé esik, a határokat is beleértve. Az egységkör nagy segítséget nyújt abban is, hogy a tanuló a pontos definíció alapján is megértse az értékkészlet kiterjesztését: *egy szög sinusán egységkör estében az egységnyi sugár függőleges tengelyre eső merőleges vetületének hosszát értjük.*



5. ábra: A sinus függvény kiterjesztése

Következő lépés a szögfüggvény karakterisztikájának bemutatása xy koordinátarendszerben, ahol x a független változó és y a függő változó (6. ábra).

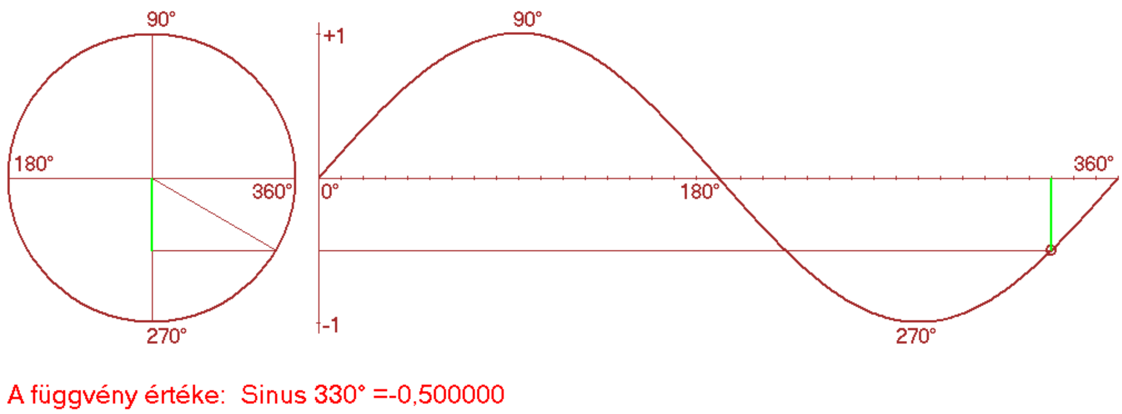


6. ábra: A sinus függvény ábrázolása

Ez az ábra nagyon fontos a pontos megértéshez! Látható, hogy a függvény periódusa 360°, azaz ilyen szakaszonként az értékkészlet ugyanaz. Egyértelműen megjelennek a maximum és minimum helyek 90°-tól indulva 180°-onként, amik +1, illetve -1. Jól mutatja az ábra még a szögfüggvény zérushelyeit is, amik 180° egészszámú többszöröseiként jelennek meg.

3.2.2. A sinus függvény gyakorlati része

A szögfüggvény gyakorlati jelzővel illetett menüpontja egy olyan hagyományos táblát helyettesít, aminek tartalma folyamatosan változik (7. ábra). Szinkronban van egymással az egységkör és az xy koordinátarendszer pillanatnyi állapota. A menüpont indításakor az x változó értéke 0°-tól indulva folyamatosan változik, amit az ábrán jól szemléltet az egységkör sugarának forgása, és a koordinátarendszer y függő változójának változása. A 360° elérése után a x független változó értéke újból felveszi a 0°-ot. Változás közben az x függvényében numerikusan is kiírja a függvény pillanatnyi értékét (pirossal írva).



7. ábra: A sinus függvény és az egységkör

A felhasználó (tanuló) átveheti a programtól a vezérlést a független változó változtatásának területén.

Egér- és billentyűzetfunkciók:	Az függvény értéke mindig leolvasható, ami az egységkör zöld szakaszának felel meg!
	Az egérrel a futókör szabadon vezérelhető
	G: megállít és balra lép 1°-al
	H: megállít és jobbra lép 1°-al
	Space és Enter: elindít vagy irányt vált
	Escape: visszalép a menürendszerbe

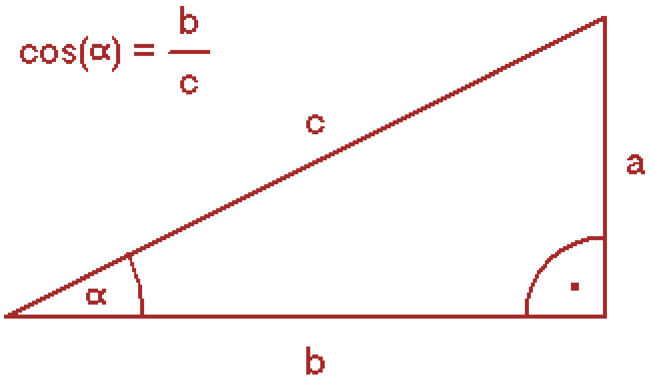
8. ábra: Egér- és billentyűzetfunkciók ismertetése

Módosíthatja a változás irányát, meg is állíthatja azt, sőt fokonként is tudja az x értékét változtatni. A vezérlést az egérrel és néhány funkcióval ellátott billentyűvel teheti meg (8. ábra).

A tanuló a vezérelés segítségével bármilyen szögnek (x) megkeresheti a sinus függvénynek megfelelő értékét (y=sin(x)). A menüpont felhívja a figyelmét a sinus függvény esetében ismert néhány nevezetes szögre, amiknek megfelelő pozicionálással szintén megtalálhatja értékét. Ezek a nevezetes szögek: 0°, 30°, 90°, 150°, 180°, 210°, 270°, 330°, 360° és ugyanezek 360°-os periódusokban.

3.2.3. A cosinus függvény elméleti része

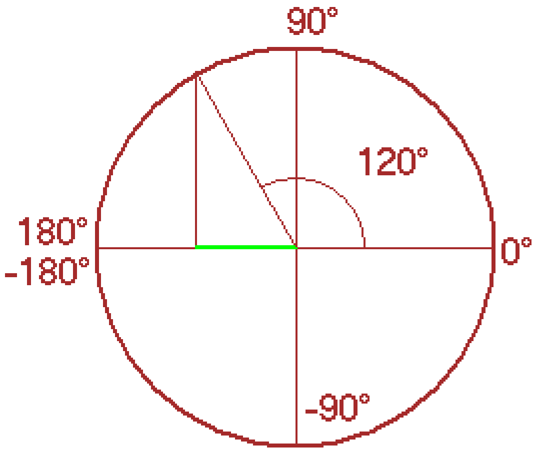
A cosinus függvény sok mindenben hasonlít a sinus függvényre. A program elméleti része előbb itt is a háromszögből származtatja a szögfüggvényt, és ekkor még nem tűnik fel a hasonlóság (9. ábra). A definíciót, miszerint ***egy derékszögű háromszögben egy szög cosinus-át a szög melletti befogó és az átfogó hányadosa adja*** az ábrán látható cos (α) képlet mutatja.



9. ábra: A cosinus függvény bevezetése

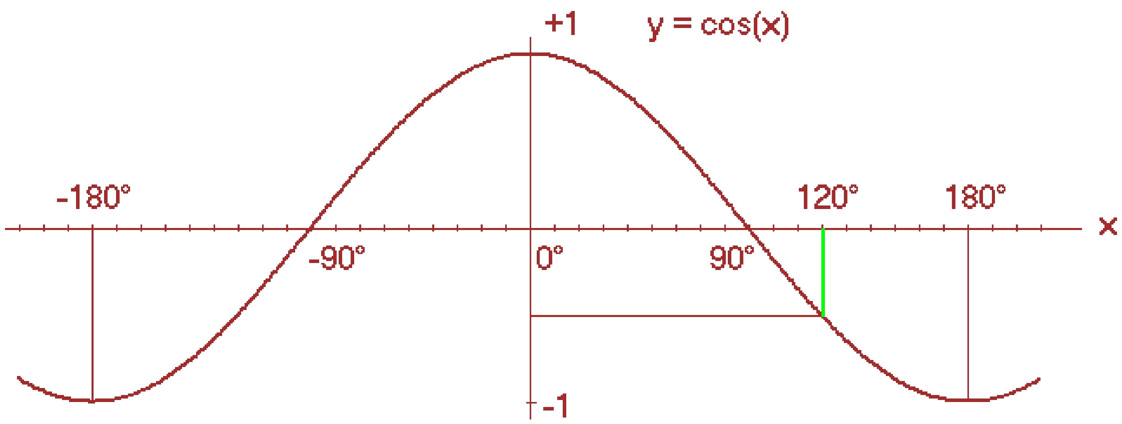
A menüpont második felében itt is kibővül az értelmezési tartomány (eddig 0° és 90° között volt a háromszög esetében) a teljes szög skálára. Megjelenik az egységkör, és ennek megfelelően a függvény definíciója is, miszerint egy szög cosinus-án ***egység sugarú körben a sugár vízszintes tengelyre eső merőleges vetületét értjük***. A sugár méretéből következően adódik, hogy a függvény értékkészlete itt is +1 és -1 közé esik, az intervallum határait is beleveszi.

A 10. ábrán már látható, hogy a sinus és cosinus függvény szoros kapcsolatban áll egymással, hiszen mindkét függvény értékkészletét a vízszintes, illetve függőleges tengelyre eső egységsugár vetülete adja. Mivel mindkét tengely hossza a -1 és +1 közötti távolság, így a két függvény értékkészletének halmaza azonos. A két tengely merőleges egymásra, így az aktuális függvény értékek 90°-al különböznek egymástól. Ha a felhasználó összeveti az 5. és 10. ábra tartalmát, azonnal feltűnik ez a kapcsolat.



10. ábra: A cosinus függvény kiterjesztése

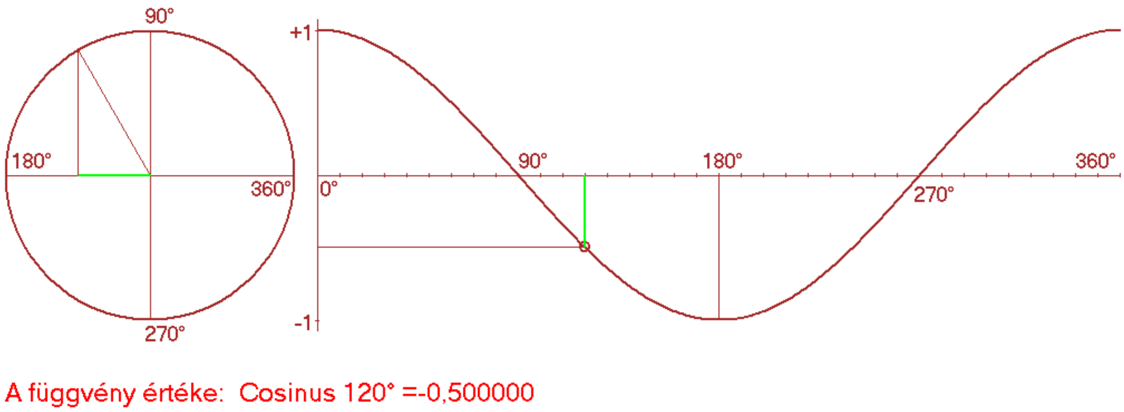
Az cosinus függvény következő ábrája (11. ábra) jól mutatja a függvény karakterisztikáját, és feltűnhet az előző ábránál már jelzett kapcsolat a két szögfüggvény között. Oda-vissza lehet alakítani a függvényeket, ha cosinus-t jobbra toljuk 90°-al, illetve a sinus-t balra. Röviden a függvények 90°-os transzformáltjai egymásnak.



11. ábra: A cosinus függvény ábrázolása

3.2.4. A cosinus függvény gyakorlati része

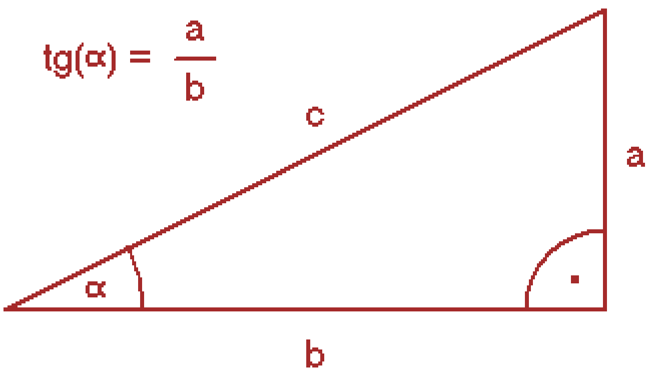
A cosinus függvénynél ugyanazok a gyakorlati lehetőségek adottak a felhasználónak, mint az előző szögfüggvénynél. Vezérelheti a független változó értékét egérrel vagy billentyűzetről, és ennek megfelelően figyelheti a függvény értékének változását. A futás megállításával megkeresheti a függvény nevezetes szögeit és kiértékelheti azokat a pirossal megjelenített függvényértékek viszonylatában. Az oldal felhívja a figyelmét a következő szögek vizsgálatára: 0°, 60°, 90°, 120°, 180°, 240°, 270°, 300°, 360°.



12. ábra: A cosinus függvény és az egységkör

3.2.5. A tangens függvény elméleti része

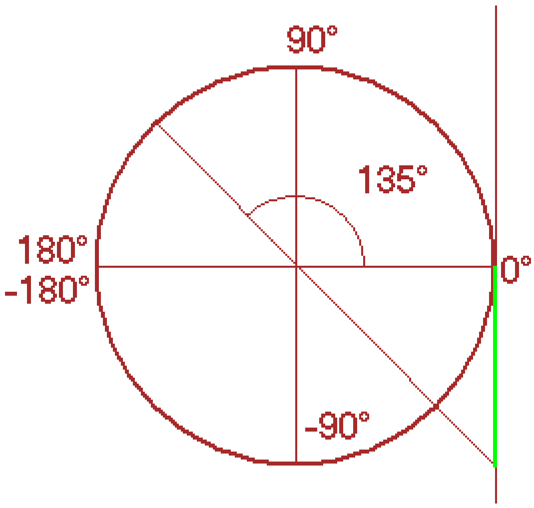
A tangens függvény a háromszögön belül már a két befogó viszonyát tárgyalja (13. ábra), ennek megfelelően a szögfüggvénynek ugyan kapcsolata van a két korábbival, de karakterisztikája és értékkészlete teljesen különbözik azokétól. A definíció ennek megfelelően: ***háromszög esetében egy szögnek a tangensét a szöggel szemközti és melletti befogó hányadosa adja.***



13. ábra: A tangens függvény bevezetése

Háromszögből kiindulva az értelmezési tartomány itt is 0° és 90° közé esik. Ha az α értéke növekszik, úgy növekszik az ’a’ befogó értéke is, s mivel a ’b’ befogó konstans úgy a tg(α) értéke aszimptotikusan tart a végtelenhez. A szögfüggvény e tulajdonsága az ábrázolásnál mutatkozik ki jobban.

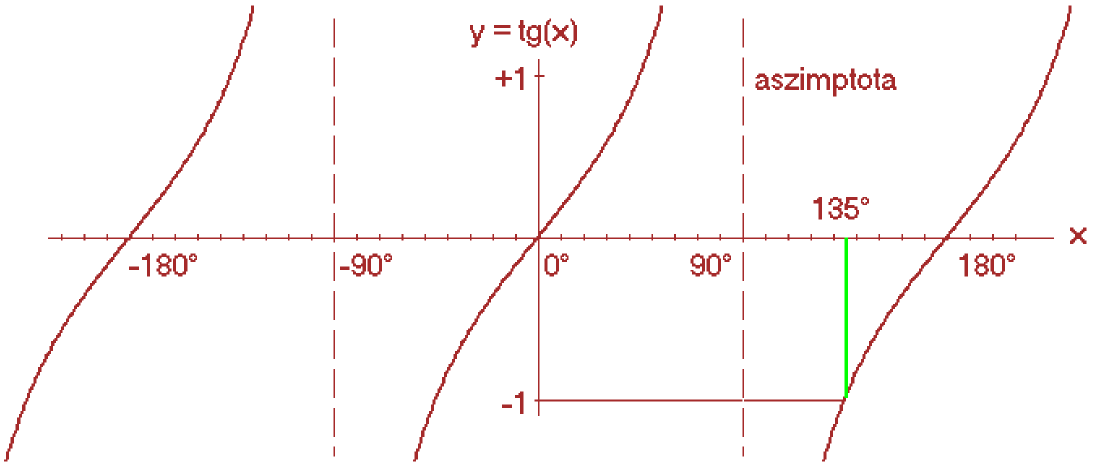
Az értelmezés tartomány kibővítésénél, olyan ábrát kellett elhelyezni az anyagban, ami jól mutatja, hogy, ha az α értéke túl lép a 90°-on vagy negatív irányban halad akkor a szögfüggvény értéke aszimptotikusan a +/- végtelen felé tart és tér onnan vissza. Az egység sugarú kör itt is megfelel a bemutatáshoz, s mivel a szemközti befogó változása (a szög melletti konstans) az, ami befolyásolja a függvény értékét, így itt ***a kör jobb oldali függőleges érintője és a sugár meghosszabbításának metszéspontja jól mutatja a szöghöz tartozó tangens értéket*** (14. ábra: a 135° és -45° tangense a zöld vonal hossza). Az előző mondat egyben a kiterjesztett értékkészletben a tangens függvény definícióját is adja.



14. ábra: A tangens függvény kiterjesztése

Végül az xy koordinátarendszerben egy olyan ábrára volt szükség, ami a függvény minden tulajdonságát, teljes karakterisztikáját jól visszaadja (15. ábra). Ezek a függvény tulajdonságok a következők:

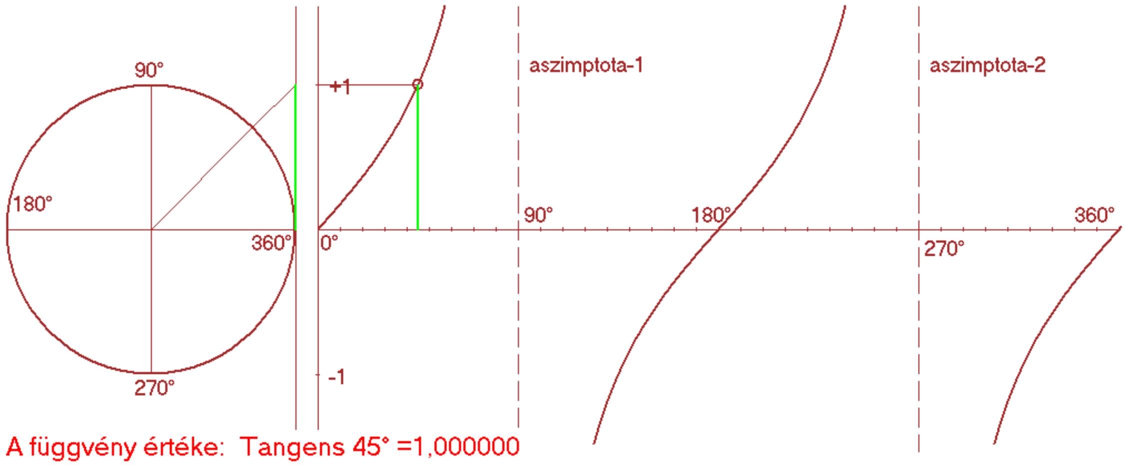
1. 180°-onként a függvény periodikus;
2. 90°-tól kezdve 180°-onként a függvény iránytól függően + vagy – végtelenhez tart aszimptotikusan;
3. az előző pontból következően, ezeken a helyeke aszimptoták jelennek meg, amiket szaggatott vonal jelez;
4. itt is megjelennek (az ábrán a 135°) a nevezetes szögek, amik 0°-tól indulva 45°-onként jelentkeznek. Itt az értékek 0, +1, -1, vagy nem értelmezhető, illetve tart a végtelenhez;



15. ábra: A tangens függvény ábrázolása

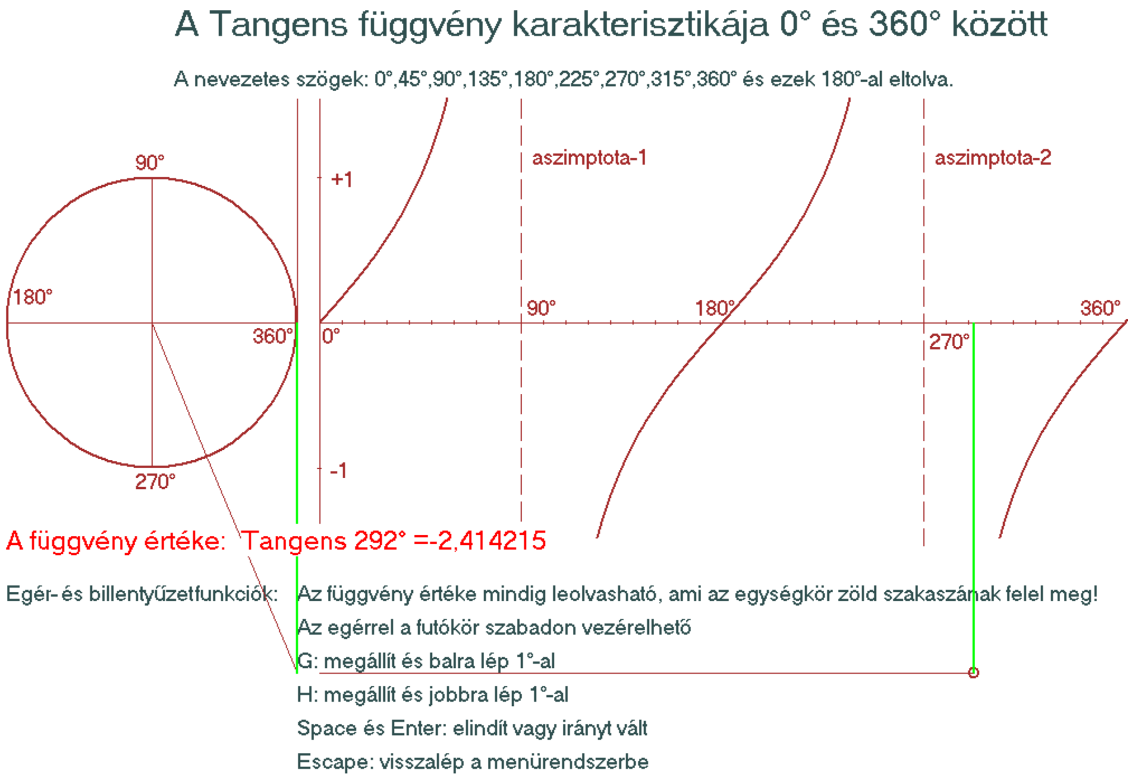
3.2.6. A tangens függvény gyakorlati része

Az előző gyakorlati részektől nem elszakadva célszerű itt is megmutatni azt, hogy a függvény egy 360°-os intervallumban milyen értékkészlettel rendelkezik (16/a. ábra).



16/a. ábra: A tangens függvény és az egységkör

Problémát jelent viszont az, hogy a függvény értéke egy szakaszon elhagyja az ábra területét, belemegy a szöveges részbe, sőt kifut a végtelenbe. A jó megértés érdekében engedtem azt, hogy a felhasználó az engedélyezett vezérlés segítségével kilépjen az ábra területéről, és lássa azt, hogy a függvény értéke tart a +/- végtelen felé (16/b. ábra). Később a vezérléssel vissza is térhet onnan a látható véges területre. Az aszimptoták itt is mutatják azt, hogy a függvény hol nem értelmezhető. Ezeken, a helyeken feltételes utasítások segítségével védhetők ki a tangens függvény problémás nem értelmezhető pontjai, és a függvény értékét egy az ablakon teljesen átívelő zöld vonal helyettesíti. Más esetekben ez a zöld vonal természetesen véges.



16/b. ábra: Kilépés az ábra területéről

4. A HÁROMSZÖG BEMUTATÁSA

4.1. A háromszögek rövid története

Babilon és Sumer már ismerte a Pitagorasz-tételt, egyik táblájukon a racionális oldalú derékszögű háromszögek sorozata látható. Valószínűleg már elő tudtak állítani pitagoraszi számhármasokat is. Már ókori egyiptomi tekercsen téglalap-, háromszög-, trapéz- és körterület-kiszámítási módszerek találhatók. A korabeli India matematikájának központjában megemlíthetjük a majszuri iskolához tartozó Mahavira (580 körül) nevét, aki háromszögekkel és négyszögekkel foglalkozott. Egyik művében matematikai problémákat a Pithagorasz-tétel és a hasonló háromszögek tulajdonságainak segítségével old meg [15].

Az ókori Hellászban Euklidész mellett az ókori matematika nagy alakjai Pithagorasz (Kr.e. 580-500) és Thalész (Kr.e.624-547), akiknek munkássága döntő jelentőségű volt a matematika történetében. Pithagorasz a hagyomány szerint nagy matematikus, neki tulajdonítják a derékszögű háromszög oldalainak hossza közötti összefüggés felfedezését és összefoglalását, az úgynevezett Pitagorasz-tételt, mely szerint a derékszögű háromszög befogóinak négyzetösszege egyenlő az átfogó négyzetével. Thalész milétoszi görög matematikus és csillagász, a görög filozófia atyjaként tartják számon. Tőle származik a róla elnevezett tétel, mely azt mondja ki, hogy ha egy kör átmérőjének két végpontját összekötjük a kör bármely más pontjával, akkor derékszögű háromszöget kapunk.

A középkori Európában a csillagászat a XV. századig semmit sem fejlődött. A görög csillagászat nagyszerű eredményei szinte teljesen feledésbe merültek. A XV. század végén a hajózás nagyarányú megindulása szükségessé tette a tengeren való tájékozódást, a matematika és ezen belül e geometria fejlődését. Sorakoztak a középkori matematikusok, csillagászok, akik a ptolemaioszi geocentrikus világszemlélet végét jelezték: Kopernikusz, Giordano Bruno, Galilei, Kepler. Innen ismerjük a Kepler-háromszöget, aminek lényege az, hogy a szakaszt úgy osztjuk két részre, b-re és c-re, ahol b>c, hogy az a/b=b/c=(1+gyök(5))/2 aránypár teljesüljön. Ha egy derékszögű háromszög átfogójához tartozó magasság így osztja ketté az átfogót, ekkor ezt a háromszöget Kepler- háromszögnek nevezzük. Ismertebb nevén aranymetszésről beszélünk.

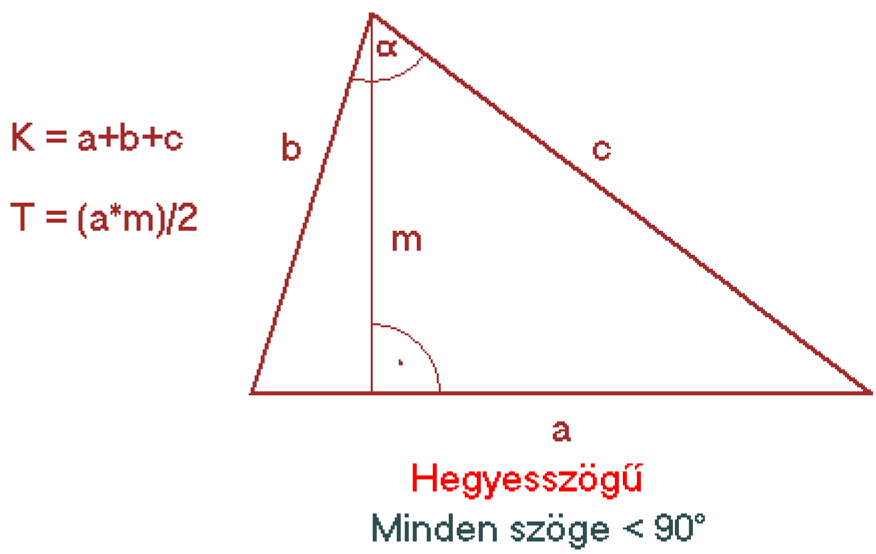
Az újkorban megjelent a háromszögekkel kapcsolatos új információknak az ismertetése egy külön dolgozat elkészítését tételezné fel. Kezdhetnénk rögtön a két Bolyaival, pontosabban a fiatalabbikkal (Bolyai Jánossal), ahol a Bolyai-Lobacsevszkij-féle, úgynevezett hiperbolikus geometriában nem minden háromszög köré írható kör…

4.2. A háromszögek elméleti része

Elvárható, hogy a háromszögeket feldolgozó programrész, minél többet bemutasson a felhasználónak abból, amit a középiskolás tananyag magában foglal. Adja meg a háromszög definícióját (Olyan zárt síkidom, aminek három csúcsa van, és három szakasz határolja.), a háromszögeknél használt alkotó elemek elnevezését, a használt jellemző kifejezéseket és nem utolsósorban a háromoldalú síkidomok fajtáit, fajtánként definiálva. A háromszöghöz kapcsolódó (nem mozgatható elméleti) grafikus felület abból a célból készült, hogy segítse, ezen információk elsajátíttatását [7].

4.2.1. A hegyesszögű általános háromszög

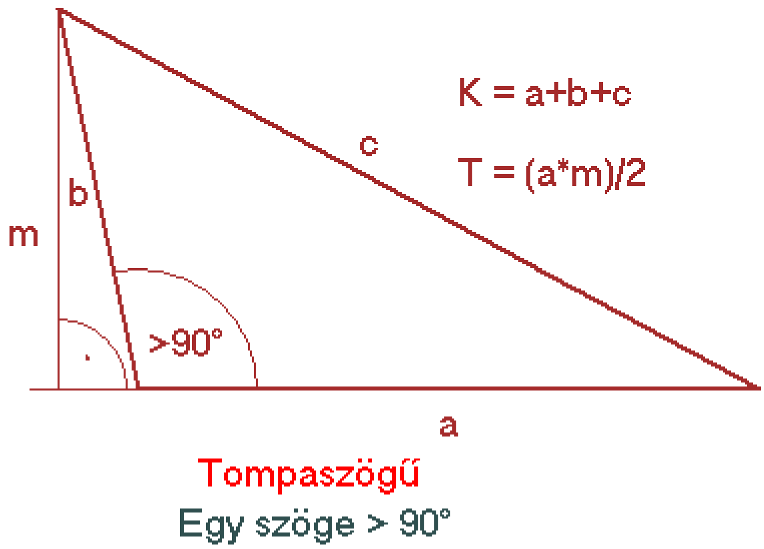
Minden szöge kisebb, mint 90° mondja a definíció erről a síkidomról, a hozzákapcsolódó rajz (17. ábra) megpróbál mindent megmutatni a hegyesszögű általános háromszögről.



17. ábra: A hegyesszögű háromszög és definíciója

4.2.2. A tompaszögű általános háromszög

Az előző és a tompaszögű háromszög megkülönböztetésénél az ábrának mutatni kell azt, hogy a magasságvonal kiesik a zárt területről. Ebből azonnal látható, hogy egyik szöge tompa (18. ábra). Persze ez nem befolyásolja a terület számításánál a magasság használatát. Az „Egy szöge > 90°" utalás definiálja is a háromszöget.

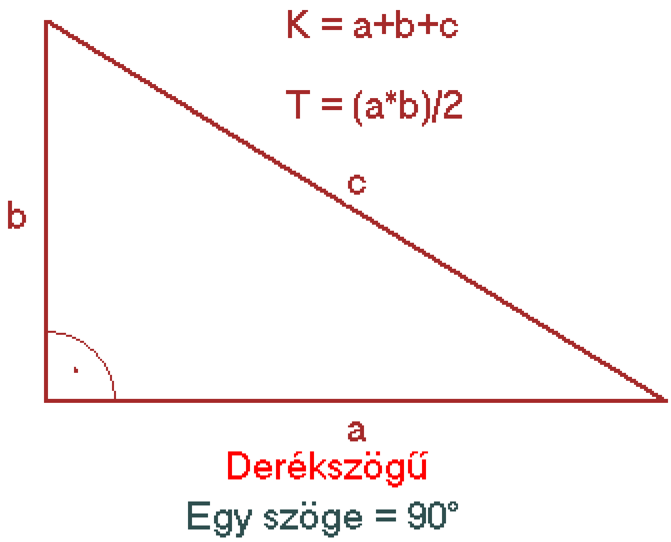


18. ábra: A tompaszögű háromszög és definíciója

4.2.3. A derékszögű háromszög

A háromszögek fajtái közül a legfontosabb. Bármilyen síkidomnál történő számításról legyen is szó, legtöbbször a derékszögű háromszögre vezetünk vissza mindent, mert itt már könnyen számíthatók a keresett értékek, ha jól megismerjük ezt a háromszöget. A számításoknál használhatjuk az előző fejezetben megismert trigonometrikus függvényeket, vagy a még későbbiekben tárgyalt Pitagorasz- és Thalész-tételeket.

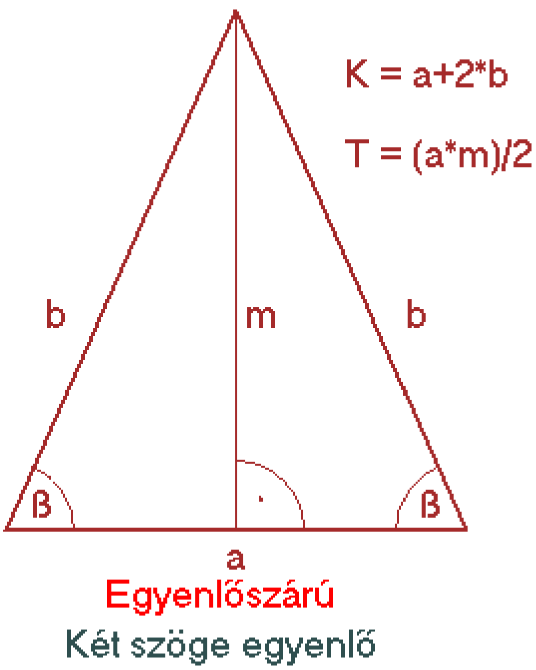
A rövid definícióból (19. ábra), mi szerint a három szög közül az egyik 90°, és a felület felső részén található a szögek összeg 180° információból a felhasználó kiszámíthatja, hogy a másik két szög összege csak 90° lehet. Az előző és a következő háromszögek esetében is adhatók egyszerű feladatok, amik segítik a háromszögek tulajdonságainak megismerését. Persze igazi feladatadási segítséget majd a következő rész a háromszögek gyakorlati része ad.



19. ábra: A derékszögű háromszög és definíciója

4.2.4. Az egyenlőszárú háromszög

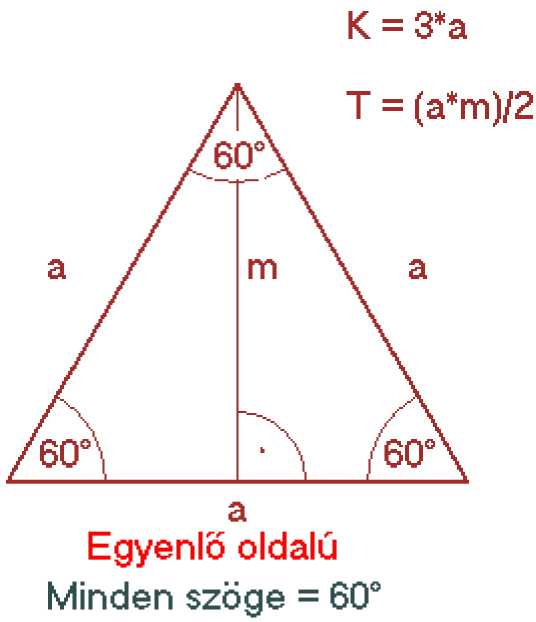
Az egyenlőszárú háromszög megjelenítésénél kiemelendő, hogy tulajdonképpen mi is az egyenlő (20. ábra). Nemcsak a két oldal, hanem a velük szemben lévő két szög is egyenlő, ebből következően az egyik magasságvonal a háromszög szimmetriatengelye lesz. A szimmetriából következően a kerület számítás képlete egy változóval kevesebbet tartalmaz.



20. ábra: Az egyenlőszárú háromszög és definíciója

4.2.5. Az egyenlő oldalú háromszög

Ez háromszög az, ahol olyan ábrát kell készíteni (21. ábra), ami megmutatja a teljes szimmetriát, és ebből következően a teljes egyenlőséget az azonos adottságú részek között (szög, oldal, magasság). A hagyományos terület és kerület összefüggések is egyszerűbbek lesznek, hiszen összesen csak kétféle szakasz hosszal dolgoznak a képletek.



21. ábra: Az egyenlő oldalú háromszög és definíciója

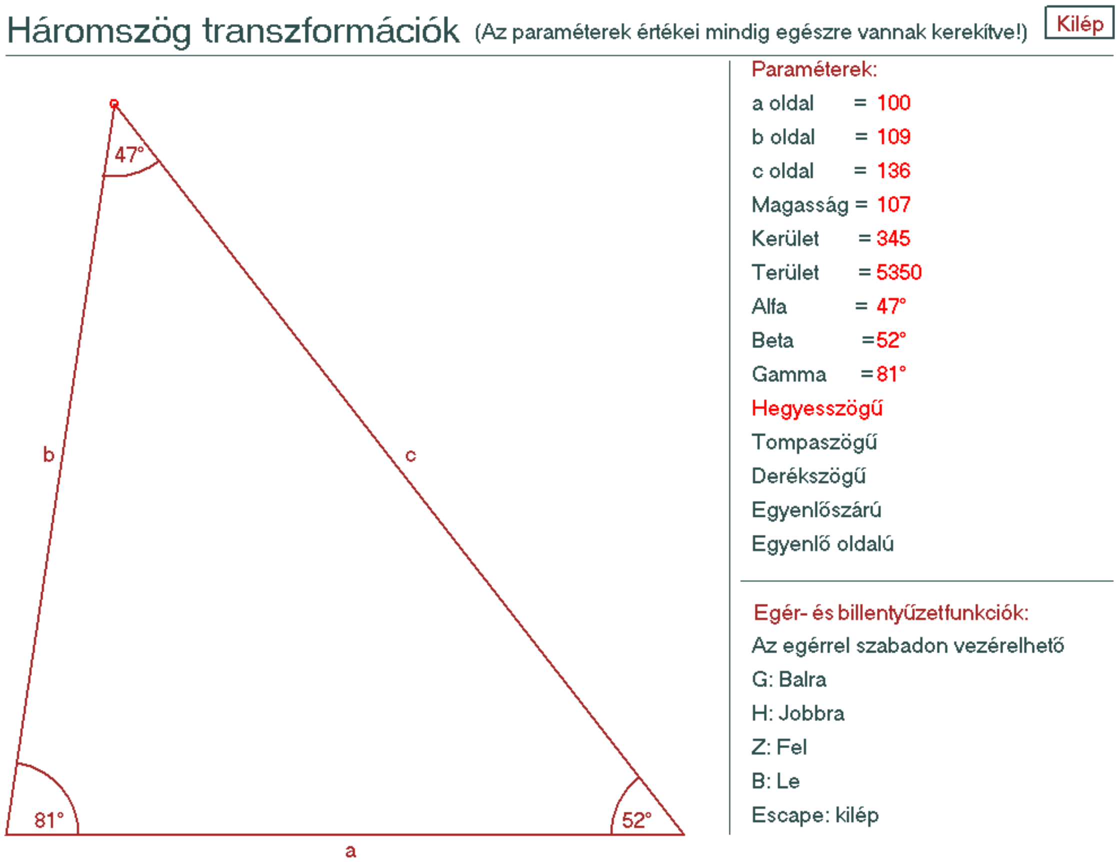
4.3. A háromszögek gyakorlati része

A módszertani oldalt nézve a felület lehetőséget kell, adjon a felhasználónak arra, hogy minél több mindent ki tudjon próbálni a háromszög paraméterinek változtatásával, és a programot használó tanár is tudjon olyan feladatokat adni, amihez szükséges az ehhez tartozó menüpont használata.

A felületen (22. ábra) a jobb felső sarokban levő kilép kapcsoló kivételével csak a háromszög felső csúcsa vezérelhető. Mozgatható egérrel, és finom mozgatáshoz használhatók a G (balra), a H (jobbra), a Z (fel), és a B (le) billentyűk. Ezen egyetlen csúcs mozgatásával minden fajta háromszög előállítható (hegyesszögű, tompaszögű, derékszögű, egyenlőszárú, és egyenlő oldalú). A jobboldali részen a háromszögfajta neve pirosra változik, ha az aktuális háromszög paraméterei kielégítik a fajta tulajdonságait. Ekkor a felhasználó a jobb felső részen lévő paramétereket (a paraméterek természetesen mindig jelzik a háromszög pillanatnyi állapotát) megvizsgálhatja és összekapcsolhatja az éppen piros színű háromszögfajtával. A geometriai törvények ismeretében persze egyszerre többfajta háromszög is piros színre válthat. Például, lehet egy háromszög egyszerre:

– egyenlőszárú és tompaszögű;
– egyenlőszárú és derékszögű;
– egyenlőszárú és hegyesszögű;
– egyenlőszárú, egyenlő oldalú és hegyesszögű;

Ezeket, a tulajdonságokat mind megismerheti a felhasználó.



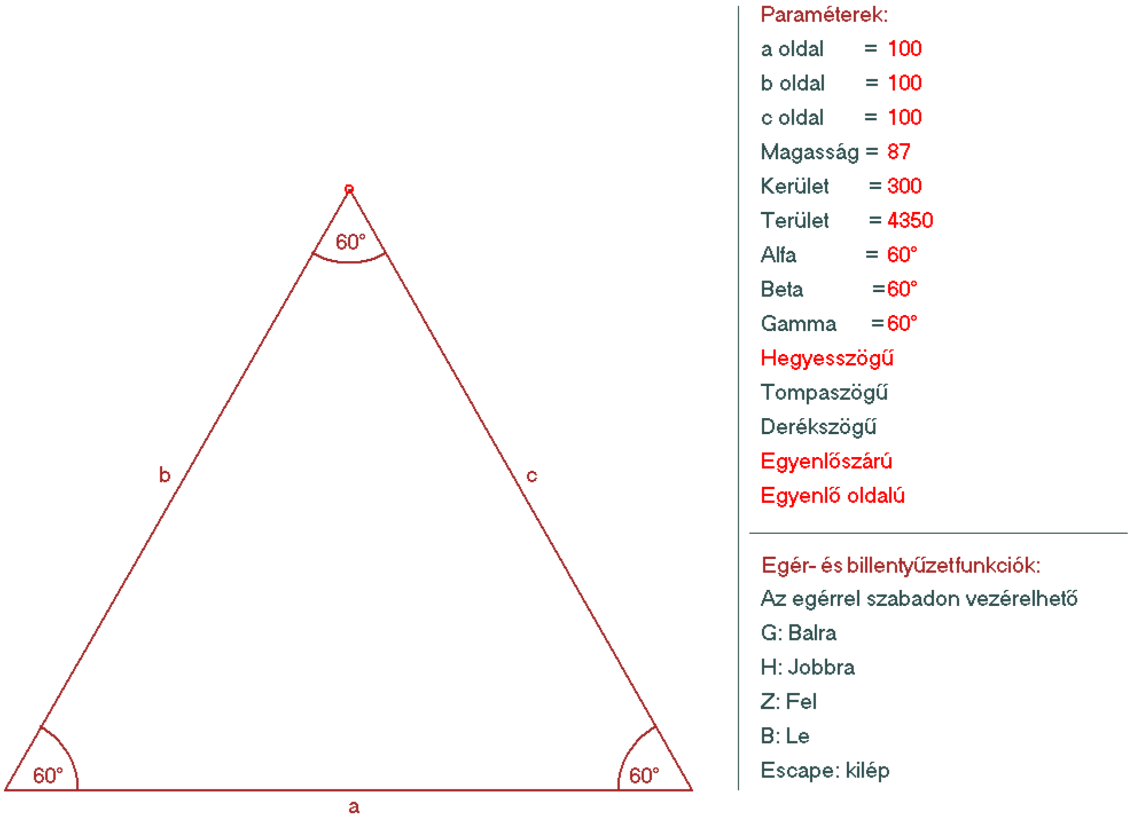
22. ábra: A háromszög felső csúcsának vezérlése

A programozói oldalt vizsgálva nem az algoritmus logikai részének elkészítése jelent problémát (bár a tökéletesség elérése időigényes és nagy precizitást igényel), hanem a háromszögön belüli matematikai törvények sértetlenségének elérése. Kérdés: miért, hiszen csak a Sinus- és a Cosinus-tétel (egyszerűbb eset: Pitagorasz-t.), előírásait kell hiba nélkül betartani. A probléma az, hogy látványosság kedvéért csak egész értékeket célszerű megjeleníteni, mert így rögtön láthatóak a háromszögfajták jellemzői. Egy 5 vagy többjegyű tizedes tört esetén nehéz elérni egér- vagy billentyűnavigálással, hogy a háromszög derékszögű, egyenlőszárú, vagy egyenlő oldalú legyen. Így minden paraméter egészre van kerekítve, és némi csalással a törvények továbbra is élnek. Ez a kerekítés az igazság kedvéért a felület címének zárójeles részében természetesen jelezve van. (22. ábra: ***A paraméterek értékei mindig egészre vannak kerekítve!***) Remélhetőleg ez nem rontja a háromszög megismerésének tökéletességét.

Ennek megfelelően tesztelhető, hogy ha a felületet beállítjuk úgy, hogy minden szög 60° legyen (23. ábra), akkor ennek megfelelően minden oldalt 100 egységnek mutat. Ekkor leolvashatjuk az aktuális magasságot, amit 87 egységnek jelez. Ilyen háromszög nem létezik!

Ugyanis az egyenlő oldalú háromszög esetében áll a következő magasság törvény: $m = a * \frac{\sqrt{3}}{2}$, ahol „a” a háromszög oldala.

A vizsgált esetben a = 100, így $m = 100 * \frac{\sqrt{3}}{2} = 86.60254037844...$ Ez az érték jól közelíti az ábrán található Magasság = 87 egységet!



23. ábra: Az egyenlő oldalú háromszög beállítása

5. NÉGYSZÖGEK BEMUTATÁSA

5.1. Néhány szó a négyszögekről

A kézzelfogható matematika a halmazokat bezárólag értelmezi. Emiatt egy négyzetről például, elmondhatjuk, hogy egyben téglalap, rombusz, stb.. Szépterület a négyszögek halmaza, amit a 24. ábra szemléletesen megmutat. Ennek segítségével számozva és definiálva sorolhatók fel a négyszögek fajtái, ahol az egy-egy halmazba eső fajták száma egyre bővül [16].

Definíció: A geometriában olyan síkidom, amelynek négy oldala és négy csúcsa van, és a belső szögeinek összege 360°.

1. **Négyzet:** mind a négy oldal egyenlő hosszúságú, és minden szöge derékszög.
2. **Téglalap:** minden szöge derékszögű. Ebből az is következik, hogy a szemközti oldalak párhuzamosak és páronként egyenlő hosszúak. (A halmaz elemei: négyzet, téglalap.)
3. **Húrtrapéz** vagy egyenlőszárú trapéz: két szemközti oldala párhuzamos, és a másik két nem párhuzamos oldal egyenlő hosszúságú, vagy létezik egy kör, ami minden csúcsán átmegy. (A halmaz elemei: négyzet, téglalap, húrtrapéz.)
4. **Húrdeltoid:** két-két egymás melletti oldal azonos hosszúságú és a négy csúcspont köré kör írható. (A halmaz elemei: négyzet, húrdeltoid.)
5. **Bicentrikus négyszög:** egyszerre húr- és érintőnégyszög. (A halmaz elemei: négyzet, húrdeltoid, bicentrikus négyszög.)
6. **Rombusz:** mind a négy oldal egyenlő hosszúságú. (A halmaz elemei: négyzet, rombusz.)
7. **Deltoid:** két-két egymás melletti oldal azonos hosszúságú. (A halmaz elemei: négyzet, rombusz, húrdeltoid**,** deltoid.)
8. **Paralelogramma:** a két-két szemközti oldal párhuzamos. (A halmaz elemei: négyzet, téglalap, rombusz, paralelogramma.)
9. **Érintőnégyszög:** minden oldala ugyanannak a beírt körnek az érintője. (A halmaz elemei: négyzet, rombusz, deltoid, húrdeltoid, bicentrikus négyszög, érintőnégyszög.)

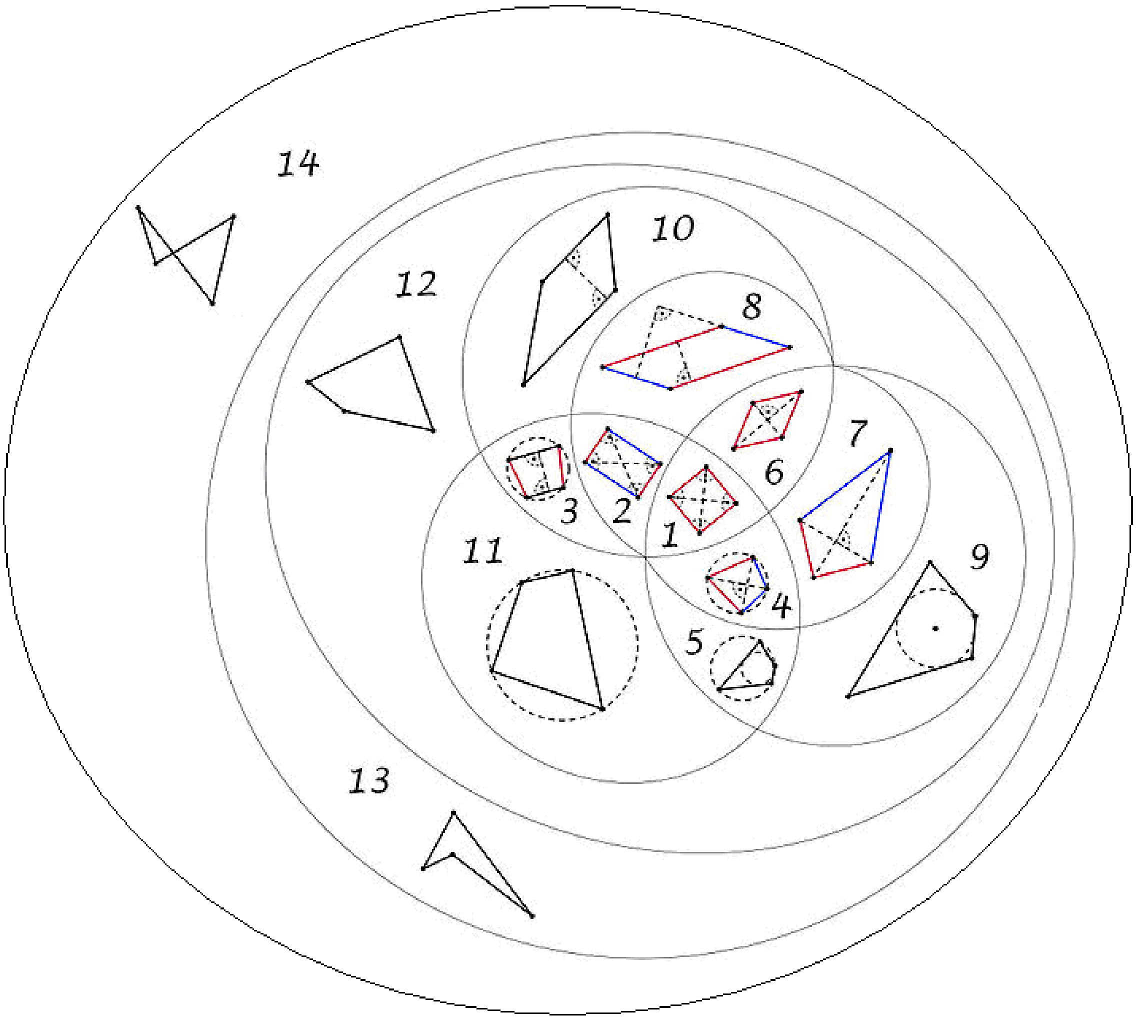
10. **Trapéz:** legalább két szemközti oldala párhuzamos. (A halmaz elemei: négyzet, téglalap, rombusz, paralelogramma, húrtrapéz, trapéz.)

11. **Húrnégyszög:** a négy csúcspont köré kör írható. (A halmaz elemei: négyzet, téglalap, húrtrapéz, húrdeltoid, bicentrikus négyszög, húrnégyszög.)

12. **Konvex négyszög:** szögei kisebbek, mint 180°. (A halmaz elemei: négyzet, téglalap, húrtrapéz, rombusz, paralelogramma, trapéz, húrdeltoid, deltoid, bicentrikus négyszög, húrnégyszög, érintőnégyszög, konvex négyszög.)

13. **Egyszerű négyszög** (önmagát nem metsző): a szemközti oldalak nem metszik egymást (A halmaz elemei: minden konvex négyszög és a konkáv négyszög.) **Konkáv négyszög:** Egy szöge nagyobb, mint 180°.

14. **Négyszög:** négy oldala és négy csúcsa van. (A halmaz elemei: minden konvex négyszög, konkáv négyszög, elfajult négyszög.) **Elfajult négyszög** (önmagát metsző): a szemközti oldalak metszik egymást.



24. ábra: A négyszögek halmaza [16]

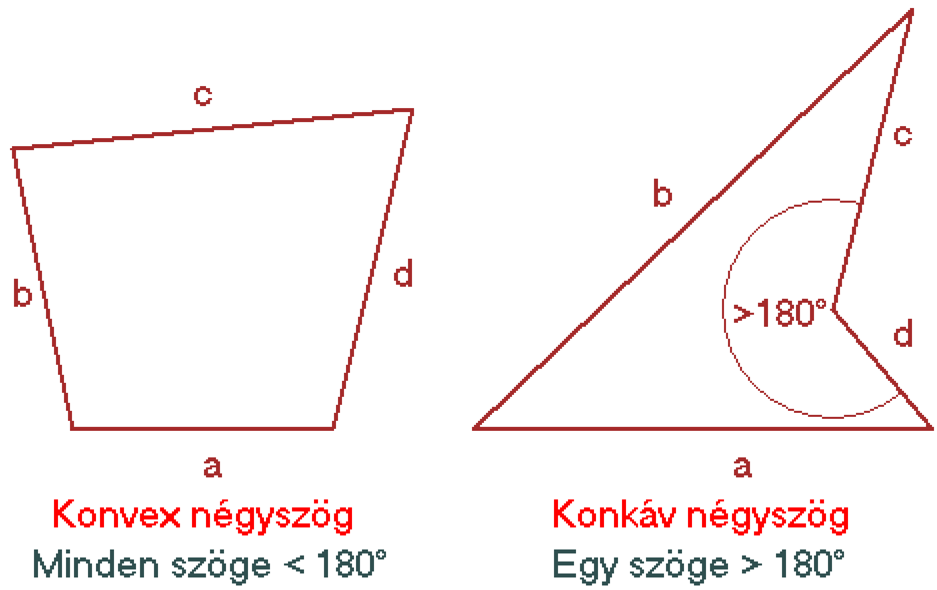
5.2. A négyszögek elméleti része

A program nem mutatja be teljesen az előző 5.1-es fejezetben tárgyalt négyszögek halmazát, de azokat, amik a középiskolás tananyag részei, feldolgozza és kiemeli a fontosabb tulajdonságait [7]. A 14 fajtából 10-et jellemez a statikus elméleti részben. A kimaradt fajták az érintő, a húr, a bicenrikus, és az elfajult négyszög. A tárgyalt fajták pedig a következők:

5.2.1. A konvex és a konkáv négyszög

Felvetődött a kérdés, hogy milyen sorrendben legyenek ábrázolva a négyszögek. Kezdődjön a tökéletessel (négyzet), és a bonyolultság szintjétől függően sorban, vagy a teljesen általánossal és úgy sorban. Célszerűnek tűnik, hogy részben így, részben úgy.

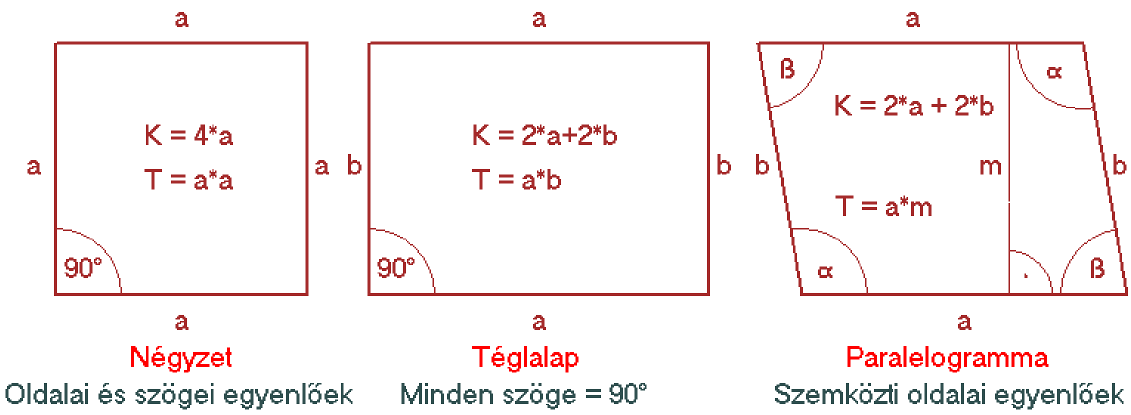
Először ismerjük meg az általános négyszög definícióját és jellemzőit, majd folytassuk a négyzettel, téglalappal és így tovább. Tehát az induló két síkidom a konvex és konkáv négyszög, amit a 25. ábra mutat, és egyben a szögek viszonylatában definiálja is mindkettőt.



25. ábra: A konvex és konkáv négyszög definícióval

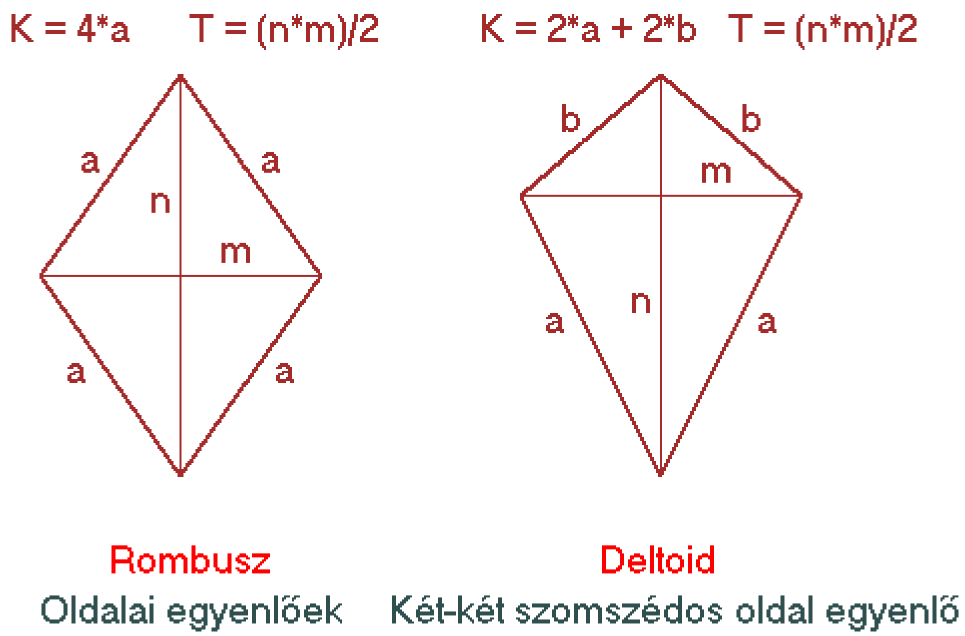
5.2.2. A négyzet, a téglalap és a paralelogramma

A négyzet maga a tökéletesség, és a konkáv és az elfajult négyszög kivételével az összes többi származtatható belőle. Célszerű a fokozatosságot betartani, és így a definíciókból is láttatva sokkal jobban meg lehet érteni, ahogy bővül a négyszögek halmaza. Például a négyzetnél még minden paraméter kötött, a téglalapnál már az oldalak párban változhatnak, de a szögek még stabilak, a paralelogramma már a szögek változását is engedi, de csak szemközti párban. A 26. ábra mutatja a változásokat.



26. ábra: A négyzet, a téglalap és a paralelogramma definícióval

5.2.3. A rombusz és a deltoid

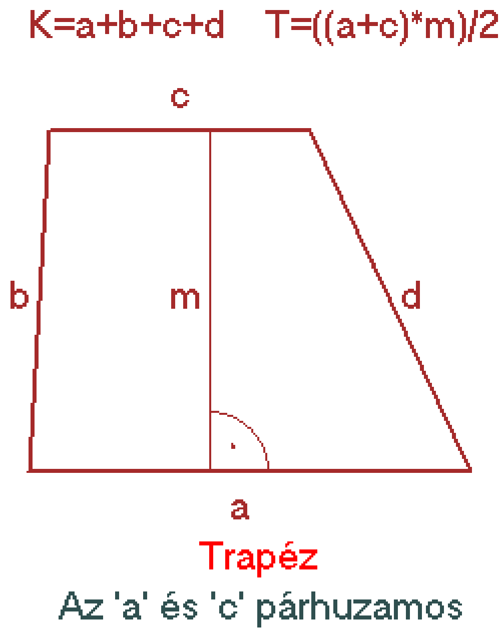


27. ábra: A rombusz és a deltoid definícióval

Adódik egy elágazási lehetőség a halmazok sorrendjében, ahol lehet folyatni a bemutatást a trapézok, vagy pedig deltoidok irányában. A rombusz nem kérdés, hiszen ez a síkidom még nagyon közel áll a paralelogrammán és négyzeten keresztül tökéletességhez. A deltoidok és a trapézok közül, a trapéz már közel sincs olyan tökéletes, mint a deltoid, mivel csak egy megkötéssel rendelkezik (egy pár szemközti oldal párhuzamos), addig, amíg a deltoid kettővel (egyik és másik pár szomszédos oldalak is egyenlők). Az sem utolsó, hogy a deltoid (27. ábra) talán könnyebben származtatható a rombuszból, mint a trapéz, és így könnyebben értelmezhetőek a tulajdonságai.

A deltoidnak van egy érdekes tulajdonsága a többi, valamilyen szempontból szabályos négyszöghöz képest. Neki ugyanis az egyik szöge akár 180°nál nagyobb is lehet, azaz megszületik a konkáv deltoid. Az angolszáz országok nagy része csak a konvex deltoidot tekinti deltoidnak. Ettől függetlenül nálunk létezik a konkáv deltoid, amit az oktatóprogram dinamikus felülete meg is jelenít megfelelő beállítás mellett.

5.2.4. A trapéz



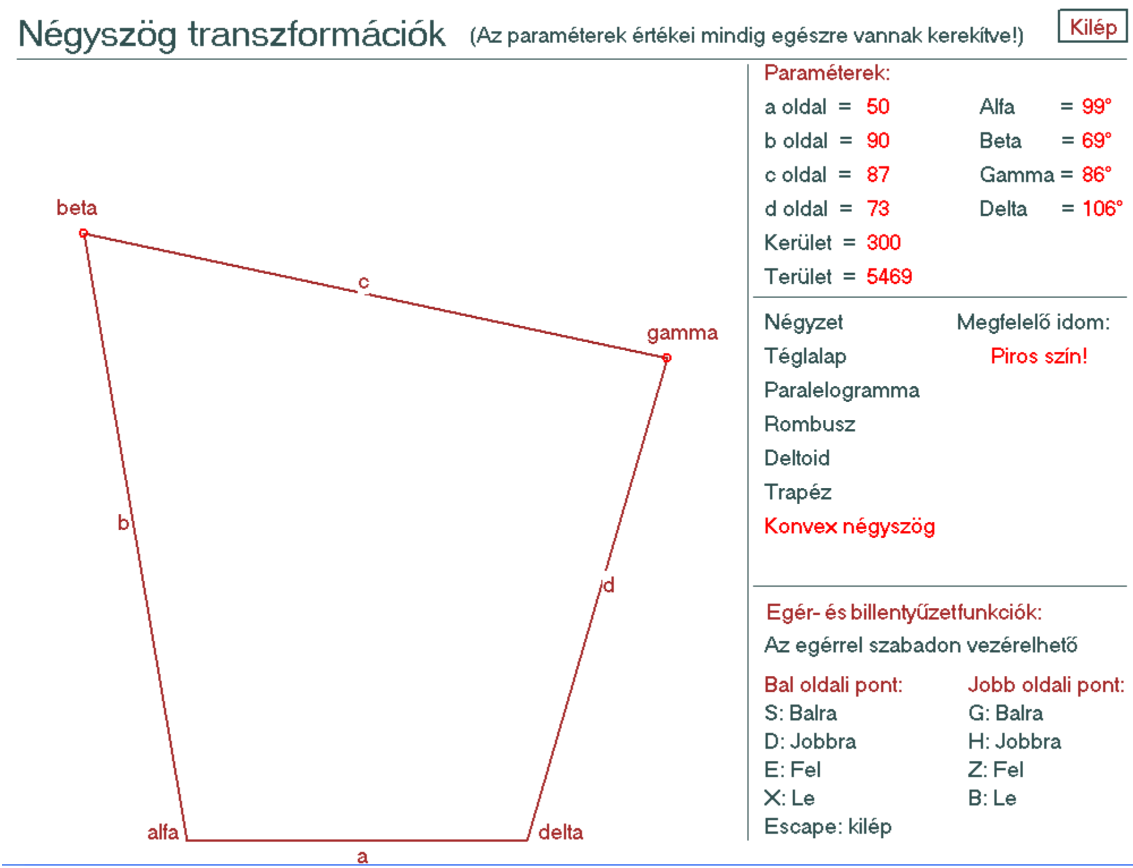
28. ábra: A trapéz ábrázolása

A bemutató elméleti oldalán a trapézt nem véletlenül helyeztem a paralelogramma alá, hiszen a paralelogrammánál van egy halmazelágazás a rombusz, illetve a trapéz felé. Az egyik irányban tökéletesedik a négyszög a másik irányban, pedig tökéletlenedik. A tökéletlenség a trapéz irányában jelentkezik, itt ugyanis már csak az egyik szemközti oldal pár párhuzamos. Remélhetőleg a felhasználónak feltűnik ez a lépcső a tökéletlenedés irányában a paralelogramma és a trapéz között (28. ábra).

A trapéz egyéb fajtáit, ahol a definícióban egyéb megkötések jelentkeznek, a következő alfejezet dinamikus felülete mutatja meg. Ezek a megkötések a derékszögű és az egyenlőszárú trapézokat határozzák meg.

5.3. A négyszögek gyakorlati része

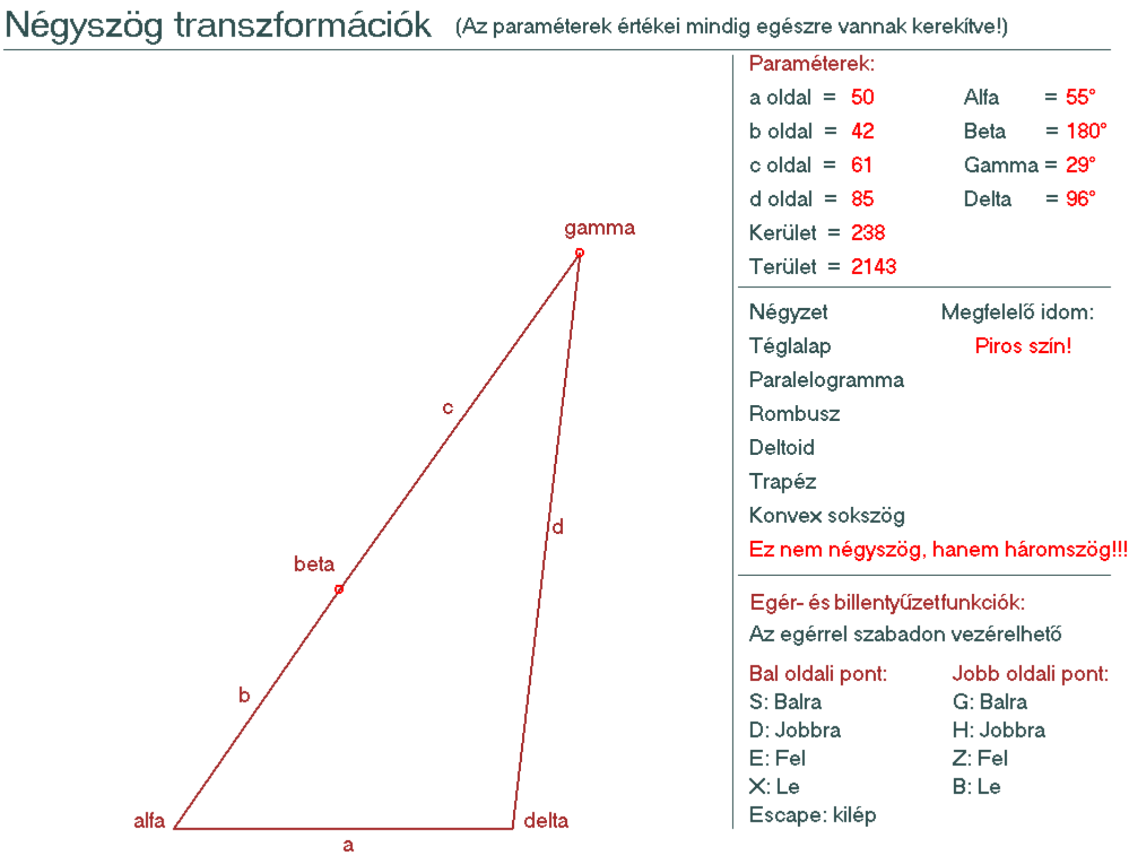
Síkgeometriailag fontos hogy négyszögcsúcsok egérrel és billentyűzetről vezérelt változásai milyen hatással vannak a négyszög fajtáira. Módszertanilag legalább annyira fontos, hogy ezt a fajtaváltozást jelezze is a felhasználó felé. A négyszögtípusok szürke színnel vannak felsorolva a dinamikus felületen (29. ábra), és ha valamelyik négyszöggel kapcsolatban az őt definiáló kritérium teljesül, úgy piros színre vált. A négyszög számszerű paraméteri mindig követik a változást és visszajelzik azt a felületen, így a felhasználó ellenőrizheti, megtanulhatja az aktuális négyszöghöz tartozó definiált tulajdonságokat.



29. ábra: A négyszög két csúcsának vezérlése

A felületen nem jelenik meg szürkével a konkáv deltoid, a derékszögű trapéz és a konkáv négyszög kifejezés, ellenben ha e négyszögek kritériumai fennállnak úgy a deltoid, trapéz és konvex négyszög kifejezések helyett ezek jelenek meg pirossal. Így a numerikus tulajdonságok kapcsolhatók hozzájuk.

Programozási problémát jelent, hogy hogyan oldható meg az elfajult négyszög (önmagát metsző, azaz a szemközti oldalak metszik egymást) problémájának elkerülése. Részben nem is középiskolás anyag, másrészt bizonyos területeken nem is tekintik négyszögnek. A négyszög két csúcsa stabil, kettő vezérelhető a korábban leírtaknak megfelelően, ebből következően, ha a két csúcspont vízszintes és függőleges irányú koordinátájának különbsége előjelet vált létrejön az elfajult négyszög. A függőleges irányban megfelezett terület, e problémát megoldja, úgy hogy a két csúcspont külön területen mozog, és nem tud találkozni egymással. Csak egy egységnyire tudják egymást megközelíteni, ami a ’c’ oldal hosszát adja vissza. Ettől a megkötéstől függetlenül az elfajult négyszög kivételével minden fajta négyszög előállítható.



30. ábra: A háromszög helyzet elkerülése

Másik programozási probléma a háromszög elkerülése, ugyanis ha csúcsmozgatás közben a négyszög egyik szöge 180°, az már nem négyszög, hanem háromszög. Ezt a problémát egy kétirányú elágazás könnyen ki tudja kerülni, ahol a négyszög szögeit vizsgálja ’vagy’ operátorral összekapcsolva. A vizsgálat ’igaz’ eredmény esetén minden fajta négyszög nevét szürkére állítja, és az alsó sorban megjelenik az „Ez nem négyszög, hanem háromszög” figyelmeztetés pirossal (30. ábra).

Az utolsósorban a felület jobb alsó részén még megjelennek a vezérlés lehetőségei egérrel és billentyűzetről (29-30. ábra). A billentyűkombinációk a finom navigálást segítik, bár megfelelő kézügyesség mellett ez egérrel is megoldható. Sok esetben szüksége lehet a felhasználónak a finom navigálásra, ha egy négyszög kritériumait pontosan be akarja állítani.

6. PITAGORASZ- ÉS THALÉSZ-TÉTEL

6.1. A két tétel kimondásának eredete

6.1.1. A Pitagorasz-tétel

A tétel kimondása és bizonyítása egyes vélemények szerint Püthagorasz közösségéhez köthető. Ennél biztosabbat sajnos nem lehet állítani, mivel a krotóni iskola belső körének tagjai nem tartották fontosnak, hogy gondolataikat megörökítsék. A fennmaradt adatok néhány lelkes kortárs munkájának köszönhetők, akik lejegyezték a mester – vagy éppen a tanítványok – elmélkedéseit. Ehhez azonban nem állt még rendelkezésükre olyan komoly geometriai apparátus, mint amit Eukleidész 200 évvel később, az időszámításunk előtti 3. században megalkotott, ezért csak a szemléletre és megérzéseikre támaszkodhattak. A sejtéshez minden bizonnyal síkidomok átdarabolásával jutottak el. A mai napig a Püthagorasz nevét őrző összefüggés szerint egy derékszögű háromszög befogóinak négyzetösszege megegyezik az átfogó négyzetével [17].

Ennek néhány speciális esetét az ókori Babilon területén talált régészeti leletek tanúsága szerint már az időszámításuk előtti második évezredben ismerték. Egyes kutatások szerint a ma ismert legősibb számelméleti dokumentumon, az időszámításunk előtti 1600 és 1900 között Babilonban készült Plimpton 322 táblán található pitagoraszi számhármasok már a fenti képlet alapján készültek. Ettől függetlenül mások ezt a képletet Püthagorasznak és követőinek tulajdonítják [18].

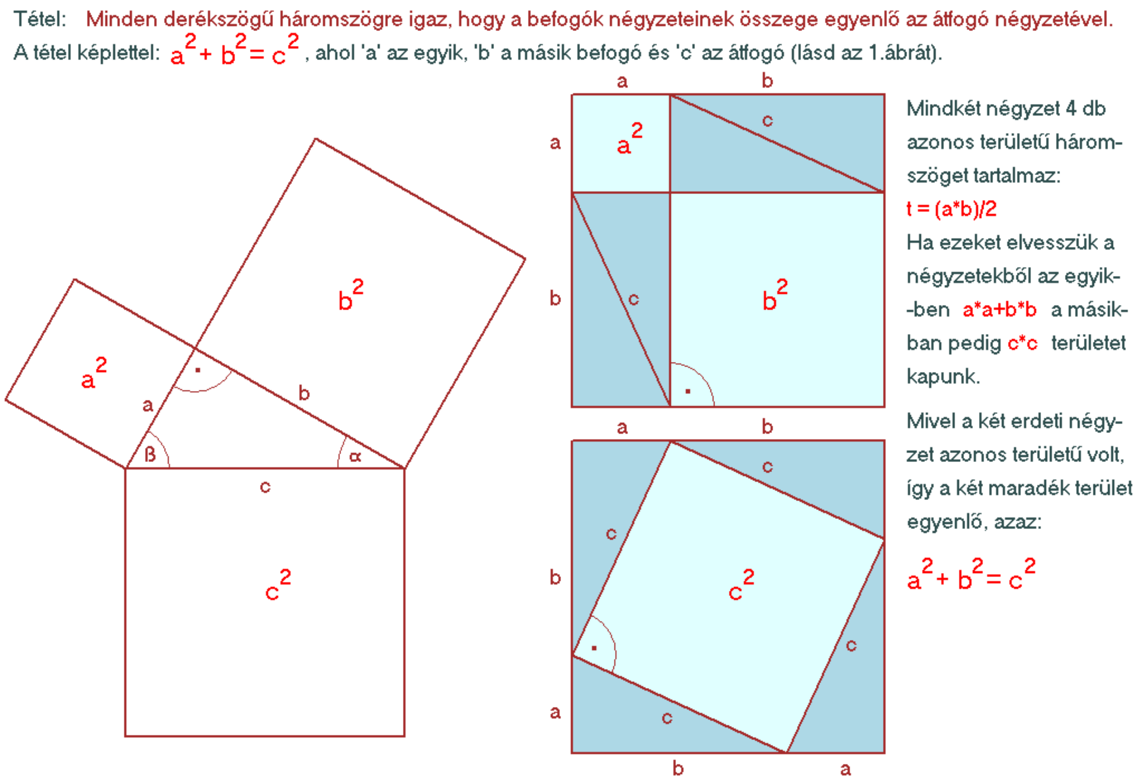
6.1.2. A Thalész-tétel

Thalész Krisztus előtt VII. században (640 táján) született a kis-ázsiai Milétoszban. Szülei révén génjeiben két kultúra, a keleti és a nyugati (görög) találkozott. Felnövekedvén útnak indult, s a hosszú utazás során járt Egyiptomban s Közel-Kelet számos vidékén. Alapvető tudását és műveltségét a kaldeus és egyiptomi papoktól szerezte [19].

Proklosz szerint Thalész bizonyította először - mégpedig egybevágóságokkal -, hogy a háromszög belső szögeinek összege 180°. Ismerte a szög fogalmát. Bebizonyította, hogy az egyenlőszárú háromszög alapon fekvő szögei egyenlők, és megmutatta, hogy két háromszög egybevágó, ha egy oldaluk és a rajta fekvő két szögük egyenlő. Geometriai állításokat ő kezdett először bizonyítani, s őt tekintik a görög matematikai fejlődés megindítójának. Elsőnek rajzolt a körbe derékszögű háromszöget. Felismerte, hogy ha egy háromszög minden csúcsa egy kör ívén található és az egyik szöge derékszög, akkor a háromszög derékszöggel szemközti oldala a kör átmérője. Ezek után a Thalész-tétel őrzi nevét.

6.2. A Pitagorasz-tétel bemutatása

Tervezéskor a tétel kimondása és bizonyítása két menüpontban, két felületen lett elhelyezve, később méretét tekintve egy menüpontban is el lehetett helyezni (31. ábra). A kimondást a könnyebb érthetőség kedvéért, minél rövidebben, de annál pontosabban kell megoldani. A kimondás három formában jelenik meg a felületen. Ezek a szöveges, a képletes valamint az ábra (a felület bal oldala). Nagyon fontos, hogy a három változat változók, és elnevezések tekintetében szinkronban legyen egymással, és kölcsönösen utaljanak is egymásra. (Ez esetben a képlet változói megjelennek az ábrán, elnevezései pedig a szöveges definícióban.)



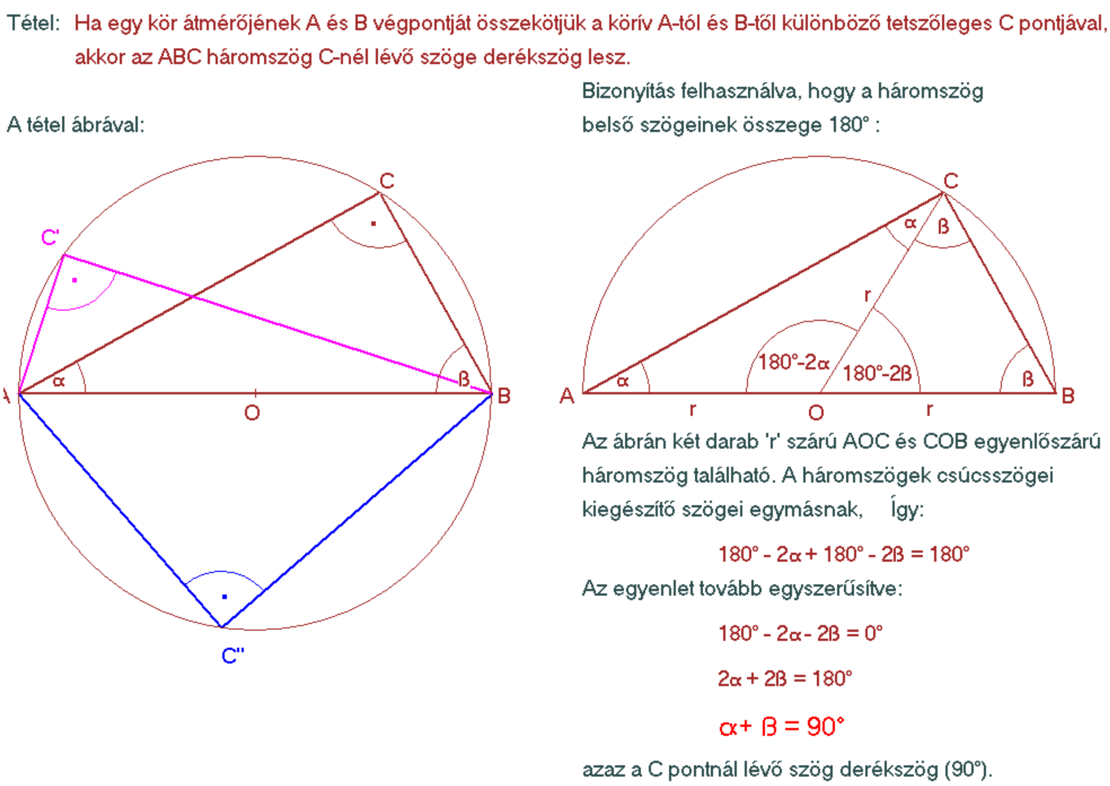
31. ábra: A Pitagorasz-tétel és bizonyítása

A Pitagorasz-tételt általánosságban valószínűleg a krotóni iskolában mondták ki és próbálták igazolni. A sejtéshez minden bizonnyal síkidomok átdarabolásával jutottak el. A bizonyításhoz a felület jobb oldalán (31. ábra) e módszer egyik legegyszerűbb alkalmazását láthatjuk. Természetesen az ábrához szöveges és képletes magyarázat is tartozik, ami szinkronban van mindkét ábra jelöléseivel.

6.3. A Thalész-tétel elméleti része

A Thalész-tétel bemutatásának módszere az előző alfejezet technikáját követei. A tétel kimondása csak szövegesen történik, amihez a felület (32. ábra) bal oldali ábrája kapcsolódik. Az ábrán lévő kör, átmérőjén több háromszög is el lett helyezve különböző színekkel, hogy ez is nyomatékosítsa azt, hogy a tétel a kör minden ilyen tulajdonságokkal rendelkező háromszögére igaz!

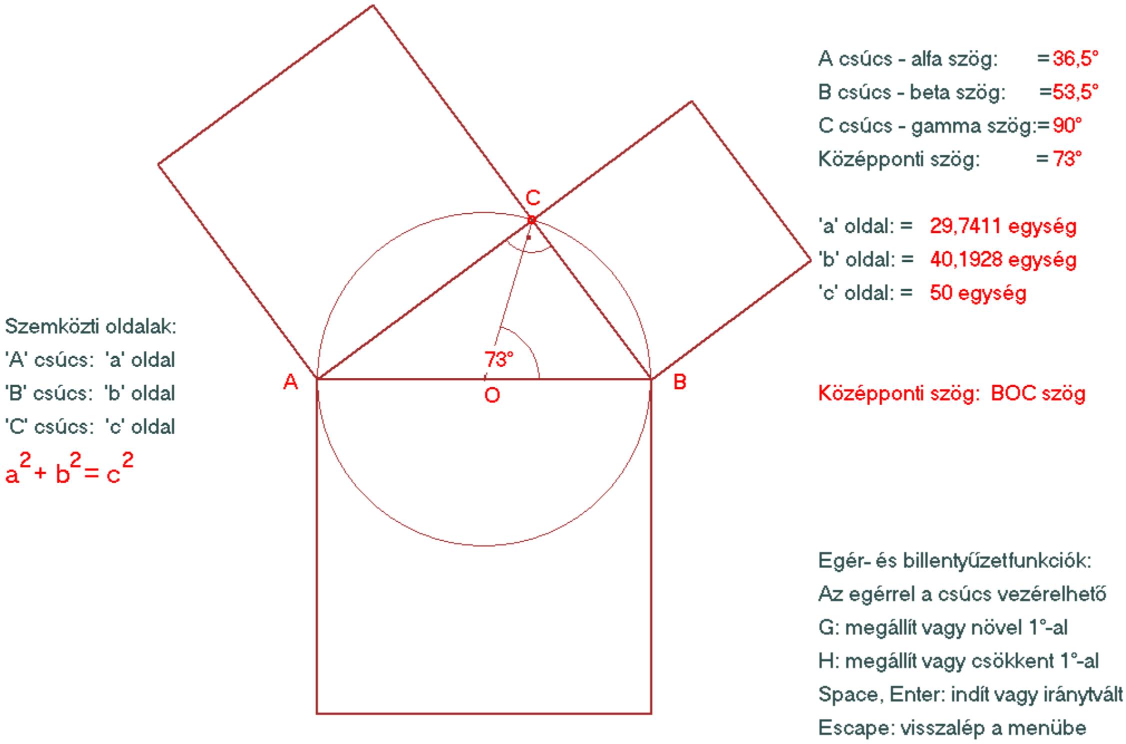
A igazolás itt is a felület jobb oldalán jelenik meg. A Thalész-tétel sok módszerrel bizonyítható. Ebben az esetben fontos, (mivel a szoftver középiskolás használatra készül) hogy minél beszédesebb ábra jelenjen meg a felületen és az abból kiemelt lineáris egyenlet minél egyszerűbb és könnyen megoldható legyen. Természetesen a két ábra jelölésrendszere itt is szinkronban van egymással.



32. ábra: A Thalész-tétel és bizonyítása

6.4. A két tétel gyakorlati része egy felületen

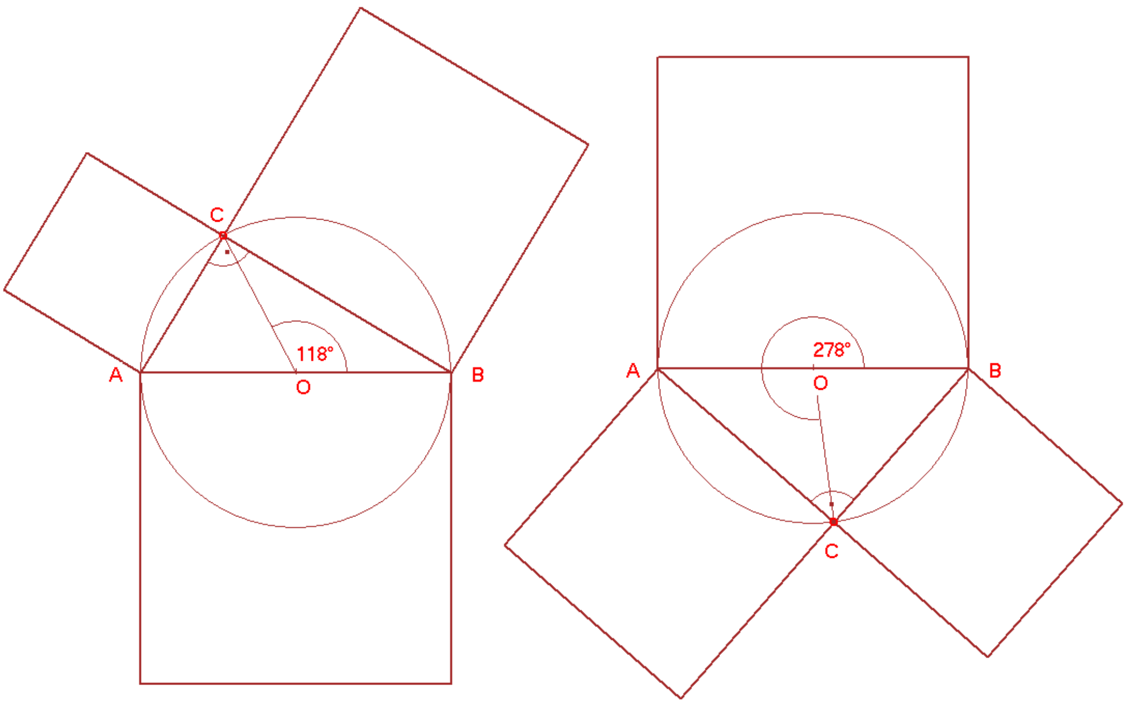
Mivel mindkét tétel derékszögű háromszögre vonatkozik a gyakorlati rész felépítése szinte kínálta magát az összevonásra (33. ábra). A Pitagorasz-tétel az oldalak, a Thalész-tétel pedig a szögek pillanatnyi állapotát vizsgálja, és ezek ugyanazon derékszögű háromszög aktuális állapotában egyszerre megjeleníthetők és a felhasználó akár egyszerre is elemezheti őket. Az egész dinamizmus arra vonatkozik, hogy a háromszög szögei és oldalai folyamatosan változnak, és a felhasználó minden egyes állapotot külön megvizsgálhat, mivel a változások vezérlését egérrel vagy billentyűzettel ő is átveheti.



33. ábra: A Pitagorasz- és Thalész-tétel egy felületen

A programozás technikai oldalnál (Az algoritmus a mellékletekben található.) felvetődik az a kérdés, hogy az ábra paramétereinek változásakor melyik az a paraméter (itt ciklusváltozó), amelyik egy cikluson belül (for vagy while) folyamatosan lineárisan változik, és a többi ettől függ a derékszögű háromszög törvényszerűségeinek megfelelően. (A sinus függvény esetében ez nem volt probléma, mert ott csak két paraméter volt, azaz maga szög és a szög sinusa, egyértelmű volt, hogy a szög változik, mert ő a független változó.) A ’c’ (átmérő ill. átfogó) oldal és a gamma (90°) szög konstans. Adódik a háromszög valamelyik befogója vagy a hegyesszögei. Ezekkel az a probléma, hogy a ’C’ csúcspont mozgása a köríven nem lesz egyenletes, mivel az nem lineárisan függ a háromszög befogóitól vagy szögeitől. Ezek után adódott a kör középponti szöge, amitől a 'C' csúcspont mozgása lineárisan függ, és könnyen számíthatók belőle a háromszög szögei, oldalai és a befogókra állított négyzetek csúcsainak koordinátái. Ennek megfelelően a felület (33. ábra) a felhasználónak visszaadja a középponti szög aktuális értékét is (háromszög oldalai és szögei mellett), és mutatja, hogy azt egy fokonként tudja változtatni.

Adódik még egy probléma, hogy a középponti szög 180° után folyamatos változás közben milyen értéket vegyen fel. A Thalész-tétel okán (a látványosság sem utolsó), mivel az teljes körre vonatkozik nem léphet vissza 0°-ra, hanem 360°-ig fut, és csak ott kezdi elejéről a ciklust. Így természetesen a pitagoraszi-négyzetek is körbe futnak, de ez csak a látványosságot növeli (34. ábra). Egyik tétel sem vonatkozik arra az esetre, amikor a háromszög alfa vagy béta szöge zérus (ekkor a középponti szög 0°, 180° vagy 360°), ugyanis ebben az esetben nincs értelme háromszögről beszélni. Ekkor a thalészi-derékszög és a pitagoraszi-négyzetek is eltűnnek az ábrán.



34. ábra: A középponti szög kiterjesztése 360°-ra

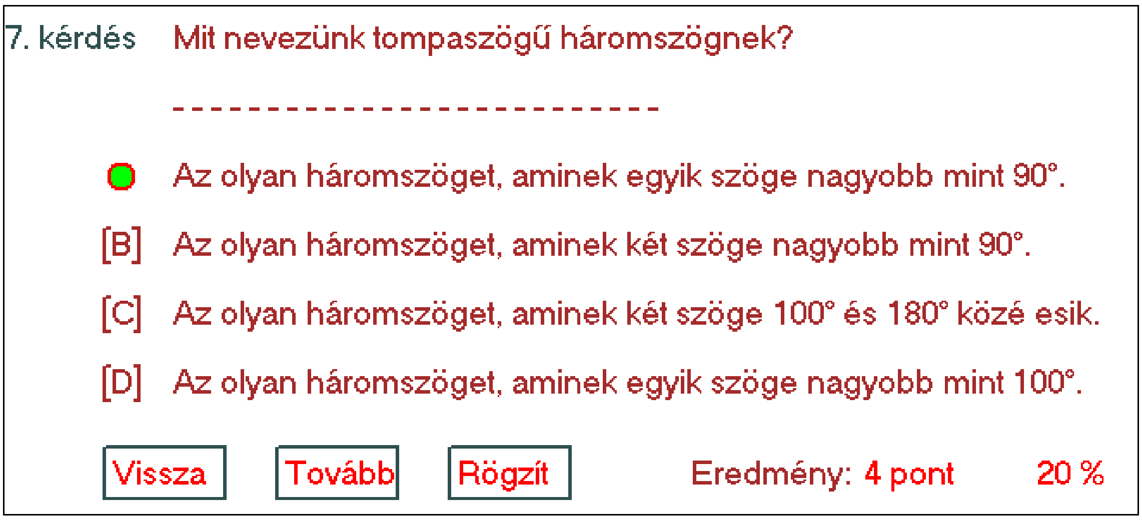
7. ELLENŐRZŐ KÉRDÉSEK, TESZTELÉS

Az ellenőrzés egy hagyományos teszt jellegű feladatsor, ami teljes egészében felöleli az előző menüpontokban megismert tananyagot.

7.1. A tesztfelület felépítése és használata

A tesztfelület egy grafikus felület (az állomány neve: teszt.exe), ami magába foglalja a felület címét, a teszt kérdéseit, a válasz lehetőségeket, a kérdésekre adott válaszok helyességét, az aktuális eredményt (pontszám és százalék), és a felület használatának paramétereit. Az aktuális teszt kitöltésénél mindig 20 kérdésre kell válaszolni. A felhasználó négy lehetséges válaszból választhat, ahol mindig csak egy válasz a jó. A kérdéseken lehet lépkedni oda-vissza, akkor is, ha már van kiválasztva válasz, vagy már le is van véglegesítve (35. ábra). A lépkedés a „Vissza” és „Tovább” kapcsolókkal történik. A kiválasztott válasz betűjele zöld körre változik, amit még lehet azonnal vagy későbbi visszatérés esetén is változtatni. A véglegesítés a „Rögzít” kapcsolóval történik, amit már visszatérés esetén sem lehet módosítani, az ekkor megjelenő jel egy piros kör.

Megoldás közben a felület (35. ábra) folyamatosan mutatja az elért eredményt pontszerűen és százalékosan is.



35. ábra: A 7. tesztkérdés

Fontos, hogy a felhasználó a „Rögzít”-ésnél azonnal lássa, hogy a kiválasztott válasz helyes vagy nem, sőt ha nem helyes megismerje a helyes választ. Hiszen az ellenőrzésnek, esetleg számonkérésnek nemcsak az a célja, hogy a felhasználó megismerje, hogy milyen szinten van a tananyag elsajátításában, hanem az is, hogy tanuljon közben. Tapasztalat, hogy a számonkérés közbeni információk igen könnyen, és erősen rögződnek.

Ezek a véglegesített kérdésekkel kapcsolatos válasz információk jelennek meg a felület jobb alsó részén (36. ábra). Ha válasz helyes pirossal, egyébként pedig világoskékkel.

1. kérdés:		11. kérdés:	
2. kérdés:		12. kérdés:	
3. kérdés:		13. kérdés:	
4. kérdés:		14. kérdés:	Helyes: B
5. kérdés:	Helyes: D	15. kérdés:	Helyes: D
6. kérdés:		16. kérdés:	
7. kérdés:		17. kérdés:	Helyes: D
8. kérdés:	Helyes: C	18. kérdés:	
9. kérdés:		19. kérdés:	
10. kérdés:	Helyes: B	20. kérdés:	Helyes: B

36. ábra: A tesztkérdések válaszai

Egér- és billentyűzetfunkciók:
Az egérrel vezérelhetők a piros színű kapcsolók,
valamint az 'A', 'B', 'C' és 'D', válaszok megjelölhetők.
Vissza: A teszt kérdés sorszámának csökkentése.
Tovább: A teszt kérdés sorszámának növelése.
Rögzít: A megjelölt válasz véglegesítése.
A válasz be van jelölve, még módosítható!
A válasz már nem módosítható!
Escape: Kilép
A teljes adatbázisban 74 darab kérdés van összesen.

37. ábra: Egér- és billentyűzetfunkciók ismertetése

A felület használatához szükséges információk a jobb alsó részen vannak elhelyezve (37. ábra). Itt láthatók még azok az információk is, amiről már a fejezet előző bekezdéseiben is szó volt. A 37. ábra utolsó sora arra utal, hogy a 20 darab kérdés, hány darab kérdésből lett kiválasztva, erről még szó lesz a következő fejezetben.

7.2. A felület elkészítésének programozás technikai oldala

A megoldás kapcsán sok probléma vetődik fel:

- hol legyen elhelyezve a kérdések teljes adatbázisa;
- az adatbázist bármikor lehessen módosítani vagy bővíteni;
- hogyan kerüljön ki onnan a 20 darab aktuális kérdés;
- milyen legyen a 20 kérdés sorrendje;
- hány darab tömb típusú változó szükséges;
- ezen tömbök dimenziója és mérete;
- a kérdések aktuális állapotának tárolása, stb…;

A feladat nagyon összetett, de megoldható (A mellékletekben találhatók a kódolások.) a következőknek megfelelően. A kérdések adatbázisa egy egyszerű szöveges (neve: teszt.txt) file-ban van elhelyezve, aminek tartalma ennek megfelelően bármikor módosítható. Fontos a file felépítése, és annak tiszteletben tartása a módosítás során. Minden kérdés 7 sort foglal el a szöveges állományban. Ebből az első kettő a kérdés maga, a következő négy sor a válasz lehetőségek, a hetedik sor pedig a helyes válasz betűjele. A struktúra szabályait feltétlen be kell tartani, mert, amikor a program olvas belőle, akkor zavarok keletkezhetnek.

Az adatbázis használatának lépései vázlatosan a következők:

1. A tesztelő programrész teljes egészében beolvassa az adatbázis anyagát egy kétdimenziós karakter típusú tömbbe, és olvasás közben rögzíti annak méretét. Így felismeri, hogy hány soros, és annak heted része a kérdések száma, amit a felület jobb alsó sarkán meg is jelenít (37. ábra).
2. A kérdések számát felhasználva véletlenül (rand) generál egy számot, és az ilyen sorszámú kérdés hét sorát átmásolja egy másik (20 kérdéses) karakter

tömbbe. Generálásnál egy egydimenziós kérdésszám méretű logikai tömbbel figyeli, hogy a kérdés nem volt-e már kiválasztva. Ha volt, akkor nem másolja át a kérdést és újra generál egy számot, addig, amíg jót nem talál. Ezt hússzor ismétli, mivel 20 kérdést keres. Ha a forrás adatbázisban nincs 20 kérdés, akkor az ott lévő összes kérdést átmásolja, és azt jelzi a felhasználó felé, Természetesen az elért pontozási eredményt is így kezeli. Mivel a generálás véletlen a kérdések nem az adatbázisban elhelyezett sorrendben jelennek meg a felhasználó előtt.

3. Szükséges még egy kétdimenziós logikai tömb (mérete: 20 * 5), ami az átmásolt kérdések aktuális állapotát tárolja. Induláskor minden eleme zérus. A 20 sor a kérdések számára utal. Minden sornak 5 eleme van, amiből az első négy azt jelöli, hogy melyik válaszlehetőség van bejelölve, az ötödik elem pedig azt, hogy történt-e már véglegesítés. Jelölés vagy véglegesítés esetén az elem értéke 0-ról 1-re változik. Jelölés esetében a szoftver még engedi a visszaállítást, ha a véglegesítés megfelelő eleme még 0.
4. Ha a véglegesítés eleme 1-re változik, akkor megtörténik a kiértékelés, és a helyes válasz megjelenik a felületen pirossal vagy világoskékkel (36. ábra). Ezen kívül még, amennyiben a válasz helyes volt a felületen módosul az elért pontszám is az eredmény rovatban (35. ábra).

8. ÖSSZEGZÉS

Az oktatóprogram elkészült, és ez a munka közben felvetődő bármilyen probléma mellett már eredmény. Öt témakör, és a tesztelési feladat volt betervezve a szoftver tartalmi részét tekintve. Az ötödik témakör, ami a sinus- és cosinus-tétel törvényszerűségeit mutatta volna be kimaradt. A program bármikor bővíthető, akár több menüponttal is, és könnyen elhelyezhető benne többféle új témakör. Nem a szakdolgozat elkészítése, hanem az oktatóprogram témájának algoritmizálása jelentett nagyobb problémát. A programírás időigényes feladat és annak tesztelése, még több időt igényel. Jelenlegi ismeret szerint, minden menüpont tökéletesen működik, és az eddigi tesztelések alapján minden nem megfelelő műveletet kivéd, sőt vissza is jelzi azt.

A menüpontok elkészítése során akadtak olyan problémák, amiket sikerült megoldani és a megoldásból újabb hasznos dolgokat elhelyezni a programban. Ilyen volt a sinus és cosinus függvény menüpontjának elkészítése, ami arra ösztönzött, hogy a tangens függvénynek is legyen helye a trigonometria témakörben. Bár az kicsit nagyobb falat volt, mivel a függvény értékkészlete mínusz és plusz végtelen között mozog, amit az értelmezési tartomány állandó vizsgálatával sikerült kivédeni.

Az előző feladatokhoz képest sokkal nagyobb falat volt a síkidomok kezelése. Itt kerültek elő olyan problémák, amiket nem sikerült teljes egészében megoldani. Sikeresen lett megoldva a csúcsok koordinátáinak kezelése, ahol mindkét vezérlés (egér és billentyűzet) esetében egészszámként jelentek meg a felületen a számított értékek. Ezt persze jelezni kellett a felhasználó felé, mert az összeshez viszonyítva nagyon kevés olyan síkidom létezik, aminek minden oldala és minden szöge egyszerre egészszám. A probléma minden állapotban a függő- és függetlenváltozó egészre kerekítésével lett megoldva.

A háromszögnek három a négyszögnek négy csúcsa van, és mindegyik csúcs mozgatása célkitűzés volt, de az időhiány és az algoritmizálási problémák nehézségei miatt ez csak részlegesen lett megoldva. A háromszögnél egy a négyszögnél két csúcs mozgatható, ettől függetlenül minden fajta háromszög illetve négyszög előállítható. A felhasználó tehát bizonyos határok között minden fajta síkidomot megismerhet. Mint ismert a négyszög két csúcsának mozgatását is két külön területre kellett korlátozni, mert különben az oldalak bizonyos helyzetekben metszik egymást. Ennek fő oka az, hogy nem sikerült azt a programozási problémát megoldani, amik e metszéseket figyelik, és tiltják a hozzájuk tartozó koordináták létrejöttét.

Sikernek könyvelhető el az, hogy a Pitagorasz- és Thalész-tétel gyakorlati részének összevonása megoldódott és így a felhasználó szinkronban figyelheti a paraméterek változását, és összekapcsolhatja a két tételt egy derékszögű háromszögön belül.

A tesztelés menüpont elkészítése nem volt más, mint egy adatbázis-kezelő feladat megoldása. A kódolási rész megoldása mellett tehát fel kellett építeni egy több táblás adatbázist, ami csak a program futása közben létezik. Természetesen az adatbázis egy táblája, (forrásanyag) egy txt állományban folyamatosan létezik. A megoldás a várt céloknak megfelelően hibátlanul sikerült, ugyanis a menüpont a táblák közötti kapcsolatok révén, véletlenül, a sorrend megváltoztatásával generál 20 darab kérdést. A txt állomány pedig a szoftver használata nélkül bármikor bővíthető.

Az oktatóprogram jelenleg olyan állapotban van, hogy minden része az első menüpontban leírtaknak megfelelően működik, és használható. Visszaadja azt, amit a menüpontok neve jelez, és az utolsó menüpontban egy bővíthető adatbázis segítségével a felhasználó ellenőrizheti, hogy milyen szinten van a szoftverben található tananyag elsajátításában.

FELHASZNÁLT IRODALOM

1. BENKŐ TIBORNÉ, BENKŐ LÁSZLÓ, TÓTH BERTALAN, VARGA BALÁZS: *Programozzunk Turbo Pascal nyelven.* ComputerBooks, Budapest, 1996.
2. BENKŐ TIBORNÉ, BENKŐ LÁSZLÓ, TÓTH BERTALAN: *Programozzunk C nyelven.* ComputerBooks, Budapest, 1999.
3. BENKŐ TIBORNÉ, Dr. POPPE ANDRÁS: *Együtt könnyebb a programozás C nyelv.* ComputerBooks, Budapest, 2009.
4. BERNÁT LÁSZLÓ: *Az oktatóprogram készítés egy hatékony alternatívája.* A BGF Kereskedelmi, Vendéglátóipari és Idegenforgalmi Kara, Informatikai Intézet, Budapest, 2002.
5. BRIAN W. KERNIGHAN, DENNIS M. RITCHIE: *A C programozási nyelv.* Műszaki Könyvkiadó, Budapest, 2003.
6. DUSZA ÁRPÁD: *Algoritmusok Pascal nyelven.* Dusza Bt., Miskolc, 2005.
7. HACK FRIGYES: *Négyjegyű függvénytáblázatok – Matematika.* Nemzeti Tankönyvkiadó Rt., Budapest, 1997.
8. Dr. JUHÁSZ ISTVÁN: *Programozás 1-2.* Jegyzet, mobiDIÁK könyvtár, Debrecen, 2003-2004.
9. Dr. NYAKÓNÉ dr. JUHÁSZ KATALIN: *Az informatika iskolai alkalmazásai.* Jegyzet, Matematikai és Informatikai Intézet, Debreceni Egyetem, 2000.
10. SCHWARCZ TIBOR: *Bevezetés a számítógépi grafikába.* Jegyzet, mobiDIÁK könyvtár, Debrecen, 2005.
11. VÁRTERÉSZ MAGDA: *Az informatika logikai alapjai 1.* Jegyzet, Informatikai Kar, Debreceni Egyetem, 2006/2007.

Használt internet felület címmel és témával:

12. http://en.wikipedia.org/wiki/Trigonometry:
    Wikipédia egy angol nyelvű változata a trigonometria történetéről. (2010.04.)
13. http://celebrate.digitalbrain.com/celebrate/accounts/pakh/web/sinus/jell/
    A szinusz függvény főbb jellemzői. (2010.04.)
14. http://en.wikipedia.org/wiki/Trigonometry
    Trigonometria. (2010.04.)
15. http://hu.wikipedia.org/wiki/H%C3%A1romsz%C3%B6g
    Háromszög. (2010.04.)
16. http://hu.wikipedia.org/wiki/N%C3%A9gysz%C3%B6g
    Négyszög. (2010.04.)
17. http://mattort.fvt.hu/cikk.php?cikk=pitagorasz_tetel
    A Pitagorasz-tétel. (2010.04.)
18. http://wapedia.mobi/hu/A_matematikafiloz%C3%B3fia_t%C3%B6rt%C3%
    A matematika filozófia története. (2010.04.)
19. http://www.sulinet.hu/tart/ncikk/Se/0/5526/thalesz_3.htm
    Thalész munkássága. (2010.04.)

MELLÉKLETEK

A menürendszer algoritmusa (Turbo Pascal)

```
program geomokt;
uses crt,dos,graph;
var m,a,tmod:integer;
    ch:char;
    ut:string;
{ 1-******************** Kiírások megoldása! ***************** }
   procedure kiiras(x,y,eloter,hatter:integer;szoveg:string);
   begin
      textcolor(eloter);textbackground(hatter);
      gotoxy(x,y);
      write(szoveg);
   end;
{ 2-******************** Ablak készítés ********************** }
  procedure ablak(x1,y1,x2,y2,keretsz,belsosz:integer);
  var i,j:integer;
  begin
   textcolor(keretsz);textbackground(belsosz);
   for i:=x1 to x2 do begin
     for j:=y1 to y2 do begin
       gotoxy(i,j);write(' ');
     end;
   end;
   gotoxy(x1,y1);write('╔');
   gotoxy(x2,y1);write('╗');
   gotoxy(x1,y2);write('╚');
   gotoxy(x2,y2);write('╝');
   for i:=x1+1 to  x2-1 do begin
    gotoxy(i,y1);write('═');
    gotoxy(i,y2);write('═');
   end;
   for i:=y1+1 to y2-1 do begin
    gotoxy(x1,i);write('║');
    gotoxy(x2,i);write('║');
   end;
  end;
{ 3-************************ Segítség *********************** }
 procedure segit;
 begin
  ablak(31,37,49,44,14,1);
  kiiras(33,37,14,1, '<< Navigáció >>');
  kiiras(33,39,14,1,' Fel:   ');kiiras(41,39,15,1,chr(024));
  kiiras(33,40,14,1,' Le:    ');kiiras(41,40,15,1,chr(025));
  kiiras(33,41,14,1,' Belép: ');kiiras(41,41,15,1,'Enter');
  kiiras(33,42,14,1,' Kilép: ');kiiras(41,42,15,1,'Esc');
 end;
{ 1-************************** Menü! *************************** }
   procedure menu;
   begin
     textcolor(15);textbackground(8);
     clrscr;
     ablak(21,1,60,35,0,7);
     kiiras(25,2,4,7,  '<<< GEOMETRIAI OKTATOPROGRAM >>>');
     if m=1 then kiiras(30,4,14,1, ' Program ismertetés... ')
         else kiiras(30,4,0,7, ' Program ismertetés... ');
     kiiras(21,5,0,7,  '╟──────────────────────────────────────╢');
     kiiras(33,6,4,7,  ' Trigonoemtria ');
     kiiras(21,7,0,7,  '╟──────────────────────────────────────╢');
     if m=2 then kiiras(23,8,14,1,  ' Sinus függvény elmélet... ')
         else kiiras(23,8,0,7,  ' Sinus függvény elmélet... ');
     if m=3 then kiiras(23,9,14,1,  ' Sinus függvény gyakorlat... ')
         else kiiras(23,9,0,7,  ' Sinus függvény gyakorlat... ');
     if m=4 then kiiras(23,10,14,1, ' Cosinus függvény elmélet... ')
         else kiiras(23,10,0,7, ' Cosinus függvény elmélet... ');
     if m=5 then kiiras(23,11,14,1, ' Cosinus függvény gyakorlat... ')
         else kiiras(23,11,0,7, ' Cosinus függvény gyakorlat... ');
     if m=6 then kiiras(23,12,14,1, ' Tangens függvény elmélet... ')
         else kiiras(23,12,0,7, ' Tangens függvény elmélet... ');
     if m=7 then kiiras(23,13,14,1, ' Tangens függvény gyakorlat... ')
         else kiiras(23,13,0,7, ' Tangens függvény gyakorlat... ');
```

```
    kiiras(21,14,0,7, '╟──────────────────────────────────────╢');
    kiiras(33,15,4,7, ' Háromszögek ');
    kiiras(21,16,0,7, '╟──────────────────────────────────────╢');
    if m=8 then kiiras(23,17,14,1, ' Háromszögek elmélet... ')
        else kiiras(23,17,0,7, ' Háromszögek elmélet... ');
    if m=9 then kiiras(23,18,14,1, ' Háromszögek gyakorlat... ')
        else kiiras(23,18,0,7, ' Háromszögek gyakorlat... ');
    kiiras(21,19,0,7, '╟──────────────────────────────────────╢');
    kiiras(34,20,4,7, ' Négyszögek ');
    kiiras(21,21,0,7, '╟──────────────────────────────────────╢');
    if m=10 then kiiras(23,22,14,1, ' Négyszögek elmélet... ')
        else kiiras(23,22,0,7, ' Négyszögek elmélet... ');
    if m=11 then kiiras(23,23,14,1, ' Négyszögek gyakorlat... ')
        else kiiras(23,23,0,7, ' Négyszögek gyakorlat... ');
    kiiras(21,24,0,7, '╟──────────────────────────────────────╢');
    kiiras(27,25,4,7, ' Pitagorasz és Thalész tétel ');
    kiiras(21,26,0,7, '╟──────────────────────────────────────╢');
    if m=12 then kiiras(23,27,14,1, ' Pitagorasz-tétel elmélet... ')
        else kiiras(23,27,0,7, ' Pitagorasz-tétel elmélet... ');
    if m=13 then kiiras(23,28,14,1, ' Thalész-tétel elmélet... ')
        else kiiras(23,28,0,7, ' Thalész-tétel elmélet... ');
    if m=14 then kiiras(23,29,14,1, ' Pitagorasz- és Thalész-tétel gy... ')
        else kiiras(23,29,0,7, ' Pitagorasz- és Thalész-tétel gy... ');
    kiiras(21,31,0,7, '╟──────────────────────────────────────╢');
    if m=15 then kiiras(30,32,14,1, ' Ellenörzö tesztelés... ')
        else kiiras(30,32,0,7, ' Ellenörzö tesztelés... ');
    kiiras(21,33,0,7, '╟──────────────────────────────────────╢');
    kiiras(35,34,0,7, ' Kilépés... ');
  end;
{ ******************* Exe file futtatás! ********************* }
procedure futtat(f:string);
begin
  clrscr;
  {$M $4000,0,0 }
  SwapVectors;
  Exec(f,'');
  SwapVectors;
end;
{ ********************** Programtörzs! *********************** }
begin
  textbackground(0); textcolor(15); clrscr;
  writeln('Adja meg a program teljes anyagának, helyének útvonalát: pl.:
c:\prog\map');
  writeln('Csak Enter nyomásra a c:\geomokt mappában dolgozik!');
  write('Utvonal: ');readln(ut);
  if ut='' then ut:='c:\geomokt';
  tmod:= LastMode;
  m:=2;
  menu;
  segit;
  repeat;
    repeat;
      ch:=readkey;
    until (ch=chr(072)) or (ch=chr(080)) or (ch=chr(013)) or (ch=chr(027));
    if ch=chr(072) then begin
     case m of
       1:begin kiiras(30,4,0,7,  ' Program ismertetés... ');
               kiiras(35,34,14,1,' Kilépés... ');m:=16;end;
       2:begin kiiras(23,8,0,7,  ' Sinus függvény elmélet... ');
               kiiras(30,4,14,1, ' Program ismertetés... ');m:=1;end;
       3:begin kiiras(23,9,0,7,  ' Sinus függvény gyakorlat... ');
               kiiras(23,8,14,1, ' Sinus függvény elmélet... ');m:=2;end;
       4:begin kiiras(23,10,0,7, ' Cosinus függvény elmélet... ');
               kiiras(23,9,14,1, ' Sinus függvény gyakorlat... ');m:=3;end;
       5:begin kiiras(23,11,0,7, ' Cosinus függvény gyakorlat... ');
               kiiras(23,10,14,1,' Cosinus függvény elmélet... ');m:=4;end;
       6:begin kiiras(23,12,0,7, ' Tangens függvény elmélet... ');
               kiiras(23,11,14,1,' Cosinus függvény gyakorlat... ');m:=5;end;
       7:begin kiiras(23,13,0,7, ' Tangens függvény gyakorlat... ');
               kiiras(23,12,14,1,' Tangens függvény elmélet... ');m:=6;end;
       8:begin kiiras(23,17,0,7, ' Háromszögek elmélet... ');
               kiiras(23,13,14,1,' Tangens függvény gyakorlat... ');m:=7;end;
       9:begin kiiras(23,18,0,7, ' Háromszögek gyakorlat... ');
               kiiras(23,17,14,1,' Háromszögek elmélet... ');m:=8;end;
      10:begin kiiras(23,22,0,7, ' Négyszögek elmélet... ');
               kiiras(23,18,14,1,' Háromszögek gyakorlat... ');m:=9;end;
      11:begin kiiras(23,23,0,7, ' Négyszögek gyakorlat... ');
               kiiras(23,22,14,1,' Négyszögek elmélet... ');m:=10;end;
      12:begin kiiras(23,27,0,7, ' Pitagorasz-tétel elmélet... ');
               kiiras(23,23,14,1,' Négyszögek gyakorlat... ');m:=11;end;
      13:begin kiiras(23,28,0,7, ' Thalész-tétel elmélet... ');
               kiiras(23,27,14,1,' Pitagorasz-tétel elmélet... ');m:=12;end;
```

```
       14:begin kiiras(23,29,0,7, ' Pitagorasz- és Thalész-tétel gy... ');
                kiiras(23,28,14,1,' Thalész-tétel elmélet... ');m:=13;end;
       15:begin kiiras(30,32,0,7, ' Ellenörzö tesztelés... ');
                kiiras(23,29,14,1,' Pitagorasz- és Thalész tétel gy...');m:=14;end;
       16:begin kiiras(35,34,0,7, ' Kilépés... ');
                kiiras(30,32,14,1,' Ellenörzö tesztelés... ');m:=15;end;
     end;
    end;
    if ch=chr(080) then begin
     case m of
        1:begin kiiras(30,4,0,7,  ' Program ismertetés... ');
                kiiras(23,8,14,1, ' Sinus függvény elmélet... ');m:=2;end;
        2:begin kiiras(23,8,0,7,  ' Sinus függvény elmélet... ');
                kiiras(23,9,14,1, ' Sinus függvény gyakorlat... ');m:=3;end;
        3:begin kiiras(23,9,0,7,  ' Sinus függvény gyakorlat... ');
                kiiras(23,10,14,1,' Cosinus függvény elmélet... ');m:=4;end;
        4:begin kiiras(23,10,0,7, ' Cosinus függvény elmélet... ');
                kiiras(23,11,14,1,' Cosinus függvény gyakorlat... ');m:=5;end;
        5:begin kiiras(23,11,0,7, ' Cosinus függvény gyakorlat... ');
                kiiras(23,12,14,1,' Tangens függvény elmélet... ');m:=6;end;
        6:begin kiiras(23,12,0,7, ' Tangens függvény elmélet... ');
                kiiras(23,13,14,1,' Tangens függvény gyakorlat... ');m:=7;end;
        7:begin kiiras(23,13,0,7, ' Tangens függvény gyakorlat... ');
                kiiras(23,17,14,1,' Háromszögek elmélet... ');m:=8;end;
        8:begin kiiras(23,17,0,7, ' Háromszögek elmélet... ');
                kiiras(23,18,14,1,' Háromszögek gyakorlat... ');m:=9;end;
        9:begin kiiras(23,18,0,7, ' Háromszögek gyakorlat... ');
                kiiras(23,22,14,1,' Négyszögek elmélet... ');m:=10;end;
       10:begin kiiras(23,22,0,7, ' Négyszögek elmélet... ');
                kiiras(23,23,14,1,' Négyszögek gyakorlat... ');m:=11;end;
       11:begin kiiras(23,23,0,7, ' Négyszögek gyakorlat... ');
                kiiras(23,27,14,1,' Pitagorasz-tétel elmélet... ');m:=12;end;
       12:begin kiiras(23,27,0,7, ' Pitagorasz-tétel elmélet... ');
                kiiras(23,28,14,1,' Thalész-tétel elmélet... ');m:=13;end;
       13:begin kiiras(23,28,0,7, ' Thalész-tétel elmélet... ');
                kiiras(23,29,14,1,' Pitagorasz- és Thalész-tétel gy...');m:=14;end;
       14:begin kiiras(23,29,0,7, ' Pitagorasz- és Thalész-tétel gy... ');
                kiiras(30,32,14,1,' Ellenörzö tesztelés... ');m:=15;end;
       15:begin kiiras(30,32,0,7, ' Ellenörzö tesztelés... ');
                kiiras(35,34,14,1,' Kilépés... ');m:=16;end;
       16:begin kiiras(35,34,0,7, ' Kilépés... ');
                kiiras(30,4,14,1, ' Program ismertetés... ');m:=1;end;
     end;
    end;
    if ch=chr(013) then begin
     menu;
     case m of
        1: futtat(ut+'\ism.exe');
        2: futtat(ut+'\sinusst.exe');
        3: futtat(ut+'\sinusd.exe');
        4: futtat(ut+'\cosinst.exe');
        5: futtat(ut+'\cosind.exe');
        6: futtat(ut+'\tangst.exe');
        7: futtat(ut+'\tangd.exe');
        8: futtat(ut+'\hszogst.exe');
        9: futtat(ut+'\hszogd.exe');
       10: futtat(ut+'\nszogst.exe');
       11: futtat(ut+'\nszogd.exe');
       12: futtat(ut+'\pitst.exe');
       13: futtat(ut+'\thast.exe');
       14: futtat(ut+'\pitthad.exe');
       15: futtat(ut+'\teszt.exe');
       16: ch:=chr(027);
     end;
     TextMode(tmod); menu; segit;
    end;
  until ch=chr(027);
end.
```

A program ismertetésének algoritmusa (C nyelv…)

```
# include <iostream>
# include <stdio.h>
# include <graphics.h>

 int main() {
 int gd,gm,page=0,bill;
 gd=IBM8514;gm=IBM8514HI;
 initgraph(&gd,&gm,"");

while (bill!=27) {
 setactivepage(page);
 setbkcolor(15); cleardevice();
 settextstyle(3,0,4);setcolor(8);
 outtextxy(350,15,"A program ismertetése");
 settextstyle(3,0,2);setcolor(6);
 outtextxy(60,70,"A program menürendszere billentyűzetről vezérelhető, a menük alatt megjelenő kis navigációs ablakban leírtaknak");
 outtextxy(20,100,"megfelelően. Bármelyik menü használható Enter billentyűvel és el is hagyható az Escape billentyűvel. A program is");
 outtextxy(20,130,"elhagyható így, de az utolsó menü is ezt a célt szolgálja.");
 setcolor(8);
 outtextxy(60,160,"Az elmélet csoportba tartozó menük szöveges információkat és ábrákat tartalmaznak, amik segítik a tananyag");
 outtextxy(20,190,"elsajátítását. A szöveges részek definíciókból, képletekből és függvényekből állnak. Az ábrák jelekkel vannak ellátva,");
 outtextxy(20,220,"amik szinkronban vannak a a képletekben található jelölésekkel. Az felületen három-négy szín jelenik meg, amik segítik");
 outtextxy(20,250,"az áttekinthetőséget és az értelmezést. A felületen csak a 'Kilép' kapcsoló vezérelhető egér segítségével.");
 setcolor(6);
 outtextxy(60,280,"A gyakorlat csoportba tartozó menük ugyanazokat az információkat tartalmazzák, mint a hozzájuk tartozó elméleti");
 outtextxy(20,310,"menük, azzal a különbséggel, hogy itt a paraméterek változtathatóak. A gyakorlai változtatás révén ismerhetők meg a");
 outtextxy(20,340,"tananyaghoz kapcsolódó definíciók és törvényszerűségek. A változás révén a felületen nemcsak általános képletek,");
 outtextxy(20,370,"hanem a változást követő számított paraméterek is megjelennek. A hatás mindig piros színnel jelenik meg. A módosítást");
 outtextxy(20,400,"egérrel vagy billentyűzet használattal lehet előidézni. Ezen egér és billentyűzet funkciók leírása többnyire a felület");
 outtextxy(20,430,"alsórészén található szürke színnel. ");
 setcolor(8);
 outtextxy(60,460,"A tananyag elsajátításának szintjét a rendszer 'Ellenőrző tesztelés' nevű menüjével lehet mérni. A menü egérrel");
 outtextxy(20,490,"vezérelhető. A menüben húsz darab teszt kérdés jelenik meg, amihez kapcsolódik négy válaszlehetőség. A válaszok");
 outtextxy(20,520,"közül mindig csak egy válasz helyes. A kérdések között igény szerint lehet lépegetni oda és vissza, akár a válasz");
 outtextxy(20,550,"megadása előtt és után is. A helyes válasz kiválasztása a válasz betűjelére való kattintással történik, de ez nem jelent");
 outtextxy(20,580,"végleges döntést, ez még módosítható, a kérdéshez való visszatérés után is. A véglegesítést a 'Rögzít' kapcsolóval lehet");
 outtextxy(20,610,"elérni, ez után már nincs módosítási lehetőség.");
 setcolor(6);
 outtextxy(60,640,"Véglegesítés után a felületen megjelenik a helyes válasz, ami pontos jelölés esetén piros, egyébként világoskék.");
 outtextxy(20,670,"Továbbá az eredmény rovatban is frissülnek a szerzett pontszámok és a százalékos érték. A felület kitöltés közben is");
 outtextxy(20,700,"elhagyható, de végleges eredmény csak minden kérdés megválaszolása után születik. A megjelenő 20 darab kérdés");
 outtextxy(20,730,"tartalma és sorrendje minden alkalommal változik!");

   setcolor(8);setlinestyle(0,0,2);rectangle(940,7,1000,35);            //kilépés egérrel
   setcolor(4);outtextxy(950,10,"Kilép");setlinestyle(0,0,1);
   if (balgomb)                                                        // egérkezelés
     if (egerx>=940 && egerx<=1000 && egery>=7 && egery<=35) bill=27;

   setvisualpage(page);
   page = 1-page;

   if (kbhit())  bill=getch();                      // billentyűzet kezelés
 }
 closegraph();
 return(0);
}
```

A sinus függvény elméleti részének algoritmusa

```
# include <iostream>
# include <stdio.h>
# include <graphics.h>

int main() {
int i,gd,gm,page=0,bill;
float pi=3.1415926;
char st[10];
gd=IBM8514;gm=IBM8514HI;
initgraph(&gd,&gm,"");

while (bill!=27) {
 setactivepage(page);
 setbkcolor(15); cleardevice();settextstyle(3,0,4);setcolor(8);
 outtextxy(280,15,"A Sinus szögfüggvény bemutatása");
 settextstyle(3,0,2);
 outtextxy(20,80,"Definícó derékszögű háromszög esetében:");
 setcolor(6);outtextxy(380,80," A szöggel szemközti befogó és az átfogó hányadosa adja a szög sinusát.");
 setcolor(8);outtextxy(20,120,"A függvény képlete:");
 setcolor(6);outtextxy(200,120,"sin(  ) = ");
 setlinestyle(0,0,2);circle(234,133,4);line(236,133,239,129);line(236,133,239,136);
 line(275,133,295,133);
 outtextxy(280,105,"a");outtextxy(280,135,"c");
 setcolor(8);outtextxy(305,120,", ahol az  'a'  a szöggel szemközti befogó és a  'c'  az átfogó (lásd az 1.ábrát).");
 outtextxy(200,250,"1. ábra");
 setcolor(6);line(300,330,600,330);setcolor(6);line(600,330,600,180);setcolor(6);line(600,180,300,330);
 circle(345,320,4);line(347,320,350,316);line(347,320,350,323);
 arc(300,330,0,27,70);arc(600,330,90,180,40);
 outtextxy(610,240,"a");outtextxy(450,340,"b");outtextxy(450,220,"c");circle(585,315,2);
 setcolor(8);outtextxy(20,360,"Definícó minden szögre kiterjesztve:");
 setcolor(6);outtextxy(325,360," Egység sugarú körben a sugár függőleges tengelyre eső merőleges vetülete.");
 setcolor(8);
 outtextxy(20,400,"A függvény a definíciónak megfelelően akár a 90° többszörösének is visszaadja a sinusát (lásd a következő menüt).");
 outtextxy(20,440,"Példaként lásd a 150° sinusát egységkörben a 2. ábrán. ");
 outtextxy(500,440,"XY koordináta rendszerben a 3. ábra mutatja a függvényt.");

 setcolor(6);circle(200,610,100);setlinestyle(0,0,1);line(100,610,300,610);line(200,510,200,710);
 line(200,610,112,560);line(112,560,200,560);arc(200,610,0,150,35);outtextxy(230,555,"150°");
 setcolor(2);setlinestyle(0,0,2);line(200,610,200,560);

 setcolor(6);moveto(430,610);
 for(i=-30;i<=390;i++) lineto((430+1.4*i),610+100*sin(i*pi/180));
 setlinestyle(0,0,1);line(380,610,1000,610);line(680,500,680,720);
 for (i=430;i<=934;i+=14) line(i,608,i,612);
 outtextxy(390,615,"-180°"); outtextxy(545,615,"-90°");
 outtextxy(685,615,"0°"); outtextxy(795,615,"90°");
 outtextxy(880,615,"150°"); outtextxy(935,580,"180°");
 outtextxy(302,600,"0°"); outtextxy(195,485,"90°");
 outtextxy(60,590,"180°");outtextxy(55,610,"-180°");
 outtextxy(205,678,"-90°");
 line(681,560,892,560);line(678,510,684,510);line(678,710,684,710);
 outtextxy(690,500,"+1");outtextxy(690,700,"-1");
 outtextxy(1010,595,"x");outtextxy(650,470,"y = sin(x)");
 setcolor(2);setlinestyle(0,0,2);line(892,610,892,560);
 setcolor(8);outtextxy(180,725,"2. ábra");outtextxy(650,725,"3. ábra");

 setcolor(8);setlinestyle(0,0,2);rectangle(940,7,1000,35);            //kilépés egérrel
 setcolor(4);outtextxy(950,10,"Kilép");setlinestyle(0,0,1);

 if (balgomb)
   if (egerx>=940 && egerx<=1000 && egery>=7 && egery<=35) bill=27;        // egérkezelés

   setvisualpage(page);
   page = 1-page;
 if (kbhit())  bill=getch();                  // billentyűzet kezelés
 }
 closegraph();
 return(0);
}
```

A sinus függvény gyakorlati részének algoritmusa

```
# include <iostream>
# include <stdio.h>
# include <graphics.h>

 int main() {
 int sz=0,i,a=2,b,gd,gm,page=0,bill;
 float pi=3.1415926;
 char st[10];
 gd=IBM8514;gm=IBM8514HI;
 initgraph(&gd,&gm,"");

 while (bill!=27) {
   setactivepage(page);
   cleardevice(); setbkcolor(15);settextstyle(3,0,4);setcolor(8);
   outtextxy(170,35,"A Sinus függvény karakterisztikája 0° és 360° között");
   settextstyle(3,0,2);
   outtextxy(170,90,"A nevezetes szögek: 0°,30°,90°,150°,180°,210°,270°,330°,360° és ezek 180°-al eltolva.");
   setcolor(6);outtextxy(20,550,"Egér és billentyűzet funkciók:");setcolor(8);
   outtextxy(280,550,"Az függvény értéke mindig leolvasható, ami az egységkör zöld szakaszának felel meg!");
   outtextxy(280,580,"Az egérrel a futókör szabadon vezérelhető.");
   outtextxy(280,610,"G: Megállít és balra lép 1°-al.");
   outtextxy(280,640,"H: Megállít és jobbra lép 1°-al.");
   outtextxy(280,670,"Space és Enter: Elindít vagy irányt vált.");
   outtextxy(280,700,"Escape: Visszalép a menürendszerbe.");

   settextstyle(3,0,1);setcolor(6);                                      // szögek felírása
   outtextxy(302,320,"0°");outtextxy(475,165,"90°");
   outtextxy(625,320,"180°"); outtextxy(820,450,"270°");
   outtextxy(145,165,"90°"); outtextxy(25,295,"180°");
   outtextxy(145,450,"270°"); outtextxy(980,295,"360°");
   outtextxy(240,320,"360°");
   outtextxy(304,180,"+1");outtextxy(304,440,"-1");
   for (i=300;i<=1020;i+=20) line(i,318,i,322);                          // egyenesek
   line(1020,320,300,320);moveto(300,320);
   line(20,320,280,320);line(150,190,150,450);
   line(300,180,300,460);line(298,190,302,190);line(298,450,302,450);
   setlinestyle(0,0,2);
   for(i=0;i<=720;i+=2) lineto(300+i,320-130*sin(i*pi/360));             // sinus görbe
   circle(150,320,130);                                                 // egységkör
   setlinestyle(0,0,1);

   if (sz>720) sz=0;                                                    // kezdő és végső sz érték váltása
   if (sz<0)  sz=720;

   moveto(150,320);                                                     // forgó vektor felrajzolása
   lineto((int)150+130*cos(sz*pi/360),(int)320-130*sin(sz*pi/360));
   lineto(150,(int)320-130*sin(sz*pi/360));
  setlinestyle(0,0,2);setcolor(2);lineto(150,320);setlinestyle(0,0,1);setcolor(6);

  setlinestyle(0,0,2);
  circle(300+sz,320-130*sin(sz*pi/360),5);                              // futó kör vagy pont
  setcolor(2);line(300+sz,320,300+sz,320-130*sin(sz*pi/360));
  setcolor(6);setlinestyle(0,0,1);
  line(300,320-130*sin(sz*pi/360),300+sz,320-130*sin(sz*pi/360));


  char fok[10]="Sinus ";strcat(fok,itoa(sz/2,st,10));strcat(fok,"° =");  // sinus érték számítása és kiírása
  long e=1000000*sin(sz*pi/360); char ertek[10];
  if (e<0) {strcpy(ertek,"-0,");e=abs(e);} else strcpy(ertek,"0,");
  if (e<100000) strcat(ertek,"0");
  itoa(e,st,10);strcat(ertek,st);
  if (sz==180) strcpy(ertek,"1,000000");if (sz==540) strcpy(ertek,"-1,000000");
  if (sz==60) strcpy(ertek,"0,500000"); if (sz==420) strcpy(ertek,"-0,500000");
  strcat(fok,ertek);
  settextstyle(3,0,3);setcolor(4);outtextxy(20,500,"A függvény értéke: ");
  outtextxy(230,500,fok);

```

```
    settextstyle(3,0,2);setcolor(8);setlinestyle(0,0,2);rectangle(940,7,1000,35);     //kilépés egérrel
    setcolor(4);outtextxy(950,10,"Kilép");setlinestyle(0,0,1);
    sz+=a;
        if (balgomb) {                                                                //  egérkezelés
          if (egerx<300) egerx=300;
          if (egerx>1020) egerx=1020;
          if (egerx>=940 && egerx<=1000 && egery>=7 && egery<=35) bill=27;
           sz=integer(egerx-300);
        }
    setvisualpage(page);
    page = 1-page;
    delay(30);

    if (kbhit()) {                                                                 // billentyűzet kezelés
      bill=getch();
      switch (bill) {
        case 13: {if (a==0) a=b; else a*=-1; break;}      // enter
        case 32: {if (a==0) a=b; else a*=-1; break;}      // space
        case 103: {a=0;b=-2;sz+=-2; break;}               // g betű
        case 104: {a=0;b=2;sz+=2; break;}                 // h betű
      }
    }
  }
  closegraph();
  return(0);
}
```

A cosinus függvény elméleti részének algoritmusa

```
# include <iostream>
# include <stdio.h>
# include <graphics.h>

int main() {
int i,gd,gm,page=0,bill;
float pi=3.1415926;
char st[10];
gd=IBM8514;gm=IBM8514HI;
initgraph(&gd,&gm,"");

while (bill!=27) {
 setactivepage(page);
 setbkcolor(15); cleardevice(); settextstyle(3,0,4);setcolor(8);
 outtextxy(280,15,"A Cosinus szögfüggvény bemutatása");
 settextstyle(3,0,2);
 outtextxy(20,80,"Definícó derékszögű háromszög esetében:");
 setcolor(6);outtextxy(380,80," A szög melletti befogó és az átfogó hányadosa adja a szög cosinusát.");
 setcolor(8);outtextxy(20,120,"A függvény képlete:");
 setcolor(6);outtextxy(200,120,"cos(   ) = ");
 setlinestyle(0,0,2);circle(240,133,4);line(242,133,245,129);line(242,133,245,136);line(275,133,295,133);
 outtextxy(280,105,"b");outtextxy(280,135,"c");
 setcolor(8);outtextxy(305,120,", ahol 'b' a szög melletti befogó és 'c'  az átfogó (lásd az 1.ábrát).");
 outtextxy(200,250,"1. ábra");
 setcolor(6);line(300,330,600,330);setcolor(6);line(600,330,600,180);setcolor(6);line(600,180,300,330);
 circle(345,320,4);line(347,320,350,316);line(347,320,350,323);
 arc(300,330,0,27,70);arc(600,330,90,180,40);
 outtextxy(610,240,"a");outtextxy(450,340,"b");outtextxy(450,220,"c");circle(585,315,2);

 setcolor(8);outtextxy(20,360,"Definícó minden szögre kiterjesztve:");
 setcolor(6);outtextxy(325,360," Egység sugarú körben a sugár vízszintes tengelyre eső merőleges vetülete.");
 setcolor(8);outtextxy(20,400,"A függvény a definíciónak megfelelően akár a 90° többszörösének is visszaadja a cosinusát.");
 outtextxy(20,440,"Példaként lásd a 120° cosinusát egységkörben a 2. ábrán. ");
 outtextxy(515,440,"XY koordináta rendszerben a 3. ábra mutatja a függvényt.");

 setcolor(6);circle(200,610,100);setlinestyle(0,0,1);line(100,610,300,610);line(200,510,200,710);
 line(200,610,150,522);line(150,522,150,610);arc(200,610,0,120,35);outtextxy(230,555,"120°");
 setcolor(2);setlinestyle(0,0,2);line(200,610,150,610);

 setcolor(6);moveto(388,695); for(i=-30;i<=390;i++) lineto((430+1.4*i),610+100*cos(i*pi/180));
 setlinestyle(0,0,1);line(380,610,1000,610);line(682,500,682,720);
 for (i=388;i<=976;i+=14) line(i,608,i,612);
 outtextxy(410,580,"-180°"); outtextxy(555,615,"-90°");
 outtextxy(685,615,"0°"); outtextxy(785,615,"90°");
 outtextxy(835,580,"120°"); outtextxy(920,580,"180°");
 outtextxy(302,600,"0°"); outtextxy(195,485,"90°");
 outtextxy(60,590,"180°");outtextxy(55,610,"-180°");
 outtextxy(205,678,"-90°");

 line(682,660,851,660);line(678,510,684,510);line(680,710,686,710);
 line(934,610,934,710);line(430,610,430,710);
 outtextxy(690,480,"+1");outtextxy(690,700,"-1");
 outtextxy(1010,595,"x");outtextxy(750,480,"y = cos(x)");
 setcolor(2);setlinestyle(0,0,2);line(851,610,851,660);
 setcolor(8);outtextxy(180,725,"2. ábra");outtextxy(650,725,"3. ábra");

 setcolor(8);setlinestyle(0,0,2);rectangle(940,7,1000,35);                //kilépés egérrel
 setcolor(4);outtextxy(950,10,"Kilép");setlinestyle(0,0,1);

 if (balgomb)
    if (egerx>=940 && egerx<=1000 && egery>=7 && egery<=35) bill=27;          // egérkezelés

   setvisualpage(page);
   page = 1-page;

   if (kbhit())  bill=getch();                                             // billentyűzet kezelés
 }
 closegraph();
 return(0);
}
```

A cosinus függvény gyakorlati részének algoritmusa

```
# include <iostream>
# include <stdio.h>
# include <graphics.h>

int main() {
  int sz=0,i,a=2,b,gd,gm,page=0,bill;
  float pi=3.1415926;
  char st[10];
  gd=IBM8514;gm=IBM8514HI;
  initgraph(&gd,&gm,"");

  while (bill!=27) {
    setactivepage(page);
    cleardevice(); setbkcolor(15);settextstyle(3,0,4);setcolor(8);
    outtextxy(170,35,"A Cosinus függvény karakterisztikája 0° és 360° között");
    settextstyle(3,0,2);
    outtextxy(170,90,"A nevezetes szögek: 0°,60°,90°,120°,180°,240°,270°,300°,360° és ezek 180°-al eltolva.");
    setcolor(6);outtextxy(20,550,"Egér és billentyűzet funkciók:");setcolor(8);
    outtextxy(280,550,"Az függvény értéke mindig leolvasható, ami az egységkör zöld szakaszának felel meg!");
    outtextxy(280,580,"Az egérrel a futókör szabadon vezérelhető.");
    outtextxy(280,610,"G: Megállít és balra lép 1°-al.");
    outtextxy(280,640,"H: Megállít és jobbra lép 1°-al.");
    outtextxy(280,670,"Space és Enter: Elindít vagy irányt vált.");
    outtextxy(280,700,"Escape: Visszalép a menürendszerbe.");

    settextstyle(3,0,1);setcolor(6);                                   // szögek felírása
    outtextxy(302,320,"0°");outtextxy(480,295,"90°");
    outtextxy(645,295,"180°"); outtextxy(835,325,"270°");
    outtextxy(145,165,"90°"); outtextxy(25,295,"180°");
    outtextxy(145,450,"270°"); outtextxy(980,295,"360°");
    outtextxy(240,320,"360°");outtextxy(275,180,"+1");outtextxy(280,440,"-1");
    for (i=300;i<=1020;i+=20) line(i,318,i,322);

    line(300,320,1020,320);line(660,320,660,450);moveto(300,320);      // egyenesek
    line(20,320,280,320);line(150,190,150,450);
    line(300,180,300,460);line(298,190,302,190);line(298,450,302,450);
    setlinestyle(0,0,2);moveto(300,190);
    for(i=0;i<=720;i+=2) lineto(300+i,320-130*cos(i*pi/360));            // cosinus görbe
    circle(150,320,130); setlinestyle(0,0,1);                          // egységkör

    if (sz>720) sz=0;                                     // kezdő és végső sz érték váltása
    if (sz<0)  sz=720;

    moveto(150,320);                                          // forgó vektor felrajzolása
    lineto((int)150+130*cos(sz*pi/360),(int)320-130*sin(sz*pi/360));
    lineto((int)150+130*cos(sz*pi/360),320);
    setlinestyle(0,0,2);setcolor(2);lineto(150,320);setlinestyle(0,0,1);setcolor(6);

    setlinestyle(0,0,2);
    circle(300+sz,320-130*cos(sz*pi/360),5);                  // futó kör vagy pont
    setcolor(2);line(300+sz,320,300+sz,320-130*cos(sz*pi/360));
    setcolor(6);setlinestyle(0,0,1);
    line(300,320-130*cos(sz*pi/360),300+sz,320-130*cos(sz*pi/360));

    setcolor(8);setlinestyle(0,0,2);rectangle(940,7,1000,35);               //kilépés egérrel
    setcolor(4);outtextxy(950,10,"Kilép");setlinestyle(0,0,1);

    char fok[10]="Cosinus ";strcat(fok,itoa(sz/2,st,10));strcat(fok,"° =");  // cosinus érték számítása és kiírása
    long e=1000000*cos(sz*pi/360); char ertek[10];
    if (e<0) {strcpy(ertek,"-0,");e=abs(e);} else strcpy(ertek,"0,");
    if (e<100000) strcat(ertek,"0");
    itoa(e,st,10);strcat(ertek,st);
    if (sz==240) strcpy(ertek,"-0,500000");if (sz==360) strcpy(ertek,"-1,000000");
    if (sz==600) strcpy(ertek,"0,500000");if (sz==720) strcpy(ertek,"1,000000");
    strcat(fok,ertek);
    settextstyle(3,0,3);setcolor(4);outtextxy(20,500,"A függvény értéke: ");
    outtextxy(230,500,fok);
```

```
    sz+=a;
      if (balgomb) {
          if (egerx<300) egerx=300;                                              // egérkezelés
          if (egerx>1020) egerx=1020;
          if (egerx>=940 && egerx<=1000 && egery>=7 && egery<=35) bill=27;
           sz=integer(egerx-300);
      }
    setvisualpage(page);
    page = 1-page;
    delay(30);

    if (kbhit()) {                                                              // billentyűzet kezelés
      bill=getch();
      switch (bill) {
        case 13: {if (a==0) a=b; else a*=-1; break;}    // enter
        case 32: {if (a==0) a=b; else a*=-1; break;}    // space
        case 103: {a=0;b=-2;sz+=-2; break;}              // g betű
        case 104: {a=0;b=2;sz+=2; break;}                // h betű
      }
    }
  }
  closegraph();
  return(0);
}
```

A tangens függvény elméleti részének algoritmusa

```
# include <iostream>
# include <stdio.h>
# include <graphics.h>

 int main() {
 int i,gd,gm,page=0,bill;
 float pi=3.1415926;
 char st[10];
 gd=IBM8514;gm=IBM8514HI;
 initgraph(&gd,&gm,"");

 while (bill!=27) {
   setactivepage(page);
   setbkcolor(15); cleardevice();settextstyle(3,0,4);setcolor(8);
   outtextxy(280,15,"A Tangens szögfüggvény bemutatása");
   settextstyle(3,0,2);
   outtextxy(20,80,"Definíció derékszögű háromszögben:");
   setcolor(6);outtextxy(340,80," A szöggel szemközti és a szög melletti befogó hányadosa adja a szög tangensét.");
   setcolor(8);outtextxy(20,120,"A függvény képlete:");
   setcolor(6);outtextxy(200,120,"  tg(  ) = ");
   setlinestyle(0,0,2);circle(235,133,4);line(237,133,240,129);line(237,133,240,136);line(275,133,295,133);
   outtextxy(280,105,"a");outtextxy(280,135,"b");
   setcolor(8);outtextxy(305,120,", ahol 'a' a szöggel szemközti és 'b' a szög melletti befogó (lásd az 1.ábrát).");
   outtextxy(200,250,"1. ábra");
   setcolor(6);line(300,330,600,330);setcolor(6);line(600,330,600,180);setcolor(6);line(600,180,300,330);
   circle(345,320,4);line(347,320,350,316);line(347,320,350,323);
   arc(300,330,0,27,70);arc(600,330,90,180,40);
   outtextxy(610,240,"a");outtextxy(450,340,"b");outtextxy(450,220,"c");circle(585,315,2);

   setcolor(8);outtextxy(20,360,"Definíció minden szögre:");
   setcolor(6);outtextxy(230,360," Egységkörben a sugár meghosszabbításának merőleges vetülete a függőleges érintőre.");
   setcolor(8);outtextxy(20,400,"A függvény a definíciónak megfelelően akár a 90° többszörösének is visszaadja a tangensét.");
   outtextxy(20,440,"Példaként lásd a 135° tangensét egységkörben a 2. ábrán. ");
   outtextxy(515,440,"XY koordináta rendszerben a 3. ábra mutatja a függvényt.");

   setcolor(6);circle(200,610,100);setlinestyle(0,0,1);line(100,610,300,610);line(200,510,200,710);
   line(300,710,130,538);line(300,480,300,740);arc(200,610,0,135,35);outtextxy(230,555,"135°");
   setcolor(2);setlinestyle(0,0,2);line(300,610,300,710);

   setcolor(6);
   for(i=-60;i<=420;i++)
      if (abs(100*tan(i*pi/180))<140) line((430+1.4*i),610-100*tan(i*pi/180),(430+1.4*(i+1)),610-100*tan((i+1)*pi/180));
   setlinestyle(0,0,1);line(380,610,1000,610);line(682,500,682,720);
   for (i=388;i<=976;i+=14) line(i,608,i,612);
   outtextxy(430,615,"-180°"); outtextxy(560,615,"-90°");
   outtextxy(685,615,"0°"); outtextxy(775,615,"90°");
   outtextxy(850,580,"135°"); outtextxy(940,615,"180°");

   outtextxy(302,600,"0°"); outtextxy(195,485,"90°");
   outtextxy(60,590,"180°");outtextxy(55,610,"-180°");
   outtextxy(205,678,"-90°");

   line(682,710,872,710);line(680,510,686,510);line(680,710,686,710);
   setlinestyle(3,0,1);line(808,475,808,745);line(556,475,556,745);
   outtextxy(815,500,"aszimptota");
   outtextxy(655,500,"+1");outtextxy(660,700,"-1");
   outtextxy(1010,595,"x");outtextxy(640,470,"y = tg(x)");
   setcolor(2);setlinestyle(0,0,2);line(872,610,872,708);
   setcolor(8);outtextxy(180,725,"2. ábra");outtextxy(650,725,"3. ábra");

   setcolor(8);setlinestyle(0,0,2);rectangle(940,7,1000,35);                    //kilépés egérrel
   setcolor(4);outtextxy(950,10,"Kilép");setlinestyle(0,0,1);

   if (balgomb)
     if (egerx>=940 && egerx<=1000 && egery>=7 && egery<=35) bill=27;          // egérkezelés

   setvisualpage(page);
   page = 1-page;
   if (kbhit())  bill=getch();                                         // billentyűzet kezelés
 }
 closegraph();
 return(0);
}
```

A tangens függvény gyakorlati részének algoritmusa

```
# include <iostream>
# include <stdio.h>
# include <graphics.h>

int main() {
int sz=0,i,a=2,b=2,gd,gm,page=0,bill;
float pi=3.1415926;
char st[10];
gd=IBM8514;gm=IBM8514HI;
initgraph(&gd,&gm,"");

while (bill!=27) {
  setactivepage(page);
  cleardevice();  setbkcolor(15);settextstyle(3,0,4);setcolor(8);
  outtextxy(170,35,"A Tangens függvény karakterisztikája 0° és 360° között");
  settextstyle(3,0,2);
  outtextxy(170,90,"A nevezetes szögek: 0°,45°,90°,135°,180°,225°,270°,315°,360° és ezek 180°-al eltolva.");
  setcolor(6);outtextxy(20,550,"Egér és billentyűzet funkciók:");setcolor(8);
  outtextxy(280,550,"Az függvény értéke mindig leolvasható, ami az egységkör zöld szakaszának felel meg!");
  outtextxy(280,580,"Az egérrel a futókör szabadon vezérelhető.");
  outtextxy(280,610,"G: Megállít és balra lép 1°-al.");
  outtextxy(280,640,"H: Megállít és jobbra lép 1°-al.");
  outtextxy(280,670,"Space és Enter: Elindít vagy irányt vált.");
  outtextxy(280,700,"Escape: Visszalép a menürendszerbe.");

  settextstyle(3,0,1);setcolor(6);
  outtextxy(302,320,"0°");outtextxy(485,295,"90°");                          // szögek felírása
  outtextxy(635,295,"180°"); outtextxy(845,325,"270°");
  outtextxy(135,165,"90°"); outtextxy(25,285,"180°");
  outtextxy(135,450,"270°"); outtextxy(980,295,"360°");
  outtextxy(240,320,"360°");
  outtextxy(310,180,"+1");outtextxy(310,440,"-1");
  for (i=300;i<=1020;i+=20) line(i,318,i,322);

  line(300,320,1020,320);moveto(300,320);                                   // egyenesek
  line(20,320,280,320);line(150,190,150,450);
  line(300,120,300,520);line(298,190,302,190);line(298,450,302,450);
  setlinestyle(3,0,1);line(480,120,480,520);line(840,120,840,520);
  outtextxy(490,160,"aszimptota-1");outtextxy(850,160,"aszimptota-2");
  setlinestyle(0,0,2);moveto(300,190);
  for(i=0;i<=720;i+=2)
    if (200>abs(130*tan(i*pi/360))) line(300+i,320-130*tan(i*pi/360),300+i+2,320-130*tan((i+2)*pi/360));  // tangens görbe
  circle(150,320,130); setlinestyle(0,0,1);                                                             // egységkör


  if (sz>720) sz=0;                                                // kezdő és végső sz érték váltása
  if (sz<0)  sz=720;

  line(150+130,120,150+130,520);moveto(150,320);
  lineto((int)150+130*cos(sz*pi/360),(int)320-130*sin(sz*pi/360));         // forgó vektor felrajzolása
  lineto(150+130,(int)320-130*tan(sz*pi/360));
  setlinestyle(0,0,2);
  if (sz/2!=90 && sz/2!=270) { setcolor(2);lineto(150+130,320);
  }  else   {  setcolor(2);line(280,800,280,0);  }
    setcolor(6);
    if (sz/2!=90 && sz/2!=270) {
               circle(300+sz,320-130*tan(sz*pi/360),5);                      // futó kör vagy pont
               setcolor(2);line(300+sz,320,300+sz,320-130*tan(sz*pi/360));
    }   else   {   setcolor(2);line(300+sz,800,300+sz,0);  }
  setcolor(6);setlinestyle(0,0,1);
  line(300,320-130*tan(sz*pi/360),300+sz,320-130*tan(sz*pi/360));


  char fok[10]="Tangens ";strcat(fok,itoa(sz/2,st,10));strcat(fok,"° =");     // tangens érték számítása és kiírása
  long e=1000000*tan(sz*pi/360); char ertek[10];
  if (e<0) strcpy(ertek,"-0,"); else strcpy(ertek,"0,");
  long d=abs(e); if (d<100000) strcat(ertek,"0");
```

```
    if (e>-1000000) e=abs(e);
    itoa(e,st,10);strcat(ertek,st);
    if (e>1000000) { strcpy(ertek,"");
        long g=e/1000000;long m=e%1000000;
        g=abs(g);itoa(g,st,10);strcat(ertek,st);strcat(ertek,",");
        if (m<100000)  strcat(ertek,"0");
        m=abs(m);itoa(m,st,10);strcat(ertek,st);
    }
    if (e<-1000000) { strcpy(ertek,"-");
        e=abs(e);
        long g=e/1000000;long m=e%1000000;
        g=abs(g); itoa(g,st,10);strcat(ertek,st);strcat(ertek,",");
        if (m<100000)  strcat(ertek,"0");
        m=abs(m);itoa(m,st,10);strcat(ertek,st);
    }
    if (sz==180) strcpy(ertek,"+/- végtelen");if (sz==540) strcpy(ertek,"+/- végtelen");
    if (sz==90) strcpy(ertek,"1,000000");if (sz==450) strcpy(ertek,"1,000000");
    if (sz==270) strcpy(ertek,"-1,000000");if (sz==630) strcpy(ertek,"-1,000000");
    strcat(fok,ertek);
    settextstyle(3,0,3);setcolor(4);outtextxy(20,500,"A függvény értéke: ");
    outtextxy(230,500,fok);
    settextstyle(3,0,2);setcolor(8);setlinestyle(0,0,2);rectangle(940,7,1000,35);           //kilépés egérrel
    setcolor(4);outtextxy(950,10,"Kilép");setlinestyle(0,0,1);
    sz+=a;
        if (balgomb) {
          if (egerx<300) egerx=300;                                                            // egérkezelés
          if (egerx>1020) egerx=1020;
          if (egerx>=940 && egerx<=1000 && egery>=7 && egery<=35) bill=27;
           sz=integer(egerx-300);
        }
    setvisualpage(page);
    page = 1-page;
    delay(30);
    if (kbhit()) {                                                                         // billentyűzet kezelés
      bill=getch();
      switch (bill) {
        case 13: {if (a==0) a=b; else a*=-1; break;}    // enter
        case 32: {if (a==0) a=b; else a*=-1; break;}    // space
        case 103: {a=0;b=-2;sz+=-2; break;}             // g betű
        case 104: {a=0;b=2;sz+=2; break;}               // h betű
      }
    }
  }
  closegraph();
  return(0);
}
```

A háromszögek elméleti részének algoritmusa

```
# include <iostream>
# include <stdio.h>
# include <graphics.h>

 int main() {
 int i,gd,gm,page=0,bill;
 char st[10];
 gd=IBM8514;gm=IBM8514HI;
 initgraph(&gd,&gm,"");

 while (bill!=27) {
   setactivepage(page);
   setbkcolor(15); cleardevice();settextstyle(3,0,4);setcolor(8);
   outtextxy(240,15,"A háromszög definíciója, jellemzői és fajtái");
   settextstyle(3,0,2);outtextxy(20,60,"Definícó:");
   setcolor(6);outtextxy(100,60,"Olyan zárt síkidom, aminek három csúcsa (szöge) van, és három szakasz határolja.");
   setcolor(8);outtextxy(20,100,"Jellemzői:");
   setcolor(6);
   outtextxy(120,100,"Szögeinek összege 180°,  Kerülete: a+b+c,  Területe: (a*m)/2, ahol 'm' az 'a'-hoz tartozó magasság.");
   setcolor(8);outtextxy(20,160,"Fajtái:");
   setcolor(4);outtextxy(220,370,"Hegyesszögű");outtextxy(720,370,"Tompaszögű");
   outtextxy(130,700,"Derékszögű");outtextxy(450,700,"Egyenlőszárú");outtextxy(770,700,"Egyenlő oldalú");
   setcolor(8);
   outtextxy(200,395,"Minden szöge < 90°");outtextxy(710,395,"Egy szöge > 90°");
   outtextxy(110,725,"Egy szöge = 90°");outtextxy(430,725,"Két szöge egyenlő");outtextxy(750,725,"Minden szöge = 60°");

   setcolor(6);setlinestyle(0,0,2);settextstyle(3,0,2);                          // hegyesszögű
   moveto(140,340);lineto(450,340);lineto(200,150);lineto(140,340);
   circle(215,325,1);setlinestyle(0,0,1);line(200,340,200,150);
   arc(200,340,0,90,35);arc(200,150,252,325,35);
   outtextxy(290,345,"a");outtextxy(155,205,"b");outtextxy(310,205,"c");outtextxy(210,250,"m");
   setlinestyle(0,0,2);circle(207,168,4);line(209,168,212,164);line(209,168,212,171);   //alfa
   outtextxy(20,200,"K = a+b+c");outtextxy(20,240,"T = (a*m)/2");
   setcolor(6);setlinestyle(0,0,2);settextstyle(3,0,2);                          // tompaszögű
   moveto(690,340);lineto(1000,340);lineto(650,150);lineto(690,340);
   circle(665,325,1);setlinestyle(0,0,1);line(650,150,650,340);line(690,340,635,340);
   arc(650,340,0,90,35);arc(690,340,0,103,60);
   outtextxy(840,345,"a");outtextxy(655,230,"b");outtextxy(800,205,"c");outtextxy(625,250,"m");
   outtextxy(850,180,"K = a+b+c");outtextxy(850,220,"T = (a*m)/2");outtextxy(690,305," >90°");
   setcolor(6);setlinestyle(0,0,2);settextstyle(3,0,2);                          // derékszögű
   moveto(40,680);lineto(350,680);lineto(40,490);lineto(40,680);
   circle(55,665,1);setlinestyle(0,0,1);circle(55,665,1);arc(40,680,0,90,35);
   outtextxy(190,681,"a");outtextxy(20,575,"b");outtextxy(190,558,"c");
   outtextxy(160,480,"K = a+b+c");outtextxy(160,520,"T = (a*b)/2");
   setcolor(6);setlinestyle(0,0,2);settextstyle(3,0,2);                          // egyenlőszárú
   moveto(400,680);lineto(630,680);lineto(515,420);lineto(400,680);
   circle(528,665,1);setlinestyle(0,0,1);circle(528,665,1);arc(515,680,0,90,35);line(515,680,515,420);
    outtextxy(520,560,"m");arc(400,680,0,68,35);arc(630,680,112,180,35);
    outtextxy(510,681,"a");outtextxy(420,560,"b");outtextxy(600,560,"b");
    outtextxy(570,440,"K = a+2*b");outtextxy(570,480,"T = (a*m)/2");
    outtextxy(413,655,"ß");outtextxy(607,655,"ß");

    setcolor(6);setlinestyle(0,0,2);settextstyle(3,0,2);                         // egyenlő oldalú
    moveto(730,680);lineto(960,680);lineto(845,481);lineto(730,680);
    circle(858,665,1);setlinestyle(0,0,1);circle(858,665,1);arc(845,680,0,90,35);
    line(845,680,845,481);outtextxy(850,560,"m");
    arc(730,680,0,60,50);arc(960,680,120,180,50);arc(845,481,240,300,50);
    outtextxy(840,681,"a");outtextxy(750,560,"a");outtextxy(930,560,"a");
    outtextxy(900,440,"K = 3*a");outtextxy(900,480,"T = (a*m)/2");
    outtextxy(746,654,"60°");outtextxy(918,654,"60°");outtextxy(833,505,"60°");

    setcolor(8);setlinestyle(0,0,2);rectangle(940,7,1000,35);                    //kilépés egérrel
    setcolor(4);outtextxy(950,10,"Kilép");setlinestyle(0,0,1);
    if (balgomb)
      if (egerx>=940 && egerx<=1000 && egery>=7 && egery<=35) bill=27;           // egérkezelés

    setvisualpage(page);
    page = 1-page;
    if (kbhit())  bill=getch();                       // billentyűzet kezelés
  }
  closegraph();
  return(0);
}
```

A háromszögek gyakorlati részének algoritmusa

```
# include <iostream>
# include <stdio.h>
# include <graphics.h>

int main() {
int f=320,v=380,i,gd,gm,page=0,bill;
float x=0,y=0,a=0,b=0,c=0;
char st[10]="";
const float pi=3.1415926;
gd=IBM8514;gm=IBM8514HI;
initgraph(&gd,&gm,"");

while (bill!=27) {
  setactivepage(page);
  cleardevice();setbkcolor(15);settextstyle(3,0,4);setcolor(8);
  outtextxy(20,10,"Háromszög transzformációk");
  settextstyle(3,0,2);outtextxy(435,17,"(Az paraméterek értékei mindig egészre vannak kerekítve!)");
  x=(float)(v-20)/6;y=(float)110-(float)(f-80)/6;                              // x, y(magasság) koordináták
  a=100;b=(float)sqrt(x*x+y*y);c=(float)sqrt((100-x)*(100-x)+y*y);            // a, b, c oldalak
  int gamma=((asin(y/b))/pi)*180;                                             // szögek
  int beta= ((asin(y/c))/pi)*180;
  if (gamma<0) gamma=90;
  if (beta<0) beta=90;
  int alfa=180-gamma-beta;
  if (v==320) {beta=gamma; b=c;}
  if (beta==60 && gamma==60) {b=100;c=100;}
  if (b==100 && c==100) {beta=60; gamma=60;}
  if (b==c) beta=gamma; if (beta==gamma) b=c;

  setcolor(6);setlinestyle(0,0,2);                                            // háromszög rajzolás
  moveto(20,740);lineto(620,740);lineto(v,f);lineto(20,740);
  outtextxy(320,741,"a");
  outtextxy((v+20)/2-15,(f+740)/2-25,"b");
  outtextxy((v+620)/2+5,(f+740)/2-25,"c");

  if (f<=610) {                                                               // szög érték kiírás
    arc(v,f,180+gamma,360-beta,65);
    arc(620,740,180-beta,180,65);
    arc(20,740,0,gamma,65);
    if (alfa==90) circle(v+32*cos((beta+45)*pi/180),f+32*sin((beta+45)*pi/180),2);
       else outtextxy(v+18*cos((beta+alfa/2)*pi/180)-v/25,f+35*sin((beta+alfa/2)*pi/180),
            strcat(itoa(alfa,st,10),"°"));
     if (beta==90)  circle(592,712,2);
        else outtextxy(565,715,strcat(itoa(beta,st,10),"°"));
     if (gamma==90) circle(48,712,2);
        else outtextxy(45,715,strcat(itoa(gamma,st,10),"°"));
   }
  setcolor(6);outtextxy(680,50, "Paraméterek:");
  setcolor(8);outtextxy(680,80, "a oldal      = ");setcolor(4);outtextxy(790,80,itoa(a,st,10));        // oldalak kiírása
  setcolor(8); outtextxy(680,110,"b oldal      = ");setcolor(4);outtextxy(790,110,itoa(b,st,10));
  setcolor(8); outtextxy(680,140,"c oldal      = ");setcolor(4);outtextxy(790,140,itoa(c,st,10));
  setcolor(8); outtextxy(680,170,"Magasság = ");setcolor(4);outtextxy(790,170,itoa(y,st,10));          // magasság kiírása
  int k=a+(int)b+(int)c;int t=((int)a*(int)y)/2;
  setcolor(8); outtextxy(680,200,"Kerület      = ");setcolor(4);outtextxy(790,200,itoa(k,st,10));     // kerület kiírása
  setcolor(8); outtextxy(680,230,"Terület      = ");setcolor(4);outtextxy(790,230,itoa(t,st,10));     // terület kiírása
  setcolor(8); outtextxy(680,260,"Alfa           = ");itoa(alfa,st,10);strcat(st,"°");setcolor(4);outtextxy(790,260,st);
  setcolor(8); outtextxy(680,290,"Beta           = ");itoa(beta,st,10);strcat(st,"°");setcolor(4);outtextxy(790,290,st);
  setcolor(8); outtextxy(680,320,"Gamma      = ");itoa(gamma,st,10);strcat(st,"°");setcolor(4);outtextxy(790,320,st);

  if (alfa<90 && beta<90 && gamma<90) {setcolor(4);outtextxy(680,350,"Hegyesszögű");}              // háromszög fajtái
     else {setcolor(8);outtextxy(680,350,"Hegyesszögű"); }
  if (alfa>90 || beta>90 || gamma>90) {setcolor(4);outtextxy(680,380,"Tompaszögű");}
     else {setcolor(8);outtextxy(680,380,"Tompaszögű"); }
  if (alfa==90 || beta==90 || gamma==90) {setcolor(4);outtextxy(680,410,"Derékszögű");}
     else {setcolor(8);outtextxy(680,410,"Derékszögű");    }
  if ((int)a==(int)b || (int)a==(int)c || (int)b==(int)c) { setcolor(4);outtextxy(680,440,"Egyenlőszárú");}
     else {setcolor(8);outtextxy(680,440,"Egyenlőszárú");  }
```

```
    if (alfa==beta && beta==gamma) {setcolor(4);outtextxy(680,470,"Egyenlő oldalú");}
        else {setcolor(8);outtextxy(680,470,"Egyenlő oldalú");  }

    setcolor(6);outtextxy(680,530,"Egér és billentyűzet funkciók:");
    setcolor(8);
    outtextxy(680,560,"Az egérrel a csúcs vezérelhető.");
    outtextxy(680,590,"G: Balra");outtextxy(680,620,"H: Jobbra");
    outtextxy(680,650,"Z: Fel");outtextxy(680,680,"B: Le");
    outtextxy(680,710,"Escape: Kilép");
    setcolor(4); circle(v,f,4);                                  // csúcs pont

    setcolor(8);setlinestyle(0,0,1);
    line(660,55,660,740);line(670,515,1000,515);line(20,50,1000,50);    // határoló vonalak
    setcolor(8);setlinestyle(0,0,2);rectangle(940,7,1000,35);          //kilépés egérrel
    setcolor(4);outtextxy(950,10,"Kilép");setlinestyle(0,0,1);
    if (v>=620) v=620;                                       // kezdő és végső érték váltása
    if (v<=20)  v=20;
    if (f<=80)  f=80;
    if (f>=728) f=728;
        if (balgomb) {                                           // egérkezelés
             if (egerx>=940 && egerx<=1000 && egery>=7 && egery<=35) bill=27;
             if (egerx<=20) egerx=20;
             if (egerx>=620) egerx=620;
             if (egery<=80) egery=80;
             if (egery>=728) egery=728;
               v=integer(egerx);
               f=integer(egery);
        }
    setvisualpage(page);
    page = 1-page;
    if (kbhit()) {                          // billentyűzet kezelés
      bill=getch();
      switch (bill) {
        case 103: if (v!=20) v-=6; break;       // g balra
        case 104: if (v!=620) v+=6; break;      // h jobbra
        case 122: f-=6; break;                  // z fel
        case 98:  f+=6; break;                  // b le
      }
    }
  }
  closegraph();
  return(0);
}
```

A négyszögek elméleti részének algoritmusa

```
# include <iostream>
# include <stdio.h>
# include <graphics.h>

int main() {
  int i,gd,gm,page=0,bill;
  char st[10];
  gd=IBM8514;gm=IBM8514HI;
  initgraph(&gd,&gm,"");

  while (bill!=27) {
    setactivepage(page);
    setbkcolor(15); cleardevice();settextstyle(3,0,4);setcolor(8);
    outtextxy(240,15,"A négyszög definíciója, jellemzői és fajtái");
    settextstyle(3,0,2);outtextxy(10,60,"Definícó:");
    setcolor(6);
    outtextxy(90,60,"Zárt síkidom, aminek négy csúcsa (szöge) van, és négy szakasz határolja, amik csak végpontjukban találkoznak.");
    setcolor(8);outtextxy(10,100,"Jellemzői:");
    setcolor(6);outtextxy(100,100,"Szögeinek összege 360°,    Kerülete: a+b+c+d,    Területe: fajtánként különbözik a meghatározás.");
    setcolor(8);outtextxy(10,140,"Fajtái:");
    setcolor(4);outtextxy(50,370,"Konvex négyszög");outtextxy(300,370,"Konkáv négyszög");
    outtextxy(600,370,"Négyzet");outtextxy(840,370,"Téglalap");
    setcolor(4);outtextxy(70,700,"Paralelogramma");outtextxy(355,700,"Rombusz");
    outtextxy(600,700,"Deltoid");outtextxy(865,700,"Trapéz");
    setcolor(8);
    outtextxy(40,395,"Minden szöge < 180°");outtextxy(300,395,"Egy szöge > 180°");
    outtextxy(515,395,"Oldalai és szögei egyenlőek");outtextxy(795,395,"Minden szöge = 90°");
    outtextxy(25,725,"Szemközti oldalai egyenlőek");outtextxy(320,725,"Oldalai egyenlőek");
    outtextxy(495,725,"Két-két szomszédos oldal egyenlő");outtextxy(795,725,"Az 'a' és 'c' párhuzamos");

    setcolor(6);setlinestyle(0,0,2);settextstyle(3,0,2);                          // konvex négyszög
    moveto(60,340);lineto(30,200);lineto(230,180);lineto(190,340);lineto(60,340);
    outtextxy(125,345,"a");outtextxy(30,260,"b");outtextxy(120,160,"c");outtextxy(220,250,"d");
    moveto(260,340);lineto(480,130);lineto(440,280);lineto(490,340);lineto(260,340);      // konkáv négyszög
    outtextxy(390,345,"a");outtextxy(350,210,"b");outtextxy(470,180,"c");outtextxy(470,280,"d");
    setlinestyle(0,0,1);line(1,1,1,1);arc(440,280,75,310,55);outtextxy(390,270,">180°");
    setcolor(6);setlinestyle(0,0,2);settextstyle(3,0,2);                          // négyzet
    moveto(550,340);lineto(550,170);lineto(720,170);lineto(720,340);lineto(550,340);
    outtextxy(630,345,"a");outtextxy(530,240,"a");outtextxy(630,140,"a");outtextxy(725,240,"a");
    setlinestyle(0,0,1);line(1,1,1,1);arc(550,340,0,90,45);outtextxy(555,310,"90°");
    outtextxy(610,220,"K = 4*a");outtextxy(610,250,"T = a*a");
    setcolor(6);setlinestyle(0,0,2);settextstyle(3,0,2);                         // téglalap
    moveto(770,340);lineto(770,170);lineto(1000,170);lineto(1000,340);lineto(770,340);
    outtextxy(880,345,"a");outtextxy(755,240,"b");outtextxy(880,140,"a");outtextxy(1005,240,"b");
    setlinestyle(0,0,1);line(1,1,1,1);arc(770,340,0,90,45);outtextxy(775,310,"90°");
    outtextxy(840,220,"K = 2*a+2*b");outtextxy(840,250,"T = a*b");
    setcolor(6);setlinestyle(0,0,2);settextstyle(3,0,2);                         // paralelogramma
    moveto(40,670);lineto(10,500);lineto(230,500);lineto(260,670);lineto(40,670);
    outtextxy(145,675,"a");outtextxy(8,570,"b");outtextxy(110,470,"a");outtextxy(250,570,"b");outtextxy(130,570,"m");
    circle(162,658,1);
    setlinestyle(0,0,1);line(150,670,150,500);arc(150,670,0,90,30);
    arc(40,670,0,100,45); arc(10,500,280,360,45); arc(230,500,180,280,45); arc(260,670,100,180,45);
    setlinestyle(0,0,2);circle(210,520,4);line(212,520,215,516);line(212,520,215,523);
    setlinestyle(0,0,2);circle(55,655,4);line(57,655,60,651);line(57,655,60,658);
    outtextxy(25,505,"ß");outtextxy(235,640,"ß");
    outtextxy(10,435,"K = 2*a + 2*b      T = a*m");
    setcolor(6);setlinestyle(0,0,2);settextstyle(3,0,2);                                   // rombusz
    moveto(390,670);lineto(320,570);lineto(390,470);lineto(460,570);lineto(390,670);
    outtextxy(333,605,"a");outtextxy(340,500,"a");outtextxy(430,500,"a");outtextxy(437,605,"a");
    outtextxy(615,570,"n");outtextxy(650,505,"m");
    setlinestyle(0,0,1);line(390,670,390,470);line(320,570,460,570);
    outtextxy(305,435,"K = 4*a      T = (n*m)/2");
    setcolor(6);setlinestyle(0,0,2);settextstyle(3,0,2);                                   // deltoid
    moveto(630,670);lineto(560,530);lineto(630,470);lineto(700,530);lineto(630,670);
    outtextxy(572,580,"a");outtextxy(580,480,"b");outtextxy(670,480,"b");outtextxy(678,580,"a");
    outtextxy(375,520,"n");outtextxy(410,545,"m");
    setlinestyle(0,0,1);line(630,670,630,470);line(560,530,700,530);
```

```
    outtextxy(540,435,"K = 2*a + 2*b    T = (n*m)/2");
    setcolor(6);setlinestyle(0,0,2);settextstyle(3,0,2);                                        // trapéz
    moveto(780,670);lineto(790,500);lineto(920,500);lineto(1000,670);lineto(780,670);
    outtextxy(880,675,"a");outtextxy(770,570,"b");outtextxy(850,470,"c");outtextxy(965,570,"d");outtextxy(850,570,"m");
    circle(882,658,1);
    setlinestyle(0,0,1);line(870,670,870,500);arc(870,670,0,90,30);
    outtextxy(780,435,"K=a+b+c+d     T=((a+c)*m)/2");

    setcolor(8);setlinestyle(0,0,2);rectangle(940,7,1000,35);                        //kilépés egérrel
    setcolor(4);outtextxy(950,10,"Kilép");setlinestyle(0,0,1);

    if (balgomb)
      if (egerx>=940 && egerx<=1000 && egery>=7 && egery<=35) bill=27;          // egérkezelés
    setvisualpage(page);
    page = 1-page;
    if (kbhit())  bill=getch();                    // billentyűzet kezelés
  }
  closegraph();
  return(0);
}
```

A négyszögek gyakorlati részének algoritmusa

```
# include <iostream>
# include <stdio.h>
# include <graphics.h>

 typedef struct pont
 {   double x,y;
     } PONT;

int main() {
  int f=140,v=110,i,v2=530,f2=230,gd,gm,page=0,bill,ap,mx;
  float alfa,beta,delta,gamma,szog;
  float x=0,y=0,x2=0,y2=0,a=0,b=0,c=0,d=0;
  char st[10]="";
  const float pi=3.1415926;
  gd=IBM8514;gm=IBM8514HI;
  initgraph(&gd,&gm,"");
  PONT pontok[2] = {f,v,v2,f2};

  while (bill!=27) {
    setactivepage(page);
    cleardevice();setbkcolor(15);settextstyle(3,0,4);setcolor(8);
    outtextxy(20,10,"Négyszög transzformációk");
    settextstyle(3,0,2);outtextxy(420,17,"(Az paraméterek értékei mindig egészre vannak kerekítve!)");
    x=(float)(v-20)/6;y=(float)110-(float)(f-80)/6;                          // x, y koordináták számítása
    x2=(float)(v2-20)/6;y2=(float)110-(float)(f2-80)/6;
     a=50; b=(float)sqrt((x-25)*(x-25)+y*y);                                  // a, b, c, d oldalak számítása
    c=(float)sqrt((x-x2)*(x-x2)+(y-y2)*(y-y2));
    d=(float)sqrt((75-x2)*(75-x2)+y2*y2);
    if (x>25) alfa =((asin(y/b))/pi)*180; else alfa =180-(((asin(y/b))/pi)*180);      // szögek számítása
    if (x==25) alfa =90; alfa=integer(alfa);
         szog=integer(((asin((y-y2)/c))/pi)*180);
    if (x<x2) beta=180-alfa-szog;
    if (x>x2 && y>y2) beta=szog-alfa;
    if (x>x2 && y<y2) beta=360+szog-alfa; beta=integer(beta);
    if (x2<75) delta =((asin(y2/d))/pi)*180; else delta =180-(((asin(y2/d))/pi)*180);
    if (x2==75) delta =90;delta=integer(delta);
    gamma=integer(360-alfa-beta-delta);
    int k=a+(int)b+(int)c+(int)d;                                              // kerület,
    int t=((x2-x)*(y+y2)+(75-x2)*y2+(x-25)*y)/2;                              // terület

    outtextxy(850,270,"Megfelelő idom:");                                      // négyszög fajtái
    setcolor(4);outtextxy(880,300,"Piros szín!");
    setcolor(8);outtextxy(680,270,"Négyzet");
    outtextxy(680,300,"Téglalap");
    outtextxy(680,330,"Paralelogramma");
    outtextxy(680,360,"Rombusz");
    outtextxy(680,390,"Deltoid");
    outtextxy(680,420,"Trapéz");
    setcolor(4);outtextxy(680,450,"Konvex négyszög");
    if (beta==180 || gamma==180) {setcolor(4);outtextxy(680,480,"Ez nem négyszög, hanem háromszög!!!");
                                  setcolor(8);outtextxy(680,450,"Konvex sokszög               ");
    }
    if (beta>180 || gamma>180) {setcolor(4);outtextxy(680,450,"Konkáv négyszög");}
    if (y==y2) {     beta=180-alfa;gamma=180-delta;t=((50+c)*y)/2;
                     setcolor(4);outtextxy(680,420,"Trapéz");
    }
    if (alfa+delta==180) {setcolor(4);outtextxy(680,420,"Trapéz");}
    if (alfa+delta==180 && (beta==90 || gamma==90)) {setcolor(4);outtextxy(680,420,"Derékszögű trapéz"); }
    if (alfa+delta==180 && (alfa==90 || delta==90)) {setcolor(4);outtextxy(680,420,"Derékszögű trapéz"); }
    if (y==y2 && (alfa==90 || delta==90)) { beta=180-alfa;gamma=180-delta;t=((50+c)*y)/2;
                     setcolor(4);outtextxy(680,420,"Derékszögű trapéz");
    }
```

```
if (a==integer(b) && integer(c)==integer(d) && gamma!=180) {
                beta=delta; alfa=360-gamma-beta-delta;
                setcolor(4);outtextxy(680,390,"Deltoid");
}
        if (a==integer(d) && integer(b)==integer(c)&& beta!=180) {
                gamma=alfa;delta=360-alfa-beta-gamma;
                setcolor(4);outtextxy(680,390,"Deltoid");
        }
        if (a==integer(d) && integer(b)==integer(c)&& beta>180) {
                gamma=alfa;delta=360-alfa-beta-gamma;
                setcolor(4);outtextxy(680,390,"Konkáv deltoid");
        }
        if (a==integer(b) && integer(d)==integer(c)&& gamma>180) {
                beta=delta;alfa=360-beta-gamma-delta;
                setcolor(4);outtextxy(680,390,"Konkáv deltoid");
        }
if (a==integer(b) && a==integer(c) && a==integer(d) && (alfa==gamma || alfa+1==gamma || alfa-1==gamma)) {
                gamma=alfa;beta=180-alfa;delta=beta;t=50*y;
                setcolor(4);outtextxy(680,360,"Rombusz");
}
if (50==integer(c) && integer(b)==integer(d) && (alfa==gamma || alfa+1==gamma || alfa-1==gamma)) {
                gamma=alfa;beta=180-alfa;delta=beta;t=50*y;
                setcolor(4);outtextxy(680,330,"Paralelogramma");
}
if (50==integer(c) && integer(b)==integer(d) && (alfa==90 || beta==90)) {
                alfa=90;beta=90;gamma=90;delta=90;t=50*integer(b);
                setcolor(4);outtextxy(680,300,"Téglalap");
}
if (a==integer(b) && a==integer(c) && a==integer(d) && (alfa==90 || beta==90)) {
                 alfa=90;beta=90;gamma=90;delta=90;t=2500;
                 setcolor(4);outtextxy(680,270,"Négyzet");
}
setcolor(6);setlinestyle(0,0,2);                                             // négyszog rajzolás
moveto(170,740);lineto(v,f);lineto(v2,f2);lineto(470,740);lineto(170,740);
outtextxy(320,741,"a");
outtextxy((v+170)/2-15,(f+740)/2-25,"b");
outtextxy((v+v2)/2-8,(f+f2)/2-25,"c");
outtextxy((v2+470)/2+5,(f2+740)/2-25,"d");
outtextxy(135,720,"alfa");outtextxy(480,720,"delta");
outtextxy(v-40-(v-320)/15,f-35,"beta");outtextxy(v2-(v2-320)/15,f2-35,"gamma");


setcolor(6); outtextxy(680, 50, "Paraméterek:");
setcolor(8); outtextxy(680, 80,"a oldal  = ");setcolor(4);outtextxy(770,80,itoa(a,st,10));   // oldalak kiírása
setcolor(8); outtextxy(680,110,"b oldal  = ");setcolor(4);outtextxy(770,110,itoa(b,st,10));
setcolor(8); outtextxy(680,140,"c oldal  = ");setcolor(4);outtextxy(770,140,itoa(c,st,10));
setcolor(8); outtextxy(680,170,"d oldal  = ");setcolor(4);outtextxy(770,170,itoa(d,st,10));  // szögek kiírása
setcolor(8); outtextxy(870, 80,"Alfa      = ");itoa(alfa,st,10);strcat(st,"°");setcolor(4);outtextxy(960, 80,st);
setcolor(8); outtextxy(870,110,"Beta      = ");itoa(beta,st,10);strcat(st,"°");setcolor(4);outtextxy(960,110,st);
setcolor(8); outtextxy(870,140,"Gamma = ");     itoa(gamma,st,10);strcat(st,"°");setcolor(4);outtextxy(960,140,st);
setcolor(8); outtextxy(870,170,"Delta     = ");itoa(delta,st,10);strcat(st,"°");setcolor(4);outtextxy(960,170,st);

setcolor(8); outtextxy(680,200,"Kerület  = ");setcolor(4);outtextxy(770,200,itoa(k,st,10));       // kerület kiírása
setcolor(8); outtextxy(680,230,"Terület  = ");setcolor(4);outtextxy(770,230,itoa(t,st,10));       // terület kiírása

setcolor(6);outtextxy(680,530,"Egér és billentyűzet funkciók:");
outtextxy(680,590,"Bal oldali pont:");outtextxy(860,590,"Jobb oldali pont:");
setcolor(8);
outtextxy(680,555,"Az egérrel a csúcsok vezérelhetők.");
outtextxy(680,615,"S: Balra");outtextxy(860,615,"G: Balra");
outtextxy(680,640,"D: Jobbra");outtextxy(860,640,"H: Jobbra");
outtextxy(680,665,"E: Fel"); outtextxy(860,665,"Z: Fel");
```

```
    outtextxy(680,690,"X: Le");outtextxy(860,690,"B: Le");
    outtextxy(680,715,"Escape: Kilép");

    setcolor(4); circle(v,f,4);circle(v2,f2,4);                      // csúcs pontok
    setcolor(8);setlinestyle(0,0,1);
    line(665,55,665,740);line(20,50,1000,50);                        // határoló vonalak
    line(670,515,1000,515);line(670,260,1000,260);

    setcolor(8);setlinestyle(0,0,2);rectangle(940,7,1000,35);        //kilépés egérrel
    setcolor(4);outtextxy(950,10,"Kilép");setlinestyle(0,0,1);
    if (v>=317) v=317;        // kezdő és végső pontok
    if (v<=20)  v=20;
    if (f<=80)  f=80;
    if (f>=734) f=734;
    if (v2>=620) v2=620;
    if (v2<=323)  v2=323;
    if (f2<=80)  f2=80;
    if (f2>=734) f2=734;
            (pontok[0].x)=v;
            (pontok[0].y)=f;
            (pontok[1].x)=v2;
            (pontok[1].y)=f2;
     if (!balgomb) ap = getactivepoint((pont2d*)pontok,2,5);          // egér kezelés
     if (balgomb && egerx>=940 && egerx<=1000 && egery>=7 && egery<=35) bill=27;
     if (ap >= 0 && balgomb)  {
            if (egerx<=20 && ap==0) egerx=20;
            if (egerx>=317 && ap==0) egerx=317;
            if (egerx>=620 && ap==1) egerx=620;
            if (egerx<=323 && ap==1) egerx=323;
            if (egery<=80) egery=80;
            if (egery>=734) egery=734;
            pontok[ap].x = egerx;
            pontok[ap].y = egery;
            v=integer(pontok[0].x);
            f=integer(pontok[0].y);
            v2=integer(pontok[1].x);
            f2=integer(pontok[1].y);
     }
    setvisualpage(page);
    page = 1-page;
    if (kbhit()) {                                          // billentyűzet kezelés
      bill=getch();
      switch (bill) {
        case 115: if (v!=20)   v-=2; break;         // s balra       bal oldali pont
        case 100: if (v!=317 ) v+=2; break;         // d jobbra
        case 101: if (f!=80)   f-=2; break;         // e fel
        case 120: if (f!=734)  f+=2; break;         // x le
        case 103: if (v2!=323)  v2-=2; break;       // g balra      jobb oldali pont
        case 104: if (v2!=620) v2+=2; break;        // h jobbra
        case 122: if (f2!=80) f2-=2; break;         // z fel
        case 98:  if (f2!=734) f2+=2; break;        // b le
      }
    }
  }
  closegraph();
  return(0);
}
```

A Pitagorasz-tétel elméleti részének algoritmusa

```
# include <iostream>
# include <stdio.h>
# include <graphics.h>

int main() {
int gd,gm,page=0,bill;
float pi=3.1415926;
char st[10];
gd=IBM8514;gm=IBM8514HI;
initgraph(&gd,&gm,"");

while (bill!=27) {
  setactivepage(page);
  setbkcolor(15); cleardevice(); settextstyle(3,0,4);setcolor(8);
  outtextxy(180,15,"A Pitagorasz-tétel bemutatása és bizonyítása");
  settextstyle(3,0,2);
  outtextxy(20,60,"Tétel:");
  setcolor(6);
  outtextxy(80,60," Minden derékszögű háromszögre igaz, hogy a befogók négyzeteinek összege egyenlő az átfogó négyzetével.");
  setcolor(8);outtextxy(20,90,"A tétel képlettel:");
  settextstyle(3,0,3);setcolor(4);outtextxy(160,90,"a  + b  = c");
  settextstyle(3,0,2);outtextxy(173,80,"2");outtextxy(218,80,"2");outtextxy(265,80,"2");
  setcolor(8);outtextxy(280,90,", ahol 'a' az egyik, 'b' a másik befogó és 'c' az átfogó (lásd az 1.ábrát).");

  setcolor(6);setlinestyle(0,0,2);                                            // háromszög és négyzetek
  moveto(120,715);lineto(370,715);lineto(370,475);lineto(120,475);lineto(120,715);
  moveto(120,475);lineto(182,367);lineto(370,475);
  moveto(120,475);lineto(12,413);lineto(74,305);lineto(182,367);
  lineto(290,179);lineto(478,287);lineto(370,475);
  outtextxy(135,405,"a");outtextxy(275,395,"b");outtextxy(240,478,"c");circle(185,387,2);
  circle(325,465,4);line(327,465,330,461);line(327,465,330,468);
  outtextxy(137,450,"ß");
  setlinestyle(0,0,1);line(1,1,1,1);arc(120,475,0,60,40);arc(182,367,240,330,40);arc(370,475,150,180,60);  // körívek
  settextstyle(3,0,3);setcolor(4);outtextxy(80,380,"a");outtextxy(310,310,"b");outtextxy(245,570,"c");
  settextstyle(3,0,2);outtextxy(93,370,"2");outtextxy(323,300,"2");outtextxy(258,560,"2");
  setcolor(8);outtextxy(200,735,"1. ábra");setcolor(8);outtextxy(645,735,"2. ábra");
  setcolor(6);setlinestyle(0,0,2);                                                 // bizonytás
  moveto(520,420);lineto(520,140);lineto(800,140);lineto(800,420);lineto(520,420);
  moveto(520,730);lineto(520,450);lineto(800,450);lineto(800,730);lineto(520,730);
  line(608,420,608,140);line(520,228,800,228);line(608,420,520,228);line(608,140,800,228);
  moveto(520,642);lineto(608,450);lineto(800,538);lineto(712,730);lineto(520,642);
  setfillstyle(1,11);floodfill(540,200,6);floodfill(620,240,6);floodfill(620,550,6);       // színezés
  setfillstyle(1,9);floodfill(780,150,6);floodfill(540,400,6);floodfill(620,200,6);floodfill(600,240,6);
  floodfill(540,460,6);floodfill(790,460,6);floodfill(540,720,6);floodfill(790,720,6);
  outtextxy(560,115,"a");outtextxy(690,115,"b");
  outtextxy(560,425,"a");outtextxy(690,425,"b");
   outtextxy(610,731,"b");outtextxy(760,731,"a");
   outtextxy(500,170,"a");outtextxy(500,310,"b");
   outtextxy(500,560,"b");outtextxy(500,680,"a");
   setbkcolor(9);outtextxy(690,150,"c");outtextxy(570,310,"c");
   outtextxy(690,460,"c");outtextxy(550,520,"c");
   outtextxy(600,685,"c");outtextxy(765,620,"c");
   setbkcolor(11);
   settextstyle(3,0,3);setcolor(4);outtextxy(560,170,"a");outtextxy(690,310,"b");outtextxy(660,570,"c");
   settextstyle(3,0,2);outtextxy(573,160,"2");outtextxy(703,300,"2");outtextxy(673,560,"2");
   setcolor(6);circle(624,406,2);circle(540,636,2);
   setlinestyle(0,0,1);line(1,1,1,1);arc(608,420,0,90,40);arc(520,642,332,62,40);

   setcolor(8);setbkcolor(15);
   outtextxy(820,140,"Mindkét négyzet 4 db");outtextxy(820,170,"azonos területű három-");
   outtextxy(820,200,"szöget tartalmaz:");outtextxy(820,260,"Ha ezeket elvesszük a");
   outtextxy(820,290,"négyzetekből az egyik-");outtextxy(820,320,"-ben                  a másik-");
   outtextxy(820,350,"ban pedig         területet");outtextxy(820,380,"kapunk.");
   outtextxy(820,420,"Mivel a két erdeti négy-");outtextxy(820,450,"zet azonos területű volt,");
   outtextxy(820,480,"így a két maradék terület");outtextxy(820,510,"egyenlő, azaz:");
   setcolor(4);outtextxy(820,230,"t = (a*b)/2");
   outtextxy(870,320,"a*a+b*b");outtextxy(910,350,"c*c");
```

```
    settextstyle(3,0,3);outtextxy(820,560,"a  + b  = c");
    settextstyle(3,0,2);outtextxy(833,550,"2");outtextxy(878,550,"2");outtextxy(925,550,"2");
    setbkcolor(15); setcolor(8);setlinestyle(0,0,2);rectangle(940,7,1000,35);          //kilépés egérrel
    setcolor(4);outtextxy(950,10,"Kilép");setlinestyle(0,0,1);
    if (balgomb)
      if (egerx>=940 && egerx<=1000 && egery>=7 && egery<=35) bill=27;              // egérkezelés

    setvisualpage(page);
    page = 1-page;
    if (kbhit())  bill=getch();                                                // billentyűzet kezelés
 }
  closegraph();
  return(0);
}
```

A Thalész-tétel elméleti részének algoritmusa

```
# include <iostream>
# include <stdio.h>
# include <graphics.h>

int main() {
int gd,gm,page=0,bill;
float pi=3.1415926;
char st[10];
gd=IBM8514;gm=IBM8514HI;
initgraph(&gd,&gm,"");

while (bill!=27) {
  setactivepage(page);
  setbkcolor(15); cleardevice(); settextstyle(3,0,4);setcolor(8);
  outtextxy(200,15,"A Thalész-tétel bemutatása és bizonyítása");
  settextstyle(3,0,2);outtextxy(20,60,"Tétel:"); outtextxy(20,150,"A tétel ábrával:");
  setcolor(6);
  outtextxy(80,60,"Ha egy kör átmérőjének A és B végpontját összekötjük a körív A-tól és B-től különböző tetszőleges C pontjával,");
  outtextxy(80,90,"akkor az ABC háromszög C-nél lévő szöge derékszög lesz.");
  setcolor(6);setlinestyle(0,0,2);                                                          // háromszög
  moveto(30,400);lineto(450,400);lineto(350,222);lineto(30,400);
  setcolor(5);moveto(450,400);lineto(70,277);lineto(30,400);circle(80,300,2); outtextxy(50,250,"C'");
  setlinestyle(0,0,1);line(1,1,1,1);arc(70,277,252,341,50);
  setlinestyle(0,0,2);setcolor(1);moveto(450,400);lineto(210,607);lineto(30,400);circle(213,580,2); outtextxy(210,620,"C''");
  setlinestyle(0,0,1);line(1,1,1,1);arc(210,607,40,130,50);
  setlinestyle(0,0,2);setcolor(6);
  outtextxy(10,390,"A");outtextxy(455,390,"B");outtextxy(350,200,"C");outtextxy(230,405,"O");circle(345,250,2);
  circle(65,390,4);line(67,390,70,386);line(67,390,70,393);
  outtextxy(420,375,"ß");
  setlinestyle(0,0,1);circle(240,400,210);line(240,396,240,404);                             // körívek
  arc(30,400,0,30,60);arc(450,400,120,180,50);arc(350,222,210,300,50);
  setlinestyle(0,0,2); moveto(530,400);lineto(950,400);lineto(850,222);lineto(530,400);      // bizonytás
  outtextxy(510,390,"A");outtextxy(955,390,"B");outtextxy(850,200,"C");outtextxy(730,405,"O");
  outtextxy(625,401,"r");outtextxy(835,401,"r");outtextxy(780,300,"r");
  circle(565,390,4);line(567,390,570,386);line(567,390,570,393);                             // alfa
  circle(820,250,4);line(822,250,825,246);line(822,250,825,253);
  circle(743,375,4);line(745,375,748,371);line(745,375,748,378);
  outtextxy(685,360,"180°-2");
  outtextxy(920,375,"ß");outtextxy(845,240,"ß"); outtextxy(760,370,"180°-2ß");
  setlinestyle(0,0,1);line(740,400,850,222);arc(740,400,0,180,210);arc(740,400,0,58,90);arc(740,400,58,180,65);  // körívek
  arc(530,400,0,30,60);arc(950,400,120,180,50);arc(850,222,210,240,60);arc(850,222,240,300,50);
  setcolor(8); outtextxy(530,120,"Bizonyítás felhasználva, hogy a háromszög");
  outtextxy(530,150,"belső szögeinek összege 180° :");
  outtextxy(530,430,"Az ábrán két darab 'r' szárú AOC és COB egyenlőszárú");
  outtextxy(530,460,"háromszög található. A háromszögek csúcsszögei");
  outtextxy(530,490,"kiegészítő szögei egymásnak,     Így:");
   setcolor(6);outtextxy(650,530,"180° - 2   + 180° - 2ß = 180°");
   setlinestyle(0,0,2);
   circle(718,545,4);line(720,545,723,541);line(720,545,723,548);
   setcolor(8);outtextxy(530,560,"Az egyenlet tovább egyszerűsítve:");
   setcolor(8);outtextxy(530,720,"azaz a C pontnál lévő szög derékszög (90°).");
   setcolor(6);outtextxy(650,600,"180° - 2   - 2ß = 0°");
   setcolor(6);outtextxy(650,640,"2   + 2ß = 180°");
   circle(718,615,4);line(720,615,723,611);line(720,615,723,618);
   circle(665,655,4);line(667,655,670,651);line(667,655,670,658);
   setcolor(4);settextstyle(3,0,3);outtextxy(650,680,"   + ß = 90°");
   circle(658,697,6);line(662,697,666,692);line(662,697,666,701);
   settextstyle(3,0,2);setbkcolor(15);
   setcolor(8);rectangle(940,7,1000,35);                              //kilépés egérrel
   setcolor(4);outtextxy(950,10,"Kilép");setlinestyle(0,0,1);
   if (balgomb)
     if (egerx>=940 && egerx<=1000 && egery>=7 && egery<=35) bill=27;    // egérkezelés
   setvisualpage(page);
   page = 1-page;
   if (kbhit())  bill=getch();                                        // billentyűzet kezelés
 }
 closegraph();
 return(0);
}
```

A Pitagorasz- és Thalész-tétel gyakorlati részének algoritmusa

```
# include <iostream>
# include <stdio.h>
# include <graphics.h>

 int main() {
 int sz=0,i,x,y,a=1,b,gd,gm,page=0,bill;
 float pi=3.1415926;
 char st[10];
 gd=IBM8514;gm=IBM8514HI;
 initgraph(&gd,&gm,"");

 while (bill!=27) {
   setactivepage(page);
   cleardevice();setbkcolor(15); settextstyle(3,0,4);setcolor(8);
   outtextxy(100,15,"A Pitagorasz- és Thalész-tétel egész középponti szögekre");
   settextstyle(3,0,3);setcolor(4);outtextxy(20,470,"a  + b  = c");
   settextstyle(3,0,2);outtextxy(33,460,"2");outtextxy(78,460,"2");outtextxy(125,460,"2");
   settextstyle(3,0,2);setcolor(8);
   outtextxy(20,340,"Szemközti oldalak:");outtextxy(20,370,"'A' csúcs:  'a' oldal");
   outtextxy(20,400,"'B' csúcs:  'b' oldal");outtextxy(20,430,"'C' csúcs:  'c' oldal");
   setcolor(6);outtextxy(750,550,"Egér és billentyűzet funkciók:");setcolor(8);
   outtextxy(750,580,"Az egérrel a csúcs vezérelhető.");
   outtextxy(750,610,"G: Megállít vagy növel 1°-al.");
   outtextxy(750,640,"H: Megállít vagy csökkent 1°-al.");
   outtextxy(750,670,"Space, Enter: Indít v. iránytvált.");
   outtextxy(750,700,"Escape: Visszalép a menübe.");

   setcolor(6);setlinestyle(0,0,1);
   if (sz>360) sz=1;                                           // kezdő és végső sz érték váltása
   if (sz<0)  sz=359;

   x=150*cos(sz*pi/180);y=150*sin(sz*pi/180);                  //  mozgó csúcspont koordinátái
   circle(450,400,150);line(450,400,450+x,400-y);
   line(450,396,450,404);
   if(sz!=0 && sz!=1) arc(450,400,0,sz,50);
   setlinestyle(0,0,2);moveto(300,400); lineto(600,400);
   lineto(450+x,400-y);lineto(300,400);

   if (sz>0 && sz<180){ moveto(300,400);lineto(300,700);lineto(600,700);       // négyzetek
                        lineto(600,400);lineto(450+150+y,400+x-150);
                        lineto(450+x+y,400-y+x-150);lineto(450+x,400-y);
                        lineto(450+x-y,400-y-x-150);lineto(450-150-y,400-x-150);
                        lineto(300,400);
                        if (sz>12 && sz<168) {                                  //derékszög
                         setlinestyle(0,0,1);line(1,1,1,1);arc(450+x,400-y,180+sz/2,270+sz/2,30);
                         setlinestyle(0,0,2);circle(450+x-17*sin((45-sz/2)*pi/180),400-y+17*cos((45-sz/2)*pi/180),2);
                         setcolor(4); outtextxy(450+x-17*sin((45-(sz+360)/2)*pi/180)-8,400-y+17*cos((45-(sz+360)/2)*pi/180)-15,"C");}
                       }
  if (sz>180 && sz<360){moveto(300,400);lineto(300,100);lineto(600,100);
                        lineto(600,400);lineto(450+150-y,400-x+150);
                        lineto(450+x-y,400-y-x+150);lineto(450+x,400-y);
                        lineto(450+x+y,400-y+150+x);lineto(450-150+y,400+150+x);
                        lineto(300,400);
                        if (sz>192 && sz<348) {
                         setlinestyle(0,0,1);line(1,1,1,1);arc(450+x,400-y,sz/2-90,sz/2,30);
                         setlinestyle(0,0,2);circle(450+x-17*sin((45-(sz+180)/2)*pi/180),400-y+17*cos((45-(sz+180)/2)*pi/180),2);
                         setcolor(4);outtextxy(450+x-17*sin((45-(sz-180)/2)*pi/180)-10,400-y+17*cos((45-(sz-180)/2)*pi/180),"C");}
                        }
  setcolor(4);circle(450+x,400-y,4);
  outtextxy(450,402,"O");outtextxy(270,390,"A");outtextxy(620,390,"B");
```

```
    char fok[10]="";                                                                                                    //szögek
    if (sz%2==1) {strcat(fok,itoa(abs(89-abs(90-(float)sz/2)),st,10));strcat(fok,",5°");}
                 else {strcat(fok,itoa(abs(90-abs(90-(float)sz/2)),st,10));strcat(fok,"°");}
                    outtextxy(960,100,fok);
    strcpy(fok,"");strcat(fok,itoa(abs(90-(float)sz/2),st,10));
    if (sz%2==1) strcat(fok,",5°"); else strcat(fok,"°");   outtextxy(960,130,fok);
    outtextxy(960,160,"90°");
    strcpy(fok,"");strcat(fok,itoa(sz,st,10));strcat(fok,"°");outtextxy(960,190,fok);outtextxy(450,370,fok);
    char oldal[10]="";strcat(oldal,itoa(50*sin(sz*pi/360),st,10));strcat(oldal,",");                                 //oldalak
    int a1=(abs(500000*sin(sz*pi/360)))%10000;
    strcat(oldal,itoa(a1,st,10));strcat(oldal," egység");
    if (sz==60 || sz==300) strcpy(oldal,"25,0 egység");if (sz==0 || sz==360) strcpy(oldal,"0,0 egység");
    if (sz==180) strcpy(oldal,"50,0 egység");outtextxy(850,250,oldal);
    strcpy(oldal,"");strcat(oldal,itoa(abs(50*cos(sz*pi/360)),st,10));strcat(oldal,",");
    int b1=(abs(500000*cos(sz*pi/360)))%10000;
    strcat(oldal,itoa(b1,st,10));strcat(oldal," egység");
    if (sz==120 || sz==240) strcpy(oldal,"25,0 egység");if (sz==0 || sz==360) strcpy(oldal,"50,0 egység");
    if (sz==180) strcpy(oldal,"0,0 egység");outtextxy(850,280,oldal);
    outtextxy(850,310,"50 egység");
    outtextxy(750,400,"Középponti szög:  BOC szög");
    setcolor(8);outtextxy(750,100,"A csúcs - alfa szög:       ="); outtextxy(750,130,"B csúcs - beta szög:      =");
    outtextxy(750,160,"C csúcs - gamma szög:=");  outtextxy(750,190,"Középponti szög:         =");
    outtextxy(750,250,"'a' oldal: ="); outtextxy(750,280,"'b' oldal: =");
    outtextxy(750,310,"'c' oldal: =");

    settextstyle(3,0,2);setcolor(8);setlinestyle(0,0,2);rectangle(940,7,1000,35);                                //kilépés egérrel
    setcolor(4);outtextxy(950,10,"Kilép");setlinestyle(0,0,1);
    sz+=a;

  if (balgomb) {                                                                                  // egérkezelés
          if (egerx>=940 && egerx<=1000 && egery>=7 && egery<=35) bill=27;
          if (egerx>=290 && egerx<=610) {
              if (sz>0 && sz<180) sz=((acos(((float)egerx-300-150)/150))/pi)*180;
                 else  sz=360-((acos(((float)egerx-300-150)/150))/pi)*180;
          }
  }
    setvisualpage(page);
    page = 1-page;
    delay(30);
    if (kbhit()) {                                // billentyűzet kezelés
      bill=getch();
      switch (bill) {
        case 13: {if (a==0) a=b; else a*=-1; break;}     // enter
        case 32: {if (a==0) a=b; else a*=-1; break;}     // space
        case 103: {a=0;b=1;sz+=1; break;}                 // g betű
        case 104: {a=0;b=-1;sz+=-1; break;}               // h betű
      }
    }
  }
  closegraph();
  return(0);
}
```

Az ellenőrzés és tesztelés algoritmusa

```
# include <iostream>
# include <stdio.h>
# include <graphics.h>
# include <string.h>

int main() {
int gd,gm,page=0,bill,sor=0,k=1,i,j,pontszam,x,y,rn,ksz;
int tov=true;
char st[100],tes[1000][200],t1[1000][200];
int a[21][6]={0,0,0,0,0,0,                          // 20 kérdés paraméterei
              0,0,0,0,0,0,
              0,0,0,0,0,0,
              0,0,0,0,0,0,
              0,0,0,0,0,0,
              0,0,0,0,0,0,
              0,0,0,0,0,0,
              0,0,0,0,0,0,
              0,0,0,0,0,0,
              0,0,0,0,0,0,
              0,0,0,0,0,0,
              0,0,0,0,0,0,
              0,0,0,0,0,0,
              0,0,0,0,0,0,
              0,0,0,0,0,0,
              0,0,0,0,0,0,
              0,0,0,0,0,0,
              0,0,0,0,0,0,
              0,0,0,0,0,0,
              0,0,0,0,0,0,
              0,0,0,0,0,0};
FILE *t;
t=fopen("Teszt.txt","a");
t=fopen("Teszt.txt","r");
srand(time(NULL));
  while (!feof(t)) {                             // kérdés beolvasás file-ból
    sor+=1;
    fgets(t1[sor],200,t);
  }
  sor-=sor%7;
  int v[sor/7+1];                               // a 20 kérdés kisorsolása
  for (j=0; j<=sor/7; j++) v[j]=0;
  if (sor>=140) ksz=20; else ksz=sor/7;
     for (i=1; i<=ksz; i++)  {
          tov=true;
          while (tov)  {
            rn=rand()%(sor/7)+1;
            if (v[rn]!=1) {
               for (j=1; j<=7; j++) { strncat(tes[7*(i-1)+j],t1[7*(rn-1)+j],strlen(t1[7*(rn-1)+j])-1);
               }
               v[rn]=1; tov=false;
            }
          }
     }
fclose(t);
gd=IBM8514;gm=IBM8514HI;
initgraph(&gd,&gm,"");
while (bill!=27) {
  setactivepage(page);
  setbkcolor(15); cleardevice();

  settextstyle(3,0,4);setcolor(8);
  outtextxy(200,25,"Teszt és ellenőrzés a teljes tananyagból");
```

```
 settextstyle(3,0,2);
 outtextxy(100,70,"(A teszt 20 kérdést véletlenül generál egy több kérdéses adatbázisból, a sorrend megváltoztatásával.)");

 settextstyle(3,0,2);setcolor(4);
 if (sor<140 && sor>0) { outtextxy(10,110,"Az adatbázis a 20 kérdés helyett csak ");itoa(sor/7,st,10);
      strcat(st," kérdést tartalmaz, de a teszt így is működik!");outtextxy(325,110,st);}
 if (sor==0) outtextxy(10,110,"Az adatbázis üres, nincs benne kérdés!");
     setcolor(6);
 if (sor>=140) { outtextxy(560,725,"A teljes adatbázisban ");itoa(sor/7,st,10);strcat(st," darab kérdés van összesen.");
       outtextxy(745,725,st);}

 setcolor(6);outtextxy(560,455,"Egér és billentyűzet funkciók:");
 setcolor(8);
 outtextxy(560,485,"Az egérrel vezérelhetők a piros színű kapcsolók,");
 outtextxy(560,515,"valamint az 'A', 'B', 'C' és 'D', válaszok megjelölhetők.");
 outtextxy(560,545,"Vissza: A teszt kérdés sorszámának csökkentése.");
 outtextxy(560,575,"Tovább: A teszt kérdés sorszámának növelése.");
 outtextxy(560,605,"Rögzít: A megjelölt válasz véglegesítése.");
 outtextxy(590,635,"A válasz be van jelölve, még módosítható!");
           setcolor(4);setfillstyle(1,2);circle(570,648,8);floodfill(570,648,4);setcolor(8);
 outtextxy(590,665,"A válasz már nem módosítható!");
           setcolor(4);setfillstyle(1,4);circle(570,678,8);floodfill(570,678,4);setcolor(8);
 outtextxy(560,695,"Escape: Kilép");
 setcolor(6);setlinestyle(0,0,1);
 line(10,445,1000,445);line(10,140,1000,140);line(530,450,530,750);
 setlinestyle(0,0,2);

for (i=k; i<k+7; i++) {                                                    // kérdés és válasz kiírás
    strcat(itoa(k/7+1,st,10),". kérdés");setcolor(8);outtextxy(2,150,st);setcolor(6);
    if (i%7!=0) outtextxy(100,150+(i-k)*40,tes[i]);
}
setcolor(6);
if (a[k/7+1][5]==0) setfillstyle(1,2); else setfillstyle(1,4);             // válasz jelölés
if (a[k/7+1][1]==0) {rectangle(60,233,80,253); outtextxy(64,230,"A");}
                else {setcolor(4);circle(70,243,8);floodfill(70,243,4);setcolor(6);}
if (a[k/7+1][2]==0) {rectangle(60,273,80,293);outtextxy(64,270,"B");}
                else {setcolor(4);circle(70,283,8);floodfill(70,283,4);setcolor(6);}
if (a[k/7+1][3]==0) {rectangle(60,313,80,333); outtextxy(64,310,"C");}
                else {setcolor(4);circle(70,323,8);floodfill(70,323,4);setcolor(6);}
if (a[k/7+1][4]==0) {rectangle(60,353,80,373); outtextxy(64,350,"D");}
                else {setcolor(4);circle(70,363,8);floodfill(70,363,4);setcolor(6);}
 pontszam=0;
 for (i=1; i<21; i++) {
   setcolor(8);                                                            // találat számlálás és kiírás
   if (i<11) { strcat(itoa(i,st,10),". kérdés:");outtextxy(65,425+i*30,st);x=180;y=i; }
   if (i>10) { strcat(itoa(i,st,10),". kérdés:");outtextxy(300,425+(i-10)*30,st);x=420;y=i-10;   }
    setcolor(10);
   if (a[i][5]==1) {
     if (strcmp(tes[7*i],"A")==0 && a[i][1]==1){pontszam+=1; setcolor(4);outtextxy(x,425+y*30,"Helyes: A"); setcolor(8);  }
     if (strcmp(tes[7*i],"A")==0 && a[i][1]==0) outtextxy(x,425+y*30,"Helyes: A");
     if (strcmp(tes[7*i],"B")==0 && a[i][2]==1){pontszam+=1; setcolor(4);outtextxy(x,425+y*30,"Helyes: B"); setcolor(8);  }
     if (strcmp(tes[7*i],"B")==0 && a[i][2]==0) outtextxy(x,425+y*30,"Helyes: B");
     if (strcmp(tes[7*i],"C")==0 && a[i][3]==1){pontszam+=1; setcolor(4);outtextxy(x,425+y*30,"Helyes: C"); setcolor(8);  }
     if (strcmp(tes[7*i],"C")==0 && a[i][3]==0) outtextxy(x,425+y*30,"Helyes: C");
     if (strcmp(tes[7*i],"D")==0 && a[i][4]==1){pontszam+=1; setcolor(4);outtextxy(x,425+y*30,"Helyes: D"); setcolor(8);  }
     if (strcmp(tes[7*i],"D")==0 && a[i][4]==0) outtextxy(x,425+y*30,"Helyes: D");

   }
 }
 setcolor(6);outtextxy(400,403,"Eredmény: ");
 setcolor(4);itoa(pontszam,st,10);strcat(st," pont");outtextxy(500,403,st);
 itoa(100*((float)pontszam/ksz),st,10);strcat(st," %");outtextxy(600,403,st);
```

```
    setcolor(8);rectangle(940,7,1000,35);          //kilépés egérrel
    setcolor(4);outtextxy(950,10,"Kilép");
    setcolor(8);rectangle(60,400,130,430);         //előre egérrel
    setcolor(4);outtextxy(65,403,"Vissza");
    setcolor(8);rectangle(160,400,230,430);        //vissza egérrel
    setcolor(4);outtextxy(165,403,"Tovább");
    setcolor(8);rectangle(260,400,330,430);        //rögzít egérrel
    setcolor(4);outtextxy(265,403,"Rögzít");

    if (balgomb)  {                                                                                  // egérkezelés
      if (egerx>=940 && egerx<=1000 && egery>=7 && egery<=35) bill=27;
      if (egerx>=60 && egerx<=130 && egery>=400 && egery<=430 && k-7>0){ k-=7;balgomb=false;}       //vissza
      if (egerx>=160 && egerx<=230 && egery>=400 && egery<=430 && k+7<ksz*7){ k+=7;balgomb=false;}  //előre
      if (egerx>=260 && egerx<=330 && egery>=400 && egery<=430 && a[k/7+1][5]==0                    //rögzít
            && (a[k/7+1][1]==1 || a[k/7+1][2]==1 || a[k/7+1][3]==1 || a[k/7+1][4]==1 )){
                     a[k/7+1][5]=1;balgomb=false;}
      if (egerx>=60 && egerx<=80 && egery>=233 && egery<=253 && a[k/7+1][1]==0 && a[k/7+1][5]==0){      // válasz jelölés
          a[k/7+1][1]=1;a[k/7+1][2]=0;a[k/7+1][3]=0;a[k/7+1][4]=0;balgomb=false;}
      if (egerx>=60 && egerx<=80 && egery>=273 && egery<=293 && a[k/7+1][2]==0 && a[k/7+1][5]==0){
          a[k/7+1][1]=0;a[k/7+1][2]=1;a[k/7+1][3]=0;a[k/7+1][4]=0;balgomb=false;}
      if (egerx>=60 && egerx<=80 && egery>=313 && egery<=333 && a[k/7+1][3]==0 && a[k/7+1][5]==0){
          a[k/7+1][1]=0;a[k/7+1][2]=0;a[k/7+1][3]=1;a[k/7+1][4]=0;balgomb=false;}
      if (egerx>=60 && egerx<=80 && egery>=353 && egery<=373 && a[k/7+1][4]==0 && a[k/7+1][5]==0){
          a[k/7+1][1]=0;a[k/7+1][2]=0;a[k/7+1][3]=0;a[k/7+1][4]=1;balgomb=false;}
    }
    setvisualpage(page);
    page = 1-page;
    if (kbhit())  bill=getch();                        // billentyűzet kezelés
  }
  closegraph();
  return(0);
}
```

Az ellenőrzés és tesztelés kérdéseinek az adatbázisa

Derékszögű háromszög esetében, mi adja valamelyik hegyesszög sinusát?
- -
A szög melletti befogó és az átfogó hányadosa adja a szög sinusát.
A szöggel szemközti befogó és az átfogó hányadosa adja a szög sinusát.
A szöggel szemközti befogó és a szög melletti befogó hányadosa adja a szög sinusát.
Az átfogó és a szög melletti befogó hányadosa adja a szög sinusát.
B
Egység sugarú körre kiterjesztve mint értünk egy szög sinusán?
- -
Az egységnyi sugár vízszintes tengelyre eső merőleges vetületét.
Az aktuális szöghöz tartozó egységkör vetületet.
Az aktuális szöghöz tartozó egység körívet.
Az egységnyi sugár függőleges tengelyre eső merőleges vetületét.
D
Milyen célból lettek kiterjesztve a szögfüggvények egységkörre?
- -
Így az értelmezési tartománya nem 0° és 90°, hanem mínusz és plusz végtelen közé esik.
Így az értelmezési tartománya nem 0° és 90°, hanem 0° és plusz végtelen közé esik.
Így az értelmezési tartománya nem 0° és 90°, hanem 0° és 180° közé esik.
Így az értelmezési tartománya nem 0° és 90°, hanem 0° és 360° közé esik.
A
Hány fokonként periodikus (az értékkészlete ismétlődik) a sinus függvény?
- -
90°
360°
180°
540°
B
Milyen intervallumban mozog a sinus függvény értékkészlete?
- -
0 és +1.
Mínusz végtelen és +1.
-1 és +1.
-1 és plusz végtelen.
C
Mennyi 150° sinusa?
- -
Nulla.
+0,5
-1
-0,5
B
Mennyi -30° sinusa?
- -
Nulla.
+0,5
-1
-0,5
D
A következők kivétel nélkül mind a sinus függvény zérushelyei:
- -
0°, 90°, 180°, 270°.
0°, 180°, 330°, 540°.
0°, 180°, 360°, 540°.
0°, 150°, 360°, 540°.
C

Derékszögű háromszög esetében, mi adja valamelyik hegyesszög cosinusát?
- -
A szöggel szemközti befogó és a szög melletti befogó hányadosa adja a szög cosinusát.
A szöggel szemközti befogó és az átfogó hányadosa adja a szög cosinusát.
A szög melletti befogó és az átfogó hányadosa adja a szög cosinusát.
Az átfogó és a szög melletti befogó hányadosa adja a szög cosinusát.
C
Egység sugarú körre kiterjesztve mint értünk egy szög cosinusán?
- -
Az aktuális szöghöz tartozó egységkör vetületet.
Az egységnyi sugár vízszintes tengelyre eső merőleges vetületét.
Az aktuális szöghöz tartozó egység körívet.
Az egységnyi sugár függőleges tengelyre eső merőleges vetületét.
B
Hány fokonként periodikus (az értékkészlete ismétlődik) a cosinus függvény?
- -
90°
180°
540°
360°
D
Milyen intervallumban mozog a cosinus függvény értékkészlete?
- -
-1 és +1.
0 és +1.
Mínusz végtelen és +1.
Mínusz végtelen és plusz végtelen.
A
Mennyi 120° cosinusa?
- -
Nulla.
+0,5
-0,5
+1
C
Mennyi 270° cosinusa?
- -
Nulla.
+0,5
+1
-0,5
A
A következők kivétel nélkül mind a cosinus függvény zérushelyei:
- -
0°, 90°, 180°, 270°.
90°, 270°, -90°, 450°.
0°, 180°, 360°, -180°.
90°, 120°, 360°, 540°.
B
Derékszögű háromszög esetében, mi adja valamelyik hegyesszög tangensét?
- -
A szöggel szemközti befogó és a szög melletti befogó hányadosa adja a szög tangensét.
A szöggel szemközti befogó és az átfogó hányadosa adja a szög tangensét.
A szög melletti befogó és az átfogó hányadosa adja a szög tangensét.
A szög melletti befogó és a szöggel szemközti befogó hányadosa adja a szög tangensét.
A
Egység sugarú körre kiterjesztve mint értünk egy szög tangensén?
- -
Az aktuális szöghöz tartozó egység körívet.
A sugár meghosszabbításának merőleges vetületét a vízszintes érintőre.
A sugár meghosszabbításának merőleges vetületét a függőleges érintőre.
Az egységnyi sugár függőleges tengelyre eső merőleges vetületét.
C

Hány fokonként periodikus (az értékkészlete ismétlődik) a tangens függvény?
- -
90°
180°
270°
360°
B
Milyen intervallumban mozog a tangens függvény értékkészlete?
- -
-1 és +1.
Mínusz végtelen és plusz végtelen.
Mínusz végtelen és +1.
-1 és 0.
B
Mennyi 180° tangense?
- -
Nulla.
-0,5
+0,5
+1
A
Mennyi 45° tangense?
- -
Nulla.
+0,5
+1
-1
C
A következők kivétel nélkül mind a tangens függvény zérushelyei:
- -
0°, 90°, 180°, 270°.
-90°, 270°, 360°, 450°.
0°, 180°, 360°, 540°.
90°, 120°, 360°, 540°.
C
Melyik értéknél tart a tangnens függvény értéke mínusz vagy plusz végtelen felé?
- -
180°
45°
360°
90°
D
Hány aszimptotája van a tangens függvénynek 0° és 360° között?
- -
Egy.
Négy.
Három.
Kettő.
D
Mi a háromszög?
- -
Olyan zárt síkidom, aminek három csúcsa van, és szögeinek összege 360°.
Olyan zárt idom, aminek három oldala van, és szögeinek összege 180°.
Olyan síkidom, aminek három csúcsa van, és szögeinek összege 180°.
Olyan zárt síkidom, aminek három csúcsa van, és három szakasz határolja.
D
Hogyan számoljuk ki a háromszög kerületét, ha oldalai 'a', 'b', 'c' és 'a' oldali magassága 'm'?
- -
(a+b)*m;
a+(c+b);
(a*m)/2;
a+b+2*c;
B

Hogyan számoljuk ki a háromszög területét, ha oldalai 'a', 'b', 'c' és 'a' oldali magassága 'm'?
- -
a+b+c;
a+(c/2);
(a*m)/2;
b+c+m;
C
Mennyi a háromszög szögeinek összege?
- -
180°
450°
270°
360°
A
Mit nevezünk hegyesszögű háromszögnek?
- -
Az olyan háromszöget, aminek minden szöge kisebb 100°.
Az olyan háromszöget, aminek minden szöge 0° és 60° közé esik.
Az olyan háromszöget, aminek minden szöge kisebb 90°.
Az olyan háromszöget, aminek minden szöge 0° és 45° közé esik.
C
Mit nevezünk tompaszögű háromszögnek?
- -
Az olyan háromszöget, aminek egyik szöge nagyobb mint 90°.
Az olyan háromszöget, aminek két szöge nagyobb mint 90°.
Az olyan háromszöget, aminek két szöge 100° és 180° közé esik.
Az olyan háromszöget, aminek egyik szöge nagyobb mint 100°.
A
Mit nevezünk derékszögű háromszögnek?
- -
Az olyan háromszöget, aminek két szöge egyenlő 50°-al.
Az olyan háromszöget, aminek két szöge egyenlő 90°-al.
Az olyan háromszöget, aminek egyik szöge 45°.
Az olyan háromszöget, aminek egyik szöge 90°.
D
Hogyan számoljuk ki a derékszögű háromszög területét, ha befogói 'a' és 'b', átfogója pedig 'c'?
- -
a+b+c;
a+(b/2);
(a*b)/2;
a+b+c/2;
C
Derékszögű-e egy háromszög ha két szögének összege 90°.
- -
Nem, mert így a harmadik szög 80°.
Nem.
Igen, mert így két oldala egyenlő.
Igen, mert így a harmadik szög 90°.
D
Mit nevezünk egyenlőszárú háromszögnek?
- -
Az olyan háromszöget, aminek két szög összege 90°.
Az olyan háromszöget, aminek minden szöge egyenlő.
Az olyan háromszöget, aminek legalább két szöge 60°-os.
Az olyan háromszöget, aminek két szöge egyenlő.
D
Egyenlőszárú-e egy háromszög ha két oldala közül ez egyik egyenlő a harmadikkal?
- -
Csak akkor ha két szög is egyenlő.
Igen.
Csak akkor ha az egyik szöge 60°-os.
Nem.
B

Hogyan számoljuk ki az egyenlőszárú háromszög területét, ha az egyik oldal 'a' a két egyenlő oldal 'b', és az 'a'-hoz tartozó magasság 'm'? - - - - - - - -
(a+b)/m;
a+(m/2);
(a*m)/2;
a+b+m/2;
C
Hogyan számoljuk ki az egyenlőszárú háromszög kerületét, ha az egyik oldal 'a' a két egyenlő oldal 'b', és az 'a'-hoz tartozómagasság 'm'? - - - - - - - -
a+2*b;
a+b/2;
a+b+m/2;
(a*m)/2;
A
Mit nevezünk egyenlő oldalú háromszögnek?
- -
Az olyan háromszöget, aminek két szög összege 90°.
Az olyan háromszöget, aminek minden szöge 90°-os.
Az olyan háromszöget, aminek két szöge 60°-os.
Az olyan háromszöget, aminek minden szöge kisebb, mint 90°.
C
Hogyan számoljuk ki az egyenlő oldalú háromszög területét, ha az oldala 'a' a magassága pedig 'm'?
- -
a/m;
a+(m/2);
(a*m)/2;
(a+m)/2;
C
Hogyan számoljuk ki az egyenlő oldalú háromszög kerületét, ha az oldala 'a' a magassága pedig 'm'?
- -
2*a+a;
(a*m)/2;
a+(m/2);
3*a+m;
A
Egyenlő oldalú-e egy háromszög ha két szöge 60°-os.
- -
Igen, mert így a harmadik szög 60°.
Nem.
Igen, mert így két oldala biztos egyenlő.
Nem, mert így a harmadik szög 90°.
A
Mi a négyszög?
- -
Olyan zárt síkidom, aminek négy csúcsa van, és szögeinek összege 180°.
Olyan zárt idom, aminek négy oldala van, és szögeinek összege 360°.
Olyan síkidom, aminek négy szöge van, és szögeinek összege 180°.
Olyan zárt síkidom, aminek négy csúcsa van, és négy szakasz határolja.
D
Hogyan számoljuk ki a négyszög kerületét, ha oldalai 'a', 'b', 'c' és 'd'?
- -
(a+b)*(c+d);
a/d+(c+b);
(a*b)+(c*d);
a+b+(d+c);
D
Mennyi a négyszög szögeinek összege?
- -
180°
450°
360°
270°
C

Mit nevezünk konvex négyszögnek?
- -
Az olyan négyszöget, aminek minden szöge kisebb 90°.
Az olyan négyszöget, aminek minden szöge 90° és 180° közé esik.
Az olyan négyszöget, aminek minden szöge kisebb 180°.
Az olyan négyszöget, aminek minden szöge 0° és 270° közé esik.
C
Mit nevezünk konkáv négyszögnek?
- -
Az olyan négyszöget, aminek két szögének összege 180°.
Az olyan négyszöget, aminek egyik szöge nagyobb mint 180°.
Az olyan négyszöget, aminek két szöge 90° és 180° közé esik.
Az olyan négyszöget, aminek egyik szöge nagyobb mint 90°.
B
Mit nevezünk négyzetnek?
- -
Az olyan négyszöget, aminek két-két szögének összege 180°.
Az olyan négyszöget, aminek minden oldala egyenlő.
Az olyan négyszöget, aminek szögeinek összege 360°.
Az olyan négyszöget, aminek minden szöge 90° és oldalai egyenlőek.
D
Hogyan számoljuk ki a négyzet területét, ha oldalának hossza 'a'?
- -
4*a;
a*a;
a+a+a+a;
(a+a)*a;
B
Hogyan számoljuk ki a négyzet kerületét, ha oldalának hossza 'a'?
- -
(a+a)*a;
a*a;
4*(a+a);
a+a+a+a;
D
Négyzet-e egy négyszög ha oldalai egyenlőek?
- -
Nem, mert ettől még a szögei különbözhetnek.
Nem, mert ettől a szögek összege még nem biztos, hogy 360°.
Igen, mert ez már rombusz is.
Igen, mert így minden szöge 90°.
A
Mit nevezünk téglalapnak?
- -
Az olyan négyszöget, aminek szemközti szögei egyenlőek.
Az olyan négyszöget, aminek minden szöge 90°.
Az olyan négyszöget, aminek legalább két szöge 90°-os.
Az olyan négyszöget, aminek szomszédos két-két szöge egyenlő.
B
Hogyan számoljuk ki a téglalap területét, ha az egyik oldala 'a' a másik pedig 'b'?
- -
a*b;
a+b;
(a+b)/2;
(a*b)/2;
A
Hogyan számoljuk ki a téglalap kerületét, ha az egyik oldala 'a' a másik pedig 'b',
- -
a*b+a*b;
2*a+b+b;
(a+b)/2;
(a*b)+a+b;
B

Mit nevezünk paralelogrammának?
- -
Az olyan négyszöget, aminek egyik pár szemközti oldala párhuzamos.
Az olyan négyszöget, aminek minden szöge 90°-os.
Az olyan négyszöget, aminek szemközti oldalai egyenlőek.
Az olyan négyszöget, aminek két-két pár szomszédos oldalai egyenlőek.
C
Hogyan számoljuk ki a paralelogramma területét, ha oldalai 'a' illetve 'b' az 'a'-hoz tartozó magassága pedig 'm'? - - - - - - - - - - - - -
(a*m)/2;
(a+b)*m;
(a*m)/2;
a*m;
D
Hogyan számoljuk ki a paralelogramma kerületét, ha oldalai 'a' illetve 'b' az 'a'-hoz tartozó magassága pedig 'm'? - - - - - - - - - - - - -
2*a+2*m;
2*(a+b);
a+b+(m/2);
a*b+m;
B
Paralelogramma-e egy négyszög ha két-két szemközti oldala párhuzamos?
- -
Igen, mert így minden szöge egyenlő.
Nem.
Igen.
Nem, mert a szemközti oldalak nem egyenlőek.
C
Mit nevezünk rombusznak?
- -
Az olyan négyszöget, aminek egyik pár szemközti oldala párhuzamos.
Az olyan négyszöget, aminek minden oldala egyenlő.
Az olyan négyszöget, aminek szemközti oldalai egyenlőek.
Az olyan négyszöget, aminek két-két pár szomszédos oldalai egyenlőek.
B
Hogyan számoljuk ki a rombusz területét, ha oldala 'a' és átlói 'm' illetve 'n'?
- -
(n+m)/2;
a+m*n;
(n*m)/a;
(n*m)/2;
D
Hogyan számoljuk ki a rombusz kerületét, ha oldala 'a' és átlói 'm' illetve 'n'?
- -
(n*m)/2;
a*a;
a+(m*n);
2*a+2*a;
D
Rombusz-e egy négyszög ha két-két szemközti szöge egyenlő?
- -
Nem, mert ettől még nem egyenlő minden oldala.
Nem, mert ettől még nem 90°-osak a szögei.
Igen.
Igen, mert így minden oldala egyenlő.
A
Mit nevezünk deltoidnak?
- -
Az olyan négyszöget, aminek egyik pár szemközti oldala párhuzamos.
Az olyan négyszöget, aminek minden oldala egyenlő.
Az olyan négyszöget, aminek szemközti oldalai egyenlőek.
Az olyan négyszöget, aminek két-két pár szomszédos oldala egyenlő.
D

Hogyan számoljuk ki a deltoid területét, ha oldalai 'a' és 'b'és átlói 'm' illetve 'n'?
- -
(a+b)/2;
(a*b)/2;
(n*m)/2;
(n*m)/(a+b);
C
Hogyan számoljuk ki a deltoid kerületét, ha oldalai 'a' és 'b'és átlói 'm' illetve 'n'?
- -
(n*m)/2;
2*(a+b);
(a*b)+(m*n);
2*n+2*m;
B
Mit nevezünk konkáv deltoidnak?
- -
A deltoid tulajdonságok mellett, van két szöge, ami nagyobb mint 90°.
A deltoid tulajdonságok mellett, van egy szöge, ami nagyobb mint 180°.
A deltoid tulajdonságok mellett, van egy szöge, ami 180°.
A deltoid tulajdonságok mellett, van egy szöge, ami nagyobb mint 90°.
B
Mit nevezünk trapéznak?
- -
Az olyan négyszöget, aminek egyik pár szemközti oldala párhuzamos.
Az olyan négyszöget, aminek két-két pár szemközti oldala párhuzamos.
Az olyan négyszöget, aminek szemközti oldalai egyenlőek.
Az olyan négyszöget, aminek van két 90°-os szöge.
A
Hogyan számoljuk ki a trapéz területét, ha oldalai 'a', 'b', 'c' és 'd', ahol 'a' és 'c' párhuzamos,
valamint az 'a'-hoz tartozó magasság 'm'? - - -
((a+c)*m)/2;
((a*b)*m)/2;
(m*m)/(a+c);
m/(a+c);
A
Hogyan számoljuk ki a trapéz kerületét, ha oldalai 'a', 'b', 'c' és 'd', ahol 'a' és 'c' párhuzamos,
valamint az 'a'-hoz tartozó magasság 'm'? - - -
a*b+c+d;
a+c+m;
2*a+2*b+m;
d+c+b+a;
D
Mit nevezünk derékszögű trapéznak?
- -
Az olyan négyszöget, aminek egyik pár szemközti oldala párhuzamos.
Az olyan négyszöget, aminek két-két pár szemközti oldala párhuzamos.
Az olyan négyszöget, aminek egyik pár két szomszédos szöge 90°.
Az olyan négyszöget, aminek van két 90°-os szöge.
C
Mit mond ki a Pitagorasz-tétel?
- -
Minden derékszögű háromszögre igaz, hogy a nem derékszögek összege 90°.
Minden derékszögű háromszögre igaz, hogy a befogók négyzeteinek összege egyenlő az átfogó négyzetével.
Minden derékszögű háromszögre igaz, hogy a befogók négyzeteinek fele egyenlő az átfogó négyzetével.
Minden derékszögű háromszögre igaz, hogy a befogók négyzetgyökei mindig kisebbek mint az átfogó.
B
A Pitagorasz-tétel képlettel, ha a derékszögű háromszög befogói 'a' és 'b', az átfogó pedig 'c':
- -
a*b-b*a=c*c;
a*c+b*c=a*b;
b*b+c*c=a*a;
b*b+a*a=c*c;
D

A tanult Pitagorasz-tétel bizonyításnál hány darab azonos területű háromszög található az azonos területű 2 darab négyzetben? - - - - - - -
2*4;
2*2;
2*6;
2*5;
A
Mit mond ki a Thalész-tétel?
- -
Ha egy kör átmérőjének két végpontját összekötjük a körív egy pontjával, a köríven derékszöget kapunk.
Ha egy kör átmérőjének felére derékszöget állítunk akkor az a kapott egyenes felezi a körívet.
Ha egy kör átmérőjének két végpontját összekötjük a kör bármely pontjával, akkor derékszöget kapunk.
Ha egy kör átmérőjének két végpontját összekötjük egy külső ponttal, a kapott szög kisebb mint 90°.
A
A tanult Thalész-tétel bizonyításnál melyik törvényt használtuk fel?
- -
A kör legnagyobb szelője az átmérője.
Az egyenlő oldalú háromszög szögei 60°-osak.
A háromszög belső szögeinek összege 180°.
A derékszögű háromszög középpontosan tükrözve téglalapot ad.
C