

Debreceni Egyetem
Informatikai Kar

A CIKLUSSZERVEZŐ UTASÍTÁSOK TANÍTÁSA

Témavezető:

Dr. Papp Zoltán
egyetemi adjunktus

Készítette:

Krisztián-Molnár Anikó
pedagógus szakvizsgára felkészítő
informatika szakirányú
továbbképzés hallgatója

Debrecen

2010

TARTALOMJEGYZÉK

Bevezetés	3
A Pascal nyelv rövid története.....	5
Magasszintű nyelvek osztályozása	6
Imperatív nyelvek	6
Deklaratív nyelvek.....	6
Más elvű nyelvek.....	7
Utasítások	7
Ciklusszervező utasítások.....	8
Tanmenet	13
Célok és feladatok	13
Követelmény.....	14
Óraterv	15
Ciklusszervező utasítások tanítása.....	16
Óravázlat 1.....	17
Óravázlat 2.....	27
Óravázlat 3.....	37
Óravázlat 4.....	48
Óravázlat 5.....	56
Óravázlat 6.....	67
Óravázlat 7.....	76
Óravázlat 8.....	87
Összegzés	98
Irodalomjegyzék	99

Bevezetés

Az elmúlt néhány évben vált meghatározóvá az objektumorientált paradigma és hódított elképzelhetetlen mértékben a Java – és sok egyéb programozási nyelv. Talán emiatt is, egyes strukturált programozási módszerek alkalmazása háttérbe szorult. Véleményem szerint azonban az elemi programozási algoritmusok megismerése továbbra is szükséges. Ennek két alapvető oka:

- Ezen algoritmusok ismerete segíti az algoritmikus gondolkodás fejlődését, ismeretükkel, használatukkal hatékonyabb objektumorientált programok készíthetők.
- Az objektumok létrehozása, működésük leírása továbbra is algoritmusokkal történik.

A számítástechnika jelenleg a legdinamikusabban fejlődő tudományág. Szinte nyomon követhetetlenek mindazon újdonságok, amelyek naponta jelennek meg a hardver és szoftver piacon. Jelentős változáson ment át a számítógépes programok készítésének elmélete is. A grafikus operációs rendszerek széleskörű elterjedésével a programozás hangsúlya jelentősen áttevődött a programok esztétikus és szabványos megjelenésére, és az új elemek megjelenésével az eseményorientált programozásra. Új kihívás elé állította a programozókat a multimédia, valamint az Internet megjelenése is, hiszen ezek is másfajta programozási igényekkel jelentkeztek, mint a korábbi programozási elvárások. A korábbi kódközpontú, soros algoritmus szemléletű programozást felváltotta a magas szintű fejlesztőeszközök használata, melyek némelyike még a konkrét kódkészítést is kiküszöbölte a programozók munkájából.

A ma emberétől természetes elvárás az, hogy legalább alapfokon kezelni tudja a számítógépet. A gépek üzemeltetése azonban sok olyan problémát vet fel, melyet kezelői ismeretekkel nem lehet megoldani.

A számítógépes programkészítés komoly, kreatív gondolkodásmódot igényel, melyet csak kellő elméleti megalapozással lehet elérni. Így általános algoritmikus és problémamegoldó gondolkodásmód kialakítására kell törekednünk. Ez igényli a klasszikus témakörök ismertetését még akkor is, ha ezek ma már közvetlenül nem is jelennek meg a programozó munka során.

„Napjainkra a számítógép (és különösen a világhálózat) univerzális szellemi munkaeszközzé vált. Az írás kialakulása óta a számítógép a legnagyobb

találmány. Közel az idő, amikor az informatikai tudás éppen olyan fontos lesz, mint az írni-olvasni tudás. Az információs társadalom küszöbén állunk. Az iskola feladata, hogy felkészítsen az információs társadalomban való életre.

A kormányprogram, valamint az ország fejlesztésének koncepciója kiemelten kezeli az Európai Unióhoz való csatlakozásunkat. Ennek feltétele, hogy közoktatási rendszerünk is korszerű legyen, és megfeleljen az információs társadalom igényeinek.” /Végh András/

Végh András találóan fogalmazta meg az informatika oktatásának fontosságát. Én is e tevékenységben igyekszem aktívan részt venni. Informatika szakos tanárként dolgozom immár 9 éve, s programozást 8 éve tanítok. Iskolám informatika szakos hallgatókat is nevel, s míg az informatikát minden osztályban legalább 2 éven keresztül oktatjuk, egyedül ők részesülnek programozás tanításában, elsajátításának lehetőségében. 11. és 12. évfolyamon, heti 4 órában tanulnak programozást a szakos diákok. Minden csoportomnál azt tapasztaltam, hogy a programozás alapjainak elsajátítása, a Pascal program kezelésének megismerése s az egyszerű programok megírása után felcsillan a diákok szeme a kissé összetettebb ciklusszervező utasítások alkalmazásakor. Ezek azok a lépések munkájukban, melyek már igazán gondolkodásra készítetik őket, s kihívást jelent számukra e feladatok megoldása. Ezen tapasztalat ösztönzött arra, hogy a ciklusok tanítását válasszam szakdolgozatom témájául.

A Pascal nyelv rövid története

A Pascal magas szintű, általános célú, strukturált programozási nyelv, amelyet Niklaus WIRTH alkotott meg Zürichben az 1960-as és 70-es évek fordulóján. A nyelvet Blaise Pascalról, a nagy 17. századi francia matematikusról és filozófusról nevezte el. Az Algol és más nyelvek logikájából kiindulva, azokat - főleg az adatszerzés terén - új elemekkel bővítve, remek programnyelvet alkotott.

Szigorú nyelv, egyszerű eszközrendszerrel, szintaktikai és szemantikai szabályokkal. Az oktatás kedvelt nyelve. A Pascal szabad formátumú nyelv, azaz a külalaknak (pl. sorhosszúság, bekezdések) csak a program áttekinthetősége szempontjából van jelentősége. A nyelv nem különbözteti meg a kis és nagybetűket.

A Standard Pascal meglehetősen puritán nyelv volt, csak a legszükségesebb parancsokat és utasításokat tartalmazta, a programozónak meglehetősen sok munkája volt. A programnyelv azonban valóban jól használható, közel áll az emberi gondolkodáshoz és könnyen elsajátítható. Ezért aztán nem kell csodálkozni, hogy rohamtempóban terjedt el a világban - főleg a személyi számítógépek megjelenése óta. Szoftvercégek sora adta ki saját változatát, kiegészítve újabb parancsokkal, függvényekkel, miközben a kezelését egyre kényelmesebbé tették, integrált keretrendszert alkottak hozzá stb. Az egymással konkuráló szoftvercégek Pascal változatai közül kétségtelenül a Borland cégé lett a legsikeresebb. Az évek során a Turbo Pascal verziók között ma a 7.0-nál tartanak, de már évekkel ezelőtt megjelent a Windows-os változat is. Sőt megjelent a Delphi nevű, szintén Pascal-kódú Windows-os fejlesztőrendszer 16 és 32 bites változata is.

Jellemzői:

- Amatőr, Neumann-elvű
- Program = 1 fordítási egység = több programegység, blokkstruktúra
- Statikus deklaráció kiértékelés
- Dinamikus memóriakezelés (statikus és véges élettartamú változók), van kézi lefoglalás, törlés is, sokféle élettartam
- Hatáskör, láthatóság: statikus hatáskör kijelölés
- Paraméterátadás: cím szerinti, érték szerinti (eljárásparaméter)
- Típus-kompatibilitás név szerint

- Azonosítók nem lapolhatók át
- Nem erősen típusos (1973 szerint), erősen típusos (1983 szerint)
- Alapszavak védettek
- Nincs sor = utasítás megfeleltetés
- Utasítás elválasztó jel a pontosvessző
- Mindent előre kell definiálni (alulról felfelé építkezés)

Az elkövetkezendőkben feltárul előttünk, hogy a programnyelvek körében hol is helyezhető el a Pascal nyelv.

Magasszintű nyelvek osztályozása

Imperatív nyelvek

Utasításszerkezetűek, tehát alapeszközei az utasítások és a változók. A program szövege utasítássorozat, minden utasítás mögött gépi kód áll. Minden program utasítássorozat, amely mögött több gépi utasítás áll. Kötődnek a Neumann-architektúrához, általában fordítóprogramosak. Algoritmikus nyelvek, a programban azt az algoritmust írják le, amelyet a gép végrehajt, és a probléma megoldása így születik meg. A program a hatását a tár egyes területein lévő értékeken fejti ki.

Eljárásorientált: A program szövege külön eszközökkel részekre tagolható.

pl. Fortran, Cobol, Algol60, PL/1, **P A S C A L**, C, Ada

Objektumorientált: Modernebb nyelvek, az absztrakció megjelenése jellemzi.

pl. Eiffel, Smalltalk, C++, Java, Simula67)

Deklaratív nyelvek

Nincs utasításfogalom, a Neumann-architektúrától távol áll. Nem algoritmikusak, a programban csak a problémát fogalmazom meg, a megoldást nem, az algoritmus a rendszerbe van beépítve. A tárhoz a programozónak kevés köze van, nem feladata a tár egyes részeinek módosítása.

Funkcionális (applikatív) nyelvek: pl. Lisp

Logikai nyelvek: Matematikai logikai alapokra épül. pl. Prolog

Más elvű nyelvek

Az előző nyelvek elveit tagadja, ide olyan nyelveket sorolunk, amiket máshová nem tudunk besorolni, pl. Apl.

Mivel a Pascal programra jellemző, hogy csak elemi tevékenységekből és a három vezérlő elem egymásutánjából illetve egymásba ágyazásából áll, megjegyezhetjük, hogy a strukturált programozás jeles képviselője.

A három vezérlő utasítás:

Szekvencia: egymás után végrehajtandó tevékenységek sorozata.

Szelekció: választás megadott tevékenységek közül.

Iteráció: megadott tevékenységek ismételt, többszöri végrehajtása.

A szakdolgozatom témájául választott iterációk (ciklusok), az utasítások egyik válfajai:

Utasítások

Az utasítások a program szövegének azon egységei, amelyeket a fordítóprogramnak elsősorban fel kell ismernie, mert a fordítóprogram ezekkel az utasításokkal dolgozik.

Típusai:

Deklarációs utasítások: A fordítóprogramnak szólnak, a működését befolyásolják, szolgáltatást kérnek, üzemmódot váltanak, információval látják el, amelyet a fordítóprogram felhasznál kód generálásánál. Nem áll mögöttük kód, a fordítóprogram nem fordítja le őket.

Végrehajtható utasítások: Kód áll mögöttük, a fordítóprogram lefordítja, és ezekből generálja a tárgyprogramot.

- **értékadó utasítások:** Szerepük, hogy egy változó értékkomponensét a program futásának bármely pillanatában be tudjuk állítani, változtatni.
- **üres utasítások:** A legtöbb magasszintű nyelvben van, néhány nyelvben elengedhetetlen, bizonyos szituációkban a nyelvek előírják. Külön gépi kódja van, jele vagy van (pl. ;) vagy nincs. Hatására a program nem csinál semmit.
- **elágaztató utasítások**
- **ciklusszervező utasítások**
- **ugró utasítások**
- **hívó utasítások**

Ezek minden eljárásorientált nyelvben megtalálhatóak. Az elágaztató, ciklusszervező, ugró és hívó utasításokat vezérlő utasításoknak hívjuk, a program vezérlési szerkezetének felírására szolgálnak

- **input-output utasítások:** Az adatmozgatást vezérlik a perifériák és a tár között valamelyik irányban.
- **egyéb utasítások:** A nyelvek között a legnagyobb eltérés az egyéb utasításoknál van.
pl. Pascal: csak egy van: WITH (a minősítést segíti (record)),
PL/1: több egyéb utasítás van, mint más utasítás.

Ciklusszervező utasítások

A ciklusok **utasítás-sorozatok** ismételt végrehajtására szolgálnak. Ezek az utasítások alkotják a **ciklusmagot**. A végrehajtásukat a **ciklusfeltétel** szabályozza, melyet a *logikai kifejezés* vagy *ciklusváltozó* biztosít. Az ismétlés eldöntésének helyétől függően megkülönböztetünk elől tesztelő és hátul tesztelő ciklust.

WHILE utasítás (elől tesztelő ciklus)

WHILE logikai kifejezés DO utasítás

Másképpen:

while=míg, amíg

Ciklus, *amíg* feltétel

a *ciklusmag* utasításai

Ciklus vége

while logikai kifejezés **do** ciklusmag

ha a ciklusmag több utasításból áll, akkor :

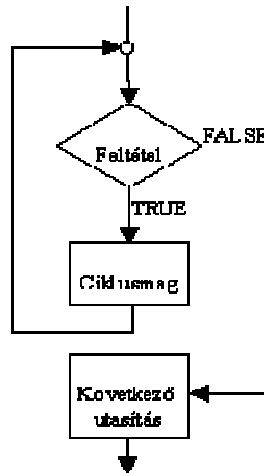
while logikai kifejezés **do**

begin

ciklusmag

end

Az utasítás folyamatábrája:



A do kulcsszó utáni utasítás, az ún. ciklusmag végrehajtása mindig addig ismétlődik, míg a while kulcsszó után álló logikai kifejezés értéke TRUE. A kifejezés kiértékelésére a ciklusmag végrehajtása előtt kerül sor, tehát ha az már induláskor hamis (false), akkor a ciklusmag egyszer sem hajtódik végre.

FOR utasítás (számlálós ciklus, adott lépésszámú ciklus, elől tesztelő léptető ciklus)

FOR változó = kezdőérték {TO | DOWNTO} végérték

DO utasítás

Másképpen:

for = -tól, -tól

to = -ig

Ciklus *ciklusváltozó* := kezdőértéktől végértékig

a *ciklusmag* utasításai

Ciklus vége

for ciklusváltozó := alsóhatár **to** felsőhatár **do** ciklusmag

ha a ciklusmag több utasításból áll, akkor :

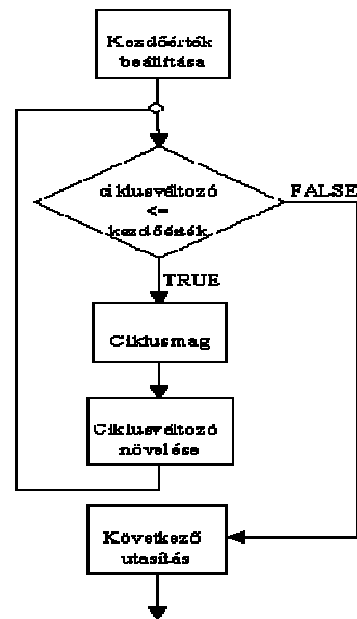
for ciklusváltozó := alsóhatár **to** felsőhatár **do**

begin

ciklusmag

end

Az utasítás folyamatábrája:



A ciklusfejhez érve a ciklusváltozó felveszi a kezdőértéket. Majd végrehajtódnak a ciklusmag utasításai. A ciklus végéhez érve a ciklusváltozó a következő értéket veszi fel. Ha ez az érték nagyobb a végértéknél, akkor a végrehajtás a ciklus utáni első programsoron folytatódik, ha pedig még nem, akkor a ciklusmag hajtódik végre ismét, s ez folytatódik mindaddig, míg az érték nagyobb nem lesz. Tehát a **do** kulcsszó utáni ciklusmag annyiszor kerül végrehajtásra, amíg a ciklusváltozó értéke az alsóhatártól egyesével növelve a felsőhatárt el nem éri. Természetesen a ciklusváltozó csak sorszámozott típusú lehet, mivel nem lehet törtrészben megismételni egy tevékenységsorozatot.

Ha a ciklusmagban több utasítás szerepel, akkor a **begin..end** kulcsszavakkal egybe kell fogni a tevékenységsorozatot.

Ritkábban használatos csökkenő ciklusváltozóval ciklust szervezni, de ha mégis ilyet akarunk, akkor a *to* kulcsszó helyett a *downto* – t használjuk.

(**for** ciklusváltozó := felsőhatár **downto** alsóhatár **do** ciklusmag)

Repeat ..UNTIL utasítás (hátraltesztelő ciklus)

REPEAT utasítások UNTIL logikai kifejezés

Másképpen:

repeat = ismétlés

until = addig – amíg

Ciklus

a *ciklusmag* utasításai

amíg feltétel

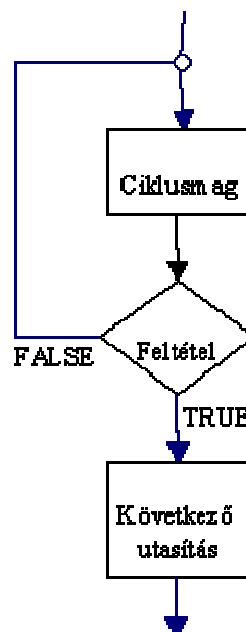
Ciklus vége

repeat

utasítás (ciklusmag)

until logikai kifejezés

Az utasítás folyamatábrája:



A ciklusmag végrehajtása mindaddig ismétlődik, míg a logikai kifejezés értéke TRUE nem lesz. Mivel a kifejezés kiértékelése a ciklus végén történik, a ciklusmag legalább egyszer végrehajtódik. Ha a ciklusmagban több utasítás van, akkor sincs szükség **begin...end** kulcsszavakra, mivel a **repeat ..until** parancspár is egybefogja a tevékenységsorozatot.

Fontos megemlíteni, hogy a **while-do** és a **repeat-until** utasításpároknál a ciklusmag olyan legyen, hogy hordozza magában a logikai kifejezés teljesülésének lehetőségét, mert ha nem teljesül, akkor végtelen ciklust hozhatunk létre.

Tanmenet

a 11. B informatika szakmacsoport

Célok és feladatok

A tantárgy oktatásának alapfeladata a tanulók **digitális kompetenciájának** fejlesztése, de informatikaoktatás, ezen belül a programozás célrendszere összhangban van a **kulcskompetenciákkal**, mely magába foglalja:

- A **társadalom információs technológiáinak magabiztos és kritikus használatát** a munka, a kommunikáció és a szabadidő terén. Ez a következő készségeken, tevékenységeken alapul: az információ felismerése, keresése, értékelése, tárolása, előállítása, feldolgozása, bemutatása és cseréje.
- Az informatikában fontos szerepet kap a **matematikai kompetencia fejlesztése**, azaz a matematikai gondolkodás és problémamegoldás fejlesztése, az algoritmusok alkalmazásának képessége, mindennapi problémák megoldása és a világ rendszereinek, folyamatainak informatikai (számítástechnikai) modellezése során.
- A **természettudományos kompetencia** készséget és képességet jelent arra, hogy ismeretek és módszerek sokaságának felhasználásával magyarázzuk, és előre jelezzük a természeti folyamatokat, illetve a rendszerek mozgását, tulajdonságait. A természettudományos és **műszaki-technikai kompetencia** magában foglalja a kételkedő, kritikus és kíváncsi attitűdöt, a megismerés és konstruálás iránti vágyat, valamint a biztonsággal és fenntarthatósággal kapcsolatos etikai kérdések iránti érdeklődést is.
- Az informatikaoktatás elősegíti a **kezdeményezőképeség és a vállalkozói kompetencia** tudatos fejlesztését is. A gyakorlati feladatok megoldása során olyan készségeket és képességeket fejleszt, mint a tervezés, szervezés, irányítás, elemzés, kommunikálás, a tapasztalatok értékelése, egyénileg és csapatban történő munkavégzés. A programozás oktatása tág lehetőségeket teremt az innovatív és kreatív problémamegoldásra.
- Az információs termékek alkotása, tervezése és gyártása során fontos szempont a mű, illetve a dokumentum esztétikai megjelenése és kifejezőképessége. Az informatikaoktatás az **esztétikai-művészeti tudatosság és kifejezőképesség kompetenciafejlesztésének** egyik korszerű területe.

A hétköznapi életből vett, változatos feladatok szövegezése lehetőséget biztosít arra, hogy tanulóinkat az **egészséges életmódra, a környezettudatos és kritikus fogyasztói magatartásra** neveljük.

A programozás oktatásának szakmai célja, hogy a tanulók megismerjék az alapvető programozási elveket és tételeket, a programozási elmélet alapjait, a strukturált programozás fogásait és az egyszerűbb programtervezési módszereket, a korszerű objektumorientált programozás alapjait. Az elmülethez kapcsolódó gyakorlat keretében önálló feladatokon keresztül mélyítsék el tudásukat. Ismerjék meg a korszerű objektumorientált és vizuális programozási elveket és azok egyszerűbb alkalmazását.

Követelmény

- A tananyag tanulása során fejlődjenek a tanulók szakmai tantárgyak tanulásához szükséges képességei, különösen az ismeretelemző-értékelő gondolkodás, az algoritmikus gondolkodás, a kreativitás, az önálló megfigyelés képessége, az önálló ismeretszerzés és az ismeretek alkalmazásának képessége.
- A tevékenység során folyamatosan erősödjenek a tanulási és szakmai motiváció. Alakuljon ki a reális önismeret, váljon tudatos döntéssé a pályaválasztás. Ismerjék meg a szakmai tevékenységek végzéséhez szükséges magatartási szabályokat, magatartásformákat.
- A szakmai tanulmányi tevékenység által alakuljon ki, fejlődjön, erősödjön a szakmacsoporthoz, az informatikai szakmákhoz való kötődés.
- Alakuljon ki a felelősségérzetük egymás iránt, a társakkal való együttműködési képesség, a közösség és az egyén érdekei összeegyeztetésének képessége.
- Szerezzenek gyakorlatot a számítógépes programok készítésében, valamelyik korszerű, általános programozási nyelv használatában. Ismerjék fel a tanult nyelven készített programok forráskódját, tudják a forráskódot olvasni, értelmezni, és tudjanak hasonlókat készíteni.

Óraterv

Témák	Óraszám
Programozás-technikai alapismeretek	20
Programozási alapok	20
<i>Elemi algoritmusok</i>	24
A programozás eszközei	70
Szabad órakeret	10
Összesen	144

Itt csupán a 3. témakört emelném ki:

Elemi algoritmusok

Cél:

Konkrét feladatokon keresztül megismertetni a tanulókkal a sorozathoz értéket rendelő „programozási tételeket”, felismertetni ezek alkalmazási lehetőségeit, valamint azt, hogy a hétköznapi életben is ezeket az algoritmusokat alkalmazzuk a felmerülő feladatok megoldásában.

Tartalom:

1. A tömb adatszerkezet:
 - szükségessége,
 - definiálása,
 - használata.
2. A ciklusok (elől tesztelős, hátul tesztelős, számlálós ciklus) alkalmazása.
3. Elemi algoritmusok megismerése:
 - összegzés,
 - eldöntés,
 - kiválasztás, (lineáris) keresés,
 - megszámlálás,
 - maximum-kiválasztás.

A továbbhaladás feltételei:

Tudja alkalmazni az elemi programozási tételeket feladatmegoldás során.

Ciklusszervező utasítások tanítása

Iskolánkban programozást az informatika szakmacsoportos tanulóknak – ahogyan azt a tanmenet is mutatja - heti 4 órában tanítunk. Mind 11., mind 12. évfolyamon. Az iterációk típusaival, azok működésével, alkalmazásával természetesen már 11.-ben megismerkednek a diákok, hiszen programozás-tanulásuk alapját képezik. Majd a 2 év során felhasználják tudásukat a különböző feladatok megoldásai során, melyek egyre összetettebbé válnak, amint újabb ismeretekre tesznek szert.

A tanulók órai munkájukat a számítógépük egy könyvtárába mentik folyamatosan, majd óra végén saját adathordozó eszközükre is elmentve el tudják vinni azokat, s így a tanultak elmélyítésére otthon is lehetőségük van. Rendelkezésünkre áll egy úgynevezett Netsupport School rendszer, mely megkönnyíti az informatika oktatását, ugyanis a hálózatban lévő gépekkel a tanári gépről lehetőség van kommunikálni. Kitehető a tanári gép, ill. bármely gép képernyője a többi gépre pl. magyarázat s; egy-egy diák egér-ill. billentyűhasználat is átvehető; egyénre szóló feladatlapokat küldhetünk az egyes tanulóknak, stb.. A programozás óráimon gyakran használom hibajavítás, magyarázat céljából, így az óravázlatokban is gyakran hivatkozom majd rá.

A programozás során alkalmazott tesztelés elvégzésére a Wink programot hívtam segítségül, s néhány bemutatóval gazdagítottam óraterveimet.

Óravázlat 1.

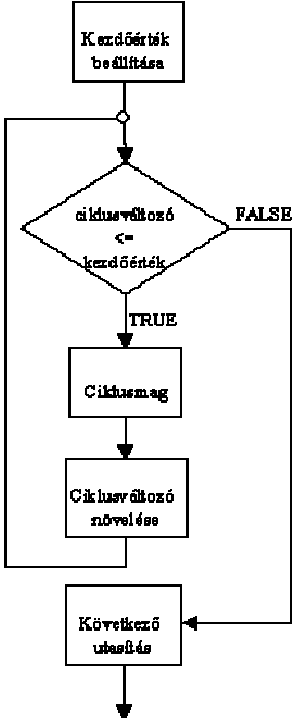
A tanítási óra anyaga:	Ciklusszervező utasítások (iterációk) típusai, Számlálós (for) ciklus
A tanítási óra típusa:	Új ismeret-szerző óra
Eszközök:	Vizuál tábla, számítógépek, Turbo Pascal program, Netsupport School rendszer, Molnár Csaba – Sági Gábor: Programozás Turbo Pascal nyelven
A tanulócsoporthoz előzetes ismeretei:	Az előző órákon jártasságot szereztek az elágazások (feltételes utasítások) alkalmazása terén, valamint a tömb összetett adattípussal ismerkedtek meg (vektor, mátrix)
A tanítási óra céljai:	Az új ismeretek beágyazása a meglévő ismeretrendszerbe Az új ismeretek összefüggéseinek feltárása, gyakorlati jelentőségének megértése Gondolkodási műveletek mozgósítása, a tanulási folyamatok segítése A programozás tantárgy iránti érdeklődés felkeltése, illetve megerősítése Alapvető programozási fogalmak ismerete

Folyamatterv:

- I. Jelentés, szervezési feladatok
- II. A ciklusok fogalma, típusai; számlálós ciklus fogalma, működési elve
- III. Feladatok ciklusokra
- IV. Értékelés

SZAKMAI GONDOLATMENET, MUNKAFÁZISOK	TANÁRI, TANULÓI TEVÉKENYSÉGEK, MUNKAFORMÁK, MÓDSZEREK	ÚJ FOGALMAK, ISMERETEK	ESZKÖZÖK ÉS FELTÉTELEK
I. Jelentés, szervezési feladatok	A hetesek jelentése A csoport üdvözlése A napló beírása A számítógépek bekapcsolása <i>/Fegyelemre, udvariasságra, a szabályok betartására nevelés./</i>		napló, számítógépek
II. Ciklusszervező utasítások 1. Iterációk típusai:	Felmerül az igény, hogy egy-egy művelet akár többször is végre lehessen hajtani, ill. több elem esetén is végrehajthasson. Ezen probléma révén jutunk el a ciklusutasítások, azaz iterációk alkalmazásához. Tehát a ciklusok segítségével megismételhető a program meghatározott része. Ide soroljuk a for (számlálós) , a while (elől tesztelő), és a repeat ... until (hátral tesztelő) utasításokat. <i>A mai órán a számlálós ciklussal ismerkedünk meg.</i> Vázlatot írok a táblára, a tanulók jegyzetelnek füzetükbe, diktálás során az új fogalmakat is lejegyzik.	ciklus(iteráció),for, while, repeat..until	vizuál tábla
2. Számlálós ciklus:	<u>FOR</u> utasítás (számlálós ciklus, adott lépszámú ciklus, elől tesztelő léptető ciklus) <u>Fogalma:</u> A feladatok egy részénél előre tudjuk, hogy a	for-„számlálós”	

SZAKMAI GONDOLATMENET, MUNKAFÁZISOK	TANÁRI, TANULÓI TEVÉKENYSÉGEK, MUNKAFORMÁK, MÓDSZEREK	ÚJ FOGALMAK, ISMERETEK	ESZKÖZÖK ÉS FELTÉTELEK
	<p><u>Működési elve:</u></p> <p>A ciklusfejhez érve a ciklusváltozó felveszi a kezdőértéket. Majd végrehajtnak a ciklusmag utasításai. A ciklus végéhez érve a ciklusváltozó a következő értéket veszi fel. Ha ez az érték nagyobb a végértéknél, akkor a végrehajtás a ciklus utáni első programsoron folytatódik, ha pedig még nem, akkor a ciklusmag hajtódik végre ismét, s ez folytatódik mindaddig, míg az érték nagyobb nem lesz.</p> <p>Tehát a do kulcsszó utáni ciklusmag annyiszor kerül végrehajtásra, amíg a ciklusváltozó értéke az alsóhatártól egyesével növelve a felsőhatárt el nem éri.</p> <p>Természetesen a ciklusváltozó csak sorszámozott típusú lehet, mivel nem lehet törtrészben megismételni egy tevékenységsorozatot.</p> <p>Ha a ciklusmagban több utasítás szerepel, akkor a begin..end kulcsszavakkal egybe kell fogni a tevékenységsorozatot.</p> <p>Ritkábban használatos csökkenő ciklusváltozóval ciklust szervezni, de ha mégis ilyet akarunk, akkor a <i>to</i> kulcsszó helyett a <i>downto</i> – t használjuk.</p>		<p>Szövegértés, összefüggések meglátása</p>

SZAKMAI GONDOLATMENET, MUNKAFÁZISOK	TANÁRI, TANULÓI TEVÉKENYSÉGEK, MUNKAFORMÁK, MÓDSZEREK	ÚJ FOGALMAK, ISMERETEK	ESZKÖZÖK ÉS FELTÉTELEK
<p>Az utasítás folyamatábrája is felkerül a táblára annak érdekében, hogy az iteráció részei közötti összefüggések, valamint szemantikájának megértését segítse. Így az algoritmus ábrázoló eszközeinek alkalmazása révén értelmezik azt.</p>	<p>(for ciklusváltozó := felsőhatár downto alsóhatár do ciklusmag)</p> <p>Folyamatábra:</p>  <pre> graph TD Start[Kezdőérték beállítása] --> Decision{ciklusváltozó <= kezdőérték} Decision -- TRUE --> Body[Ciklusmag] Body --> Increment[Ciklusváltozó növelése] Increment --> Decision Decision -- FALSE --> Next[Következő utasítás] Next --> Exit[] </pre>		<p>Az algoritmus ábrázoló eszközeinek ismerete, folyamatábra értelmezése.</p>
<p>III. Feladatok ciklusokra A feladat megfogalmazásával előáll a probléma, mely megoldásra vár.</p>	<p>Frontális munka tanári irányítással A feladat megoldása közösen történik, a táblára írom a diákok, kérdéseimre adott válaszait, magát a programot.</p>		

SZAKMAI GONDOLATMENET, MUNKAFÁZISOK	TANÁRI, TANULÓI TEVÉKENYSÉGEK, MUNKAFORMÁK, MÓDSZEREK	ÚJ FOGALMAK, ISMERETEK	ESZKÖZÖK ÉS FELTÉTELEK
<p><u>Egyszerű ciklusok</u> <u>S1.pas</u>: Írassuk ki a nevünket tízszer egymás alá!</p> <p>Miket kell deklarálnunk? Milyen adattípusként deklaráljuk az <i>i</i>-t?</p>	<p><i>/problémamegoldó, algoritmikus gondolkodás fejlesztése/</i></p> <p><i>Az i változót és a nevünket megadó konstans. Integerként vagy akár byte-ként, méretét tekintve ez utóbbi is megfelel.</i></p> <pre> program S1; uses crt; var i:integer; const nev='BÚGÓ CSIGA'; begin clrscr; for i:=1 to 10 do writeln(nev); readln; end. </pre>		<p>Szövegértés, elvonatkoztatás, probléma meglátása</p> <p>Sejtések megfogalmazása</p> <p>Vizuál tábla, számítógépek</p> <p>A Turbo Pascal program ismerete, használata</p>
<p><u>S2.pas</u>: Írjuk ki a nevünket többször egymás alá a képernyőre! A kiírások számát a billentyűzetről kérjük be!</p>	<p>Önálló munka</p> <p>Ellenőrzés oly módon, hogy a táblán egy diák kiegészíti az előző feladatot piros színnel.</p> <pre> program S2; uses crt; var i,db:integer; const nev='BÚGÓ CSIGA'; </pre>		<p>rugalmasság</p>

SZAKMAI GONDOLATMENET, MUNKAFÁZISOK	TANÁRI, TANULÓI TEVÉKENYSÉGEK, MUNKAFORMÁK, MÓDSZEREK	ÚJ FOGALMAK, ISMERETEK	ESZKÖZÖK ÉS FELTÉTELEK
	<pre>begin clrscr; write('Hányszor írjam ki a nevet?'); readln(db); clrscr; for i:=1 to db do writeln(nev); readln; end.</pre>		összefüggések meglátása
<p><u>A ciklusváltozó felhasználása a ciklusmagban</u> <u>S3.pas</u>: Írassuk ki egymás alá a számokat növekvő sorrendben 1-től 20-ig! Az előzőekben egy konstanst írtunk ki, most viszont változót. Hogyan valósítsuk meg?</p>	<p>Frontális munka tanári irányítással A program felkerül a táblára.</p> <p><i>/gondolkodási műveletek mozgósítása/</i></p> <p><i>A ciklusmagban magát az i-t kell kiíratnunk.</i></p> <pre>program S3; uses crt; var i:integer; begin clrscr; for i:=1 to 20 do writeln('i=',i); readln; end.</pre>		megoldás megtervezésének képessége

SZAKMAI GONDOLATMENET, MUNKAFÁZISOK	TANÁRI, TANULÓI TEVÉKENYSÉGEK, MUNKAFORMÁK, MÓDSZEREK	ÚJ FOGALMAK, ISMERETEK	ESZKÖZÖK ÉS FELTÉTELEK
<p><i>S4.pas:</i> Írassuk ki a képernyő 20. oszlopába a számokat 1-től 20-ig egymás alá, és minden szám mellé írassuk ki a kétszeresét is!</p> <p><i>Hogyan tudunk a képernyő adott pontjára pozícionálni, milyen függvény segítségével? S a 20.oszlopba hogyan? Lássuk magát a programot!</i></p>	<p><i>A gotoxy-nal.</i> <i>gotoxy(20,i)</i></p> <pre> program S4; uses crt; var i:integer; begin clrscr; for i:=1 to 20 do begin gotoxy(20,i); writeln(i,' ',i*2); end; readln; end. </pre>	<p>2 érték megjelenítése egymás mellett</p>	<p>együttműködési képesség, sejtések megfogalmazása</p> <p><i>gotoxy(x,y)</i> – képernyőn megjelenítés : x: oszlop y: sor</p>
<p><i>S6.pas:</i> Visszafelé írassuk ki egymás alá a számokat 20-tól 1-ig!</p>	<p><i>Óra elején lejegyeztük, hogy az ismétlődés irányának megfordítása a downto parancsszóval történhet, a kezdőérték maximálisra, míg a végérték minimálisra állítása mellett.</i></p>	<p><i>downto</i> - visszafelé történő megjelenítés</p>	<p>következtetési képesség megléte</p>

SZAKMAI GONDOLATMENET, MUNKAFÁZISOK	TANÁRI, TANULÓI TEVÉKENYSÉGEK, MUNKAFORMÁK, MÓDSZEREK	ÚJ FOGALMAK, ISMERETEK	ESZKÖZÖK ÉS FELTÉTELEK
Az új ismeretre épülő alapeladatok készségszintre történő begyakorlása	<pre> program S6; uses crt; var i:integer; begin clrscr; for i:=20 downto 1 do writeln('i=',i); readln; end. </pre>		
<p><u>S7.pas</u>: Adjuk össze a számokat 1-től 10-ig!</p> <p>Mivel a bemutató minden egyes része késleltetve jelenik meg, közösen megbeszélve, lépésenként haladva értelmezhetjük a ciklus működését.</p>	<p><i>Itt már nem csupán megjelenítés a feladatunk, össze kell adnunk az értékeket, s ezen eredményt megjeleníteni. Ez már kicsit összetettebb feladat. Elemezzük a ciklus egyes elemeinek értékváltozását az ismétlések során, teszteljük azokat! Az alábbi bemutató segítségével értelmezhetjük e változásokat:</i> TESZT1.htm</p> <p>Programírás előtt: az összeg kezdő értékének és a ciklusmag utasításának meghatározására kérem a diákokat.</p> <p><i>/ismeretek alkalmazása képességének fejlesztése, együttműködési képesség fejlesztése/</i></p> <pre> program S7; uses crt; var i,össz:integer; </pre>	<p>Újabb műveletvégzés a ciklusmagban</p>	<p>Netsupport School rendszer alkalmazása</p>

SZAKMAI GONDOLATMENET, MUNKAFÁZISOK	TANÁRI, TANULÓI TEVÉKENYSÉGEK, MUNKAFORMÁK, MÓDSZEREK	ÚJ FOGALMAK, ISMERETEK	ESZKÖZÖK ÉS FELTÉTELEK
	<pre>begin clrscr; ossz:=0; for i:=1 to 10 do ossz:=ossz+i; writeln('Összegük: ',ossz); readln; end.</pre>		
<p>Házi feladat</p> <p><i>HF1.pas</i>: Írassuk ki a számokat 50-től 60-ig egymás alá és mellé a számok négyzetét!</p>	<p>A házi feladat célja az órán tanultak alapos elmélyítése, begyakorlása.</p>		
<p>IV. Értékelés</p>	<p>A csoportra vonatkozóan A legügyesebbek egyéni kiemelése A magatartás, munkafegyelem értékelése</p>		

Óravázlat 2.

A tanítási óra anyaga:	Számlálós ciklus alkalmazása összetett feladatok kapcsán; lépésköz kezelése a ciklusban
A tanítási óra típusa:	Új ismeret-szerző óra
Eszközök:	Vizuál tábla, számítógépek, Turbo Pascal program, Netsupport School rendszer, Molnár Csaba – Sági Gábor: Programozás Turbo Pascal nyelven
A tanulócsoporthoz előzetes ismeretei:	Ciklusszervező utasítások (iterációk) típusai, Számlálós (for) ciklus
A tanítási óra céljai:	A korábban megszerzett ismeretek megerősítése Az egzakt, pontos feladatmegoldásra való igény fejlesztése Önálló munkára, felelősségvállalásra, nyilvános szereplésre nevelés A megtanult fogalmak és összefüggések alkalmazás szintű elsajátítása Gondolkodási műveletek mozgósítása, a tanulási folyamatok segítése A tantárgyak közötti kapcsolatok szemléltetése A programozás tantárgy iránti érdeklődés felkeltése, illetve megerősítése

Folyamatterv:

- I. Jelentés, szervezési feladatok
- II. Ismétlés, házi feladat ellenőrzése
- III. Lépésköz kezelése a ciklusban
- IV. Gyakorló feladatok
- V. Összetett feladatok
- VI.Értékelés

SZAKMAI GONDOLATMENET, MUNKAFÁZISOK	TANÁRI, TANULÓI TEVÉKENYSÉGEK, MUNKAFORMÁK, MÓDSZEREK	ÚJ FOGALMAK, ISMERETEK	ESZKÖZÖK ÉS FELTÉTELEK
I. Jelentés, szervezési feladatok	<p>A hetesek jelentése A csoport üdvözlése A napló beírása A számítógépek bekapcsolása <i>/Fegyelemre, udvariasságra, a szabályok betartására nevelés./</i></p>		napló, számítógépek
<p>II. Ismétlés, házi feladat ellenőrzése A házi feladattal kapcsolatos esetleges problémák megbeszélése</p> <p><i>HF1.pas</i>: Írassuk ki a számokat 50-től 60-ig egymás alá és mellé a számok négyzetét</p> <p>Az ismétlés egy tanuló felelése folyamán valósul meg</p>	<p>Egy tanuló még szünetben felírja a programot a táblára ellenőrzés, hibajavítás céljából.</p> <pre> program hf1; uses crt; var i:integer; begin clrscr; for i:=50 to 60 do writeln(i,'i*i'); readln; end. </pre> <p>Feleltetés az elmúlt óra anyagából <i>/Egzakt, pontos megfogalmazásra való nevelés; Megértésen alapuló fejlesztés; kifejezőképesség fejlesztése/</i></p>		<p>megoldás megtervezésének, ismeretek alkalmazásának képessége</p> <p>vizuál tábla</p>

SZAKMAI GONDOLATMENET, MUNKAFÁZISOK	TANÁRI, TANULÓI TEVÉKENYSÉGEK, MUNKAFORMÁK, MÓDSZEREK	ÚJ FOGALMAK, ISMERETEK	ESZKÖZÖK ÉS FELTÉTELEK
<p>III. Lépésköz kezelése a ciklusban <i>S9.pas: Írassuk ki a képernyőre 10-től 100-ig a számokat 10-essel egymás alá!</i> <i>A kiiratás adott, egymást követő számok esetén hogyan történik?</i> <i>Hogyan valósítható meg vajon, hogy 10-essel jelenjenek meg az értékek?</i> <i>Hogyan valósítható meg?</i> <i>Helyes, ám elegánsabb formája a megoldásnak, s komplexebb feladatok megoldása során is segítséget nyújthat, ha egy újabb változót vezetünk be ezen érték megadáására, s ezt írattuk ki. j:=i*10;</i> <i>Mindkét megoldást elfogadom.</i></p> <p>Összefüggések feltárása</p>	<p>Frontális munka tanári irányítással</p> <p><i>A ciklusváltozót írattuk ki.</i></p> <p><i>A cv-t 10-zel szorozva írattuk ki.</i> <i>writeln(i*10);</i></p> <pre> program S9; uses crt; var i,j:integer; begin clrscr; for i:=1 to 10 do begin j:=i*10; writeln(j); end; readln; end. </pre> <p><i>/rugalmasság képességének fejlesztése/</i></p>	<p>újabb változó (j) bevezetése</p>	

SZAKMAI GONDOLATMENET, MUNKAFÁZISOK	TANÁRI, TANULÓI TEVÉKENYSÉGEK, MUNKAFORMÁK, MÓDSZEREK	ÚJ FOGALMAK, ISMERETEK	ESZKÖZÖK ÉS FELTÉTELEK
<p><i>S99.pas</i>: Írassuk ki 400-tól visszafelé 20-assával a számokat 20-ig!</p> <p>Új problémahelyzetben az eddig tanult ismeretek alkalmazása</p>	<p>Önálló munka</p> <p>Ellenőrzés oly módon, hogy a táblán egy diák kiegészíti az előző feladatot piros színnel.</p> <p>program S99; uses crt; var i,j:integer;</p> <p>begin clrscr; for i:=20 downto 1 do begin j:=i*20; writeln(j); end; readln; end.</p>		<p>Problémameglátás, logikus gondolkodás</p> <p>A visszafelé iratás parancsszava a <i>downto</i> a számlálós ciklusban</p>
<p>IV. Gyakorlás</p> <p><i>Az alábbi két algoritmust számítógép nélkül elemezzük, oldjuk meg!</i></p> <p><i>S81.pas</i>: Mit csinál a program?</p> <p>for i:=1 to 12 do write (3*i, a*a);</p> <p>matematikai kompetencia fejlesztése</p>	<p>Netsupport segítségével kivetítem a tanulók képernyőjére a feladatot. A diákok kapnak egy kis gondolkodási időt.</p> <p><i>1-től 12-ig megjeleníti az egészek 3-szorosát és négyzetét.</i></p>		<p>Netsupport rendszer</p> <p>for ciklus működésének ismerete</p>

SZAKMAI GONDOLATMENET, MUNKAFÁZISOK	TANÁRI, TANULÓI TEVÉKENYSÉGEK, MUNKAFORMÁK, MÓDSZEREK	ÚJ FOGALMAK, ISMERETEK	ESZKÖZÖK ÉS FELTÉTELEK
<p><i>S82.pas</i>: Melyik program írja ki a négyzetszámokat 1-től 400-ig?</p> <pre>a, for i:=1 to 20 do write(i*i);</pre> <pre>b, for j:=0 to 19 do begin n:=n+2*j+1; write(n); end;</pre> <pre>c, for k:=1 to 20 do begin k:=k*k; write(k); end;</pre> <p>matematikai és vállalkozói kompetencia fejlesztése</p> <p><i>A c, miért nem?</i></p>	<p><i>Az a, és a b, írja ki. Mert a k értéke megváltozik a ciklusmagban, így az 1-20-ig intervallum nem teljesül.</i></p>		<p>lényeglátás, kreatív problémamegoldó képesség</p>
<p>V. Összetett feladatok <i>Újabb műveletvégzés a ciklusmagban</i></p> <p><i>S89.pas</i>: Határozzuk meg az első 10 természetes szám átlagát! <i>Soroljátok fel az első 10 természetes számot!</i> <i>Így van, tehát nehogy 1-10-ig vizsgáljátok a számokat!</i> Mielőtt hozzá látnátok a feladathoz, végezzetek tesztelést a füzetetekbe, a múlt óraihoz hasonlóan, s</p>	<p>Önálló munka</p> <p>0, 1, 2, ...9</p>		<p>természetes számok fogalmának ismerete</p>

SZAKMAI GONDOLATMENET, MUNKAFÁZISOK	TANÁRI, TANULÓI TEVÉKENYSÉGEK, MUNKAFORMÁK, MÓDSZEREK	ÚJ FOGALMAK, ISMERETEK	ESZKÖZÖK ÉS FELTÉTELEK
<p>értelmezzétek a ciklus működését!</p> <p>matematikai kompetencia fejlesztése</p>	<p>Ellenőrzés a <i>TESZT2.htm</i> állomány segítségével, levetítésével. <i>/összefüggések meglátásának, következtetőképesség fejlesztése/</i></p> <pre> program S89; uses crt; var i,ossz:integer; atl:real; begin clrscr; ossz:=0; atl:=0; for i:=0 to 9 do ossz:=ossz+i; atl:=ossz/10; writeln('átlaguk: ',atl:2:2); readln; end. </pre>		<p>Netsupport rendszer</p> <p>átlagszámítás</p>
<p>Néhány órával ezelőtt ismerkedtetek meg a tömb adattípussal, melynek egyik típusával, a vektorokkal fogunk ma egyszerű feladatokat megoldani.</p> <p><i>VEKTL.pas</i>: Egy 10 elemű vektort töltünk fel egészekkel! Mivel tömbfeltöltéssel még nem találkoztunk, kezdjük ennek értelmezését is teszteléssel!</p>	<p>Frontális munka tanári irányítással</p>		<p>tömb (vektor, mátrix) adattípusok ismerete</p>

SZAKMAI GONDOLATMENET, MUNKAFÁZISOK	TANÁRI, TANULÓI TEVÉKENYSÉGEK, MUNKAFORMÁK, MÓDSZEREK	ÚJ FOGALMAK, ISMERETEK	ESZKÖZÖK ÉS FELTÉTELEK
<p><i>A vektor deklarációját követően mire kell még figyelniük? A ciklusmag mit fog tartalmazni?</i></p> <p>Új problémahelyzetben az eddig tanult ismeretek alkalmazása</p> <p>A korábban megszerzett ismeretek megerősítése</p>	<p><i>TESZT3.htm</i> állomány segítségével kigyűjtjük a szükséges változókat, s a ciklusmenetnek megfelelően értékeket adunk nekik.</p> <p><i>/ismeretek alkalmazása képességének fejlesztése, együttműködési képesség fejlesztése/</i></p> <p><i>A ciklusmag: readln(a[i]);</i> A táblára is felkerül. program vekt1; uses crt; const n=10; var a:array[1..n] of integer; i:integer;</p> <p>begin clrscr; for i:=1 to n do readln(a[i]); end.</p>		<p>Netsupport rendszer</p> <p>Netsupport rendszer</p> <p>ismeretek alkalmazásának képessége</p> <p>vizuál tábla</p>
<p><i>VEKT2.pas</i>: Írassuk is ki az előző vektor elemei!</p> <p>Gyakorlatszerzés a programozási nyelv használatában</p> <p>A vállalkozói kompetencia fejlesztése</p>	<p>Önálló munka</p> <p>Ellenőrzés oly módon, hogy a táblán egy diák kiegészíti az előző feladatot piros színnel.</p> <p>program vekt2; uses crt; const n=10; var a:array[1..n] of integer;</p>		<p>megoldás megtervezésének képessége</p>

SZAKMAI GONDOLATMENET, MUNKAFÁZISOK	TANÁRI, TANULÓI TEVÉKENYSÉGEK, MUNKAFORMÁK, MÓDSZEREK	ÚJ FOGALMAK, ISMERETEK	ESZKÖZÖK ÉS FELTÉTELEK
	<pre> i:integer; begin clrscr; for i:=1 to n do readln(a[i]); for i:=1 to n do write(' ',a[i]); readln; end. </pre>		
<p><i>VEKT3.pas</i>: Határozzuk meg az előzően feltöltött tömb elemeinek szorzatát!</p> <p><i>A szorzat meghatározásánál mivel is kezdjük az előző programunk kiegészítését?</i></p> <p><i>Szükséges-e megadnunk a szorzatváltozó kezdő értékét?</i></p> <p><i>Mennyi legyen?</i></p> <p>Amennyiben úgy látom, hogy a tanulók nem tudják értelmezni vagy elemezni a feladat lépéseit, a táblánál teszteltem egyikőjünkkel a problémát!</p>	<p>Frontális munka tanári irányítással Program a táblára kerül.</p> <p><i>Deklaráljunk egy változót!</i> <i>Majd az s értékének változása, mint művelet bekerül a ciklusmagba.</i></p> <pre> igen s:=1 /ötletgazdagság, a meglévő ismeretek alkalmazásának, lényeglátás, kommunikációs képesség fejlesztése/ program vekt3; uses crt; const n=10; var a:array[1..n] of integer; i,s:integer; </pre>		<p>ismeretelemző-értékelő gondolkodás, önálló megfigyelés képessége</p>

SZAKMAI GONDOLATMENET, MUNKAFÁZISOK	TANÁRI, TANULÓI TEVÉKENYSÉGEK, MUNKAFORMÁK, MÓDSZEREK	ÚJ FOGALMAK, ISMERETEK	ESZKÖZÖK ÉS FELTÉTELEK
	<pre> begin clrscr; s:=1; for i:=1 to n do begin readln(a[i]); s:=s*a[i]; end; write('A tomb elemei: '); for i:=1 to n do write(' ',a[i]); writeln(' Az elemek szorzata: ',s); readln; end. </pre>		
<p>Házi feladat <u>S10.pas:</u> Írassuk ki minden második sorba a nevünket a képernyő aljáig, az első sorban kezdve. Hogyan határozzuk meg, hogy minden második sorban jelenítsen meg adatot?</p> <p><u>VEKT4.pas:</u> Tölts fel egy 5 elemű tömböt valós számokkal, s határozd meg az elemek átlagát! Mire figyeljünk?</p>	<p><i>gotoxy esetén az $x := 1$, az $y := i * 2 - 1$</i></p> <p><i>Az adattípusra, real. Az átlag új változó.</i></p>		

SZAKMAI GONDOLATMENET, MUNKAFÁZISOK	TANÁRI, TANULÓI TEVÉKENYSÉGEK, MUNKAFORMÁK, MÓDSZEREK	ÚJ FOGALMAK, ISMERETEK	ESZKÖZÖK ÉS FELTÉTELEK
<i>Amennyiben az utóbbi feladat első nekiindulásra nehéznek bizonyul, végezzetek tesztelést a tanult módon, ez biztosan segít majd!</i>			
VI. Értékelés	<p>A csoportra vonatkozóan Az önállóan jól dolgozó tanulók dicsérete (Az egyéni elismerés rendkívül ösztönző hatású) A csoport egészének értékelése. <i>Az elismerés mellett jelezzük, ha nem volt mindenkinek megfelelő a munkája. A tanulók érezzék, hogy mindenkire figyelünk.</i> A magatartás, munkafegyelem értékelése.</p>		

Óravázlat 3.

A tanítási óra anyaga:	Komplex feladatmegoldás számlálós ciklussal; elágazás ciklusban; egymásba ágyazott ciklusok
A tanítási óra típusa:	Új ismeret-szerző óra
Eszközök:	Vizuál tábla, számítógépek, Turbo Pascal program, Netsupport School rendszer, Molnár Csaba – Sági Gábor: Programozás Turbo Pascal nyelven
A tanulócsoport előzetes ismeretei:	Számlálós ciklus alkalmazása összetett feladatok kapcsán; lépésköz kezelése a ciklusban
A tanítási óra céljai:	A korábban megszerzett ismeretek megerősítése A megtanult fogalmak és összefüggések alkalmazás szintű elsajátítása Gondolkodási műveletek mozgósítása, a tanulási folyamatok segítése A tantárgyak közötti kapcsolatok szemléltetése A programozás tantárgy iránti érdeklődés felkeltése, illetve megerősítése

Folyamatterv:

- I. Jelentés, szervezési feladatok
- II. Ismétlés, házi feladat ellenőrzése
- III. Összetett feladatok, elágazás ciklusban
- IV. Egymásba ágyazott ciklusok
- V. Értékelés

SZAKMAI GONDOLATMENET, MUNKAFÁZISOK	TANÁRI, TANULÓI TEVÉKENYSÉGEK, MUNKAFORMÁK, MÓDSZEREK	ÚJ FOGALMAK, ISMERETEK	ESZKÖZÖK ÉS FELTÉTELEK
I. Jelentés, szervezési feladatok	<p>A hetesek jelentése A csoport üdvözlése A napló beírása A számítógépek bekapcsolása <i>/Fegyelemre, udvariasságra, a szabályok betartására nevelés./</i></p>		napló, számítógépek
<p>II. Ismétlés, házi feladat ellenőrzése</p> <p>A házi feladattal kapcsolatos esetleges problémák megbeszélése <i>S10.pas:</i> Írassuk ki minden második sorba a nevünket a képernyő aljáig, az első sorban kezdve.</p> <p>Az új ismeretre vonatkozó alapeladatok készség szintre történő begyakoroltatása</p>	<p>Egy tanuló még szünetben felírja a programokat a táblára ellenőrzés, hibajavítás céljából.</p> <pre> program S10; uses crt; var i,j:integer; begin clrscr; for i:=1 to 12 do begin j:=i*2-1; gotoxy(1,j); writeln('BÚGÓ CSIGA'); end; readln; end. </pre>		vizuál tábla

SZAKMAI GONDOLATMENET, MUNKAFÁZISOK	TANÁRI, TANULÓI TEVÉKENYSÉGEK, MUNKAFORMÁK, MÓDSZEREK	ÚJ FOGALMAK, ISMERETEK	ESZKÖZÖK ÉS FELTÉTELEK
<p><i>VEKT4.pas:</i> Tölts fel egy 5 elemű tömböt valós számokkal, s határozd meg az elemek átlagát!</p> <p>A korábban megszerzett ismeretek megerősítése</p> <p>Felelevenítjük az előző órán tanultakat néhány mondatban, hiszen erre építünk a mai órán</p>	<pre> program vekt4; uses crt; const n=5; var a:array[1..n] of real; i:integer; s,atl:real; begin clrscr; s:=0; atl:=0; for i:=1 to n do begin readln(a[i]); s:=s+a[i]; end; atl:=s/n; write('A tömb elemei: '); for i:=1 to n do write(' ',a[i]:2:3); writeln(' Az elemei átlaga: ',atl:5:2); readln; end. </pre> <p>Tanári kérdések, feleletek önkéntes jelentkezés alapján. (Egzakt, pontos megfogalmazásra való nevelés; Kifejezőképesség, megértésen alapuló fejlesztés)</p>		<p>tömb (vektor, mátrix) adattípusok ismerete</p> <p>A megtanult fogalmak és összefüggések alkalmazás</p> <p>kommunikációs készség</p>

SZAKMAI GONDOLATMENET, MUNKAFÁZISOK	TANÁRI, TANULÓI TEVÉKENYSÉGEK, MUNKAFORMÁK, MÓDSZEREK	ÚJ FOGALMAK, ISMERETEK	ESZKÖZÖK ÉS FELTÉTELEK
<p>III. Gyakorlás</p> <p><i>SSZ.pas</i>: Készítsen a gép számtani sorozatot (adott az első szám, a differencia ,s elemszám)!</p> <p><i>Tesztelés során értelmezzük a feladatot!</i></p> <p>Összefüggések feltárása</p> <p>Komplexitás a matematikával.</p>	<p>Frontális munka tanári irányítással A program a táblára kerül.</p> <p>TESZT4.htm állomány segítségével kigyűjtjük a szükséges változókat, s a ciklus menetének megfelelően értékeket adunk neki. <i>/ismeretek alkalmazása képességének fejlesztése, együttműködési képesség fejlesztése/</i></p> <pre> program SSZ; uses crt; var i,k:integer; const a=10;d=8;esz=5; begin clrscr; k:=a; for i:=1 to esz do begin write (' ',k); k:=k+d; end; readln; end. </pre>		<p>Számtani sorozat, annak képzésének ismerete; differencia értelmezése</p> <p>Netsupport rendszer</p>

SZAKMAI GONDOLATMENET, MUNKAFÁZISOK	TANÁRI, TANULÓI TEVÉKENYSÉGEK, MUNKAFORMÁK, MÓDSZEREK	ÚJ FOGALMAK, ISMERETEK	ESZKÖZÖK ÉS FELTÉTELEK
<p><i>SSZM.pas</i>: Az előző példánál írjuk ki a szomszédos elemek különbségét!</p> <p>A korábban megszerzett ismeretek megerősítése</p>	<p>Önálló munka; ellenőrzés oly módon, hogy a táblán egy diák kiegészíti az előző feladatot piros színnel.</p> <pre> program SSZM; uses crt; var i,k,s:integer; const a=5;esz=5;d=2; begin clrscr; write('Szamtani:'); s:=a; for i:=1 to esz do begin write (' ',s); s:=s+d; end; writeln(' különbség:',d); readln; end. </pre>		<p>A megtanult összefüggések alkalmazás</p>

SZAKMAI GONDOLATMENET, MUNKAFÁZISOK	TANÁRI, TANULÓI TEVÉKENYSÉGEK, MUNKAFORMÁK, MÓDSZEREK	ÚJ FOGALMAK, ISMERETEK	ESZKÖZÖK ÉS FELTÉTELEK
<p><i>Elágaztatás ciklusban</i> <i>S810.pas: Írassuk ki 50-ig az összes pozitív, páros számot!</i></p> <p><i>Korábban tanultak alapján hogyan tudnánk megoldani ezt a vizsgáldást: pozitív, páros?</i></p> <p>A korábban megszerzett ismeretek megerősítése</p>	<p>Frontális munka tanári irányítással</p> <p><i>/ötletgazdagság, sejtések megfogalmazása/</i></p> <p><i>Vizsgálni: if..then utasítás segítségével.</i> <i>Pozitív:1-50.</i> <i>Páros: $i \bmod 2=0$.</i></p> <pre> program S810; uses crt; var i:integer; begin clrscr; for i:=1 to 50 do if (i mod 2=0) then write(' ',i); readln; end. </pre>	<p>elágaztatás ciklusban</p>	<p>if..then szelekiós utasítás alkalmazásában való jártasság</p> <p>maradékos osztás, maradékképzés:mod</p> <p>A megtanult fogalmak és összefüggések alkalmazás</p> <p>vizuál tábla</p>

SZAKMAI GONDOLATMENET, MUNKAFÁZISOK	TANÁRI, TANULÓI TEVÉKENYSÉGEK, MUNKAFORMÁK, MÓDSZEREK	ÚJ FOGALMAK, ISMERETEK	ESZKÖZÖK ÉS FELTÉTELEK
<p><i>Tovább nehezédnek feladataink. A kiíratásra már sok példát néztünk, de próbáljuk megoldani a következő feladatot!</i></p> <p>IV. Egymásba ágyazott ciklusok</p> <p><i>S11.pas: Írassuk tele a képernyőre első 20 sorát „#” karakterekkel!</i></p> <p><i>Vizsgáljuk meg, a sorazonosító milyen értékhatár között mozoghat? Az oszlopazonosító? Két változó különböző értékhatárok között mozog, mely már nem oldható meg egy for ciklussal. Egymásba ágyazunk 2 ciklust, s ennek segítségével próbáljuk megoldani a problémát.</i></p> <p>Összefüggések megláttatása</p>	<p>1-20 1-80</p> <p>A feladat megoldását a táblára írom.</p> <pre> program S11; uses crt; var sor,oszl:integer; begin clrscr; for sor:=1 to 20 do for oszl:=1 to 80 do begin gotoxy(oszl,sor); write('#'); </pre>	<p>egymásba ágyazott ciklusok</p>	<p>Megismerés, konstruálás iránti vágy</p>

SZAKMAI GONDOLATMENET, MUNKAFÁZISOK	TANÁRI, TANULÓI TEVÉKENYSÉGEK, MUNKAFORMÁK, MÓDSZEREK	ÚJ FOGALMAK, ISMERETEK	ESZKÖZÖK ÉS FELTÉTELEK
	<p>end; readln; end.</p> <p><i>//logikus gondolkodás, megfigyelőképesség fejlesztése/</i></p>		
<p><i>S12.pas:</i> Írassuk tele a képernyőre első 20 sorát „#” karakterekkel úgy, hogy nyomon követhető legyen a kiíratás sorrendje (várakoztatással)!</p>	<p>A módosítást a táblán lévő program kiegészítésével tüntetem fel.</p> <p>A delay()függvény segítségével –a zárójelben értéket megadva-érhetjük el azt, hogy a program egyes lépései között kis szünetet tartva, ne egyszerre írja ki az adatokat.</p> <pre> program S12; uses crt; var sor,oszl:integer; begin clrscr; for sor:=1 to 20 do for oszl:=1 to 80 do begin gotoxy(oszl,sor); write('#'); delay(500); end; readln; end.</pre>	<p>várakoztatás: delay()</p>	<p>Kíváncsiság, rugalmasság</p>

SZAKMAI GONDOLATMENET, MUNKAFÁZISOK	TANÁRI, TANULÓI TEVÉKENYSÉGEK, MUNKAFORMÁK, MÓDSZEREK	ÚJ FOGALMAK, ISMERETEK	ESZKÖZÖK ÉS FELTÉTELEK
<p><i>A múlt órán a vektorokkal kapcsolatban oldottunk meg feladatokat, ma ha már az egymásba ágyazott ciklusokról is szó esett, nézzünk néhány feladatot a tömb másik fajtájára, a mátrixra!</i></p> <p><i>A kétdimenziós mátrixokkal foglalkoztatok, folytassuk megismerésüket!</i></p> <p><i><u>MATRIX1.pas</u>:Egy 3x3-as mátrixot töltünk fel egészekkel!</i></p> <p><i>Értelmezzük a feladatot tesztelés során!</i></p> <p><i>Az előző feladathoz hasonlóan, ha már 2 változóról van szó, egymásba kell ágyaznunk a ciklusokat!</i></p> <p>Összefüggések megláttatása</p>	<p><i>TESZT_M.htm</i> állomány segítségével kigyűjtjük a szükséges változókat, s a ciklus menetének megfelelően értékeket adunk neki. A megoldást a táblára felírom, a diákok másolják.</p> <p><i>/összefüggések meglátásának, ismeretelemző gondolkodás, együttműködési képesség fejlesztése/</i></p> <pre> program matrix1; uses crt; const n=3; var a:array[1..n,1..n] of integer; i,j:integer; begin clrscr; for i:=1 to n do for j:=1 to n do readln(a[i][j]); readln; end. </pre>	<p>mátrix feltöltése</p>	<p>Netsupport School rendszer</p> <p>tömb (vektor, mátrix) adattípusok ismerete</p> <p>vizuál tábla</p>

SZAKMAI GONDOLATMENET, MUNKAFÁZISOK	TANÁRI, TANULÓI TEVÉKENYSÉGEK, MUNKAFORMÁK, MÓDSZEREK	ÚJ FOGALMAK, ISMERETEK	ESZKÖZÖK ÉS FELTÉTELEK
<p><i>MATRIX2.pas</i>: Jelenítsük meg az előző mátrix elemeit mátrix formában!</p> <p><i>Mivel a feltöltés ciklusok egymásba ágyazását igényelte, a kiíratás is ily módon valósulhat meg. A sortörés miatt egy újabb writeln-t szükséges alkalmaznunk.</i></p>	<p>Az előző feladat kiegészítése a táblánál.</p> <pre> program matrix2; uses crt; const n=3; var a:array[1..n,1..n] of integer; i,j:integer; begin clrscr; for i:=1 to n do for j:=1 to n do readln(a[i][j]); for i:=1 to n do begin for j:=1 to n do write(",a[i][j]); writeln; end; readln; end. </pre> <p><i>/Algoritmikus gondolkodás fejlesztése/</i></p>	<p>mátrix kiíratása mátrix formában</p>	<p>vizuál tábla következtető képesség</p>
<p>Házi feladat <i>SM.pas</i>: Készítsen a gép mértani sorozatot(adott az első szám, a qvóciens ,s elemszám)! Majd írja ki a szomszédos elemek hányadosát!</p>	<p>A házi feladat célja az órán tanultak alapos elmélyítése, begyakorlása.</p>		

SZAKMAI GONDOLATMENET, MUNKAFÁZISOK	TANÁRI, TANULÓI TEVÉKENYSÉGEK, MUNKAFORMÁK, MÓDSZEREK	ÚJ FOGALMAK, ISMERETEK	ESZKÖZÖK ÉS FELTÉTELEK
V. Értékelés	A csoportra vonatkozóan A legügyesebbek egyéni kiemelése A magatartás, munkafegyelem értékelése		

Óravázlat 4.

A tanítási óra anyaga:

A számlálós ciklus alkalmazása

A tanítási óra típusa:

Gyakorló óra

Eszközök:

Vizuál tábla, számítógépek, Turbo Pascal program, Netsupport School rendszer,
Molnár Csaba – Sági Gábor: Programozás Turbo Pascal nyelven

A tanulócsoporthoz előzetes ismeretei:

Számlálós ciklus; elágazás ciklusban; egymásba ágyazott ciklusok; lépésköz kezelése; vektorok, mátrixok feltöltése ciklusok segítségével

A tanítási óra céljai:

A korábban megszerzett ismeretek megerősítése

Önálló munkára, felelősségvállalásra, nyilvános szereplésre nevelés

A megtanult fogalmak és összefüggések alkalmazás szintű elsajátítása

Gondolkodási műveletek mozgósítása, a tanulási folyamatok segítése

A tantárgyak közötti kapcsolatok szemléltetése

A programozás tantárgy iránti érdeklődés felkeltése, illetve megerősítése

Az alapeladatok megoldási algoritmusának készségszintre fejlesztése

Folyamatterv:

- I. Jelentés, szervezési feladatok
- II. Ismétlés, házi feladat ellenőrzése
- III. Gyakorlás
- IV. Értékelés

SZAKMAI GONDOLATMENET, MUNKAFÁZISOK	TANÁRI, TANULÓI TEVÉKENYSÉGEK, MUNKAFORMÁK, MÓDSZEREK	ÚJ FOGALMAK, ISMERETEK	ESZKÖZÖK ÉS FELTÉTELEK
I. Jelentés, szervezési feladatok	<p>A hetesek jelentése A csoport üdvözlése A napló beírása A számítógépek bekapcsolása <i>(Fegyelemre, udvariasságra, a szabályok betartására nevelés.)</i></p>		napló, számítógépek
<p>II. Ismétlés, házi feladat ellenőrzése <i>Bízom benne, hogy mindenkinek hibátlanul sikerült megoldani a házi feladatot, hiszen a múlt órán megoldott számtani sorozatos feladatokat segíthetett!</i></p> <p>A házi feladattal kapcsolatos esetleges problémák megbeszélése <u>SM.pas</u>: Készítsen a gép mértani sorozatot (adott az első szám, a q vóciens, s elemszám)! Majd írja ki a szomszédos elemek hányadosát!</p>	<p>Egy tanuló még szünetben felírja a programot a táblára, s onnan történik az ellenőrzés, hibajavítás.</p> <pre> program SM; uses crt; var i,k:integer; const a=5;q=2;esz=3; begin clrscr; k:=a; for i:=1 to esz do begin write (' ',k); k:=k*q; end; writeln(' hányados:',q); readln; end. </pre>		<p>vizuál tábla</p> <p>matematikai fogalmak</p> <p>A megtanult fogalmak és összefüggések alkalmazása</p>

SZAKMAI GONDOLATMENET, MUNKAFÁZISOK	TANÁRI, TANULÓI TEVÉKENYSÉGEK, MUNKAFORMÁK, MÓDSZEREK	ÚJ FOGALMAK, ISMERETEK	ESZKÖZÖK ÉS FELTÉTELEK
<p><i>A mai órán a számlálás ciklusról szerzett ismereteinket felhasználva oldunk meg komplex feladatokat.</i></p> <p>III. Gyakorlás</p> <p>1. feladat</p> <p><u>S5.pas</u>: Írassunk ki a képernyőre 20 db „#” karaktert a bal felső sarokból indulva átlósan!</p> <p><i>Gondoljuk át, hogyan változik a sor és oszlop értéke, mely értékeket vesz fel, ha átlósan haladunk?</i></p> <p><i>Így ha csupán 1 értékre lesz szükségünk, 1 változó, akkor is a megszokott egymásba ágyazott ciklust alkalmazzuk?</i></p> <p><i>Próbáljátok önállóan megoldani a feladatot!</i></p>	<p>Önálló munka; javítás, ellenőrzés oly módon, hogy a helyes példát tartalmazó diákképernyőt közzé teszem a többieknek, s az alapján javítanak. Amennyiben sok a hibalehetőség, egy diák felírja a táblára és onna történik a javítás.</p> <p>A helyes megoldást plusszal (+) jutalmazom. (Megszokott módszer:10 plussz egy 5-t ér.)</p> <p><i>1-1;2-2;3-3;... ugyanazokat</i></p> <p><i>Nem, nincs rá szükség, csupán 1 ciklust használunk.</i></p> <p><i>/lényeglátás fejlesztése/</i></p> <pre> program S5; uses crt; var i:integer; begin clrscr; for i:=1 to 20 do begin </pre>		<p><i>gotoxy(x,y) – képernyőn megjelenítés</i></p> <p><i>egymásba ágyazott ciklusok - 2 változó esetén szükséges</i></p> <p><i>probléma-érzékenység</i></p>

SZAKMAI GONDOLATMENET, MUNKAFÁZISOK	TANÁRI, TANULÓI TEVÉKENYSÉGEK, MUNKAFORMÁK, MÓDSZEREK	ÚJ FOGALMAK, ISMERETEK	ESZKÖZÖK ÉS FELTÉTELEK
<p>Önálló munkára, felelősségvállalásra, nyilvános szereplésre nevelés</p> <p>Az alapfeladatok megoldási algoritmusának készségi szintre fejlesztése</p>	<pre>gotoxy(i,i); writeln('#'); end; readln; end.</pre>		
<p>2. feladat</p> <p><i>Skamat.pas:</i> Tegyük A forintot évi P %-os kamatra takarékbba! Írassuk ki, hogyan alakul a megtakarított pénzünk N év alatt!</p> <p><i>Lássuk a változókat! Mire figyeljünk az adattípusoknál?</i></p> <p><i>Az A,P,N-eket hogyan deklaráljuk? Próbáljátok a kívánt műveletet egyedül meghatározni!</i></p> <p>matematikai, vállalkozói kompetencia fejlesztése</p>	<p>Önálló munka; javítás, ellenőrzés oly módon, hogy a helyes példát tartalmazó diák képernyőjét közzé teszem a többieknek, s az alapján javítanak. A helyes megoldást plusszal (+) jutalmazom.</p> <p><i>A %-os kamat miatt valószínűleg nem egész értékkel fogunk dolgozni, válasszuk a real típust!</i></p> <p><i>Konstansként adjuk meg!</i></p> <pre>program Skamat; uses crt; var i:integer; k:real; const a=50000;p=8;n=5; begin clrscr; k:=a; for i:=1 to n do</pre>		<p>logikus gondolkodás</p> <p>A százalékszámítás - alap, százalékláb, százaléktérkép fogalma, értelmezése; százaléktérkép kiszámítása.</p>

SZAKMAI GONDOLATMENET, MUNKAFÁZISOK	TANÁRI, TANULÓI TEVÉKENYSÉGEK, MUNKAFORMÁK, MÓDSZEREK	ÚJ FOGALMAK, ISMERETEK	ESZKÖZÖK ÉS FELTÉTELEK
	<pre>begin writeln (i,',v: ',k:10:2); k:=k*(1+p/100); end; readln; end.</pre>		megoldás megtervezésnek képessége
<p>3. feladat <i>S811.pas</i>: Írassuk ki 99-től csökkenő sorrendben az összes pozitív, 3-mal nem osztható egészet!</p> <p>A korábban megszerzett ismeretek megerősítése</p> <p>Önálló munkára, felelősségvállalásra, nyilvános szereplésre nevelés</p>	<pre>program S811; uses crt; var i:integer; begin clrscr; for i:=99 downto 1 do if (i mod 3>0) then write(' ',i); readln; end.</pre> <p><i>/Gondolkodási műveletek mozgósítása/</i></p>		oszthatóság szabálya, maradékos osztás: mod használata downto-visszafelé történő vizsgálódás, kiíratás probléma-érzékenység
<p>4. feladat <i>S812.pas</i>: Jelenítsük meg egy adott egész szám valódi osztóit! <i>Melyek egy szám nem valódi osztói?</i> <i>Így igaz, tehát figyeljete a ciklusváltozó kezdő és végértékének meghatározásánál! Vegyétek</i></p>	1 és önmaga		valódi és nem valódi osztó fogalma, értelmezése

SZAKMAI GONDOLATMENET, MUNKAFÁZISOK	TANÁRI, TANULÓI TEVÉKENYSÉGEK, MUNKAFORMÁK, MÓDSZEREK	ÚJ FOGALMAK, ISMERETEK	ESZKÖZÖK ÉS FELTÉTELEK
<p><i>figyelembe!</i></p> <p>matematikai kompetencia fejlesztése</p>	<pre> program S812; uses crt; var i,a:integer; begin clrscr; writeln('Kerem a számot! '); readln(a); for i:=2 to a-1 do if (a mod i=0) then write(' ',i); readln; end. </pre> <p><i>/Gondolkodási műveletek mozgósítása/</i></p>		<p>a tanult ismeretek alkalmazása</p> <p>lényeglátás, kreatív problémamegoldó képesség</p>
<p>5. feladat</p> <p><u>VEKT5.pas</u>: Töltsünk fel egy 8 elemű vektort egészekkel, s számoljuk meg a páratlanokat!</p> <p><i>A számlálásnál mire figyeljünk?</i></p> <p>Összefüggések feltárása</p>	<p><i>db:=0; kezdőértékre</i></p> <p><i>/sejtések megfogalmazása/</i></p> <p>Amennyiben szükséges, tesztelés segítségével elemezhetjük a feladat megoldását:</p> <p>TESZT5.htm állomány révén.</p> <p><i>/megfigyelőképesség fejlesztése/</i></p> <pre> program vekt5; uses crt; const n=8; var a:array[1..n] of integer; </pre>		<p>páratlan számok fogalma, tulajdonságaik, oszthatóság</p>

SZAKMAI GONDOLATMENET, MUNKAFÁZISOK	TANÁRI, TANULÓI TEVÉKENYSÉGEK, MUNKAFORMÁK, MÓDSZEREK	ÚJ FOGALMAK, ISMERETEK	ESZKÖZÖK ÉS FELTÉTELEK
<p>A korábban megszerzett ismeretek megerősítése</p> <p>Az alapfeladatok megoldási algoritmusának készségi szintre fejlesztése</p>	<pre> i,db:integer; begin clrscr; db:=0; for i:=1 to n do begin readln(a[i]); if (a[i] mod 2>0) then db:=db+1; end; write('A tomb elemei: '); for i:=1 to n do write(' ',a[i]); writeln(' Paratlanok szama: ',db); readln; end. </pre>		<p>lényeglátás, kreatív problémamegoldó képesség</p>
<p>6. feladat <i>S13.pas</i>: Készítsünk a képernyő közepére egy működő digitális órát, ami 5 percig jár!</p> <p>Amennyiben nem jut idő erre a feladatra, szorgalmi feladatként feladom jövő órára.</p>	<pre> program ora; uses crt; var mp,perc:integer; begin clrscr; for perc:=0 to 4 do for mp:=0 to 59 do begin gotoxy(37,12); writeln(perc,' : ',mp:2); delay(5000); end; end; end. </pre>		<p>Az idő mértékegységei</p>

SZAKMAI GONDOLATMENET, MUNKAFÁZISOK	TANÁRI, TANULÓI TEVÉKENYSÉGEK, MUNKAFORMÁK, MÓDSZEREK	ÚJ FOGALMAK, ISMERETEK	ESZKÖZÖK ÉS FELTÉTELEK
	end; end.		
Házi feladat <i>SS10.pas</i> : Írassuk ki 101-től 50-ig csökkenő sorrendben az 5-tel osztható számok kétszeresét!	A házi feladat célja az órán tanultak alapos elmélyítése, begyakorlása.		További műveletek a ciklusmagban-oszthatóság
IV. Értékelés	A csoportra vonatkozóan A szerzett plusszok beírása a kisnaplóba, ezúton a legügyesebbek egyéni kiemelése A magatartás, munkamorál minősítése		

Óravázlat 5.

A tanítási óra anyaga:

Az elől tesztelő ciklusok fogalma, működési elve; alkalmazása

A tanítási óra típusa:

Új ismeret-szerző óra

Eszközök:

Vizuál tábla, számítógépek, Turbo Pascal program, Netsupport School rendszer,

Molnár Csaba – Sági Gábor: Programozás Turbo Pascal nyelven

A tanulócsoporthoz előzetes ismeretei:

A számlálós ciklus fogalma, szemantikája, alkalmazása

A tanítási óra céljai:

Az új ismeretek beágyazása a meglévő ismeretrendszerbe

Az új ismeretek összefüggéseinek feltárása, gyakorlati jelentőségének megértése

Gondolkodási műveletek mozgósítása, a tanulási folyamatok segítése

A programozás tantárgy iránti érdeklődés felkeltése, illetve megerősítése

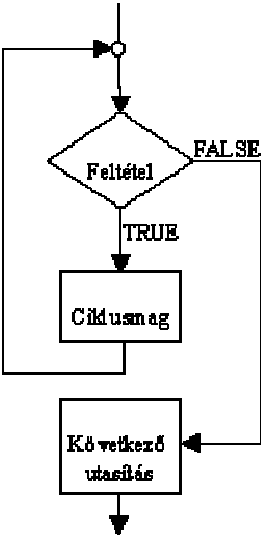
Alapvető programozási fogalmak ismerete

Folyamatterv:

- I. Jelentés, szervezési feladatok
- II. Ismétlés, házi feladat ellenőrzése
- III. Elől tesztelő ciklusok
- IV. Feladatok ciklusokra
- V. Értékelés

SZAKMAI GONDOLATMENET, MUNKAFÁZISOK	TANÁRI, TANULÓI TEVÉKENYSÉGEK, MUNKAFORMÁK, MÓDSZEREK	ÚJ FOGALMAK, ISMERETEK	ESZKÖZÖK ÉS FELTÉTELEK
I. Jelentés, szervezési feladatok	<p>A hetesek jelentése A csoport üdvözlése A napló beírása A számítógépek bekapcsolása <i>/Fegyelemre, udvariasságra, a szabályok betartására nevelés./</i></p>		napló, számítógépek
<p>II. Ismétlés, házi feladat ellenőrzése A házi feladattal kapcsolatos esetleges problémák megbeszélése</p> <p><i>SS10.pas</i>: Írassuk ki 101-től 50-ig csökkenő sorrendben az 5-tel osztható számok kétszeresét!</p>	<p>Egy diák már óra előtt felírja a táblára a programot. Megbeszéljük helyességét, s aki tévesztett, kijavítja hibáját.</p> <pre> program SS10; uses crt; var i,j:integer; begin clrscr; for i:=101 downto 50 do if (i mod 5=0) then writeln(i*2); readln; end. </pre>		<p>További műveletek a ciklusmagban-oszthatóság</p> <p>vizuál tábla</p>

SZAKMAI GONDOLATMENET, MUNKAFÁZISOK	TANÁRI, TANULÓI TEVÉKENYSÉGEK, MUNKAFORMÁK, MÓDSZEREK	ÚJ FOGALMAK, ISMERETEK	ESZKÖZÖK ÉS FELTÉTELEK
<p>III. Elöl tesztelő ciklusok</p>	<p>Az új ismeret felkerül a táblára. Az új fogalmakat diktálás során rögzítik a tanulók a füzetükbe.</p> <p>Azon ciklusokat, melyek esetében nem tudjuk előre, hogy hányszor kell végrehajtaniuk az adott utasítássorozatot, nevezzük tesztelő ciklusoknak.</p> <p>Az ismétlés eldöntésének helyétől függően, megkülönböztetünk elöl tesztelő és hátul tesztelő ciklust.</p> <p><u>WHILE</u> utasítás (elöl tesztelő ciklus)</p> <p style="padding-left: 40px;">WHILE logikai kifejezés DO utasítás</p> <p><i>Másképpen:</i></p> <p style="padding-left: 40px;">while=míg, amíg</p> <p>Ciklus, <i>amíg</i> feltétel a <i>ciklusmag</i> utasításai</p> <p>Ciklus vége</p> <p>while logikai kifejezés do ciklusmag</p> <p>ha a ciklusmag több utasításból áll, akkor:</p> <p>while logikai kifejezés do begin ciklusmag end</p>	<p>while..do</p> <p>„elöl tesztelő”</p>	<p>alapvető programozási fogalmak ismerete</p>

SZAKMAI GONDOLATMENET, MUNKAFÁZISOK	TANÁRI, TANULÓI TEVÉKENYSÉGEK, MUNKAFORMÁK, MÓDSZEREK	ÚJ FOGALMAK, ISMERETEK	ESZKÖZÖK ÉS FELTÉTELEK
<p>Az utasítás folyamatábrája is felkerül a táblára annak érdekében, hogy az iteráció részei közötti összefüggések, valamint szemantikájának megértését segítse. Így az algoritmus ábrázoló eszközeinek alkalmazása révén értelmezik azt.</p> <p>Az új ismeretek összefüggéseinek feltárása</p>	<p>Az utasítás folyamatábrája:</p>  <pre> graph TD Start(()) --> Feltétel{Feltétel} Feltétel -- TRUE --> Ciklusmag[Ciklusmag] Ciklusmag --> Feltétel Feltétel -- FALSE --> Kovetkezo[Következő utasítás] Kovetkezo --> End(()) </pre> <p>A do kulcsszó utáni utasítás, az ún. ciklusmag végrehajtása mindig addig ismétlődik, míg a while kulcsszó után álló logikai kifejezés értéke true.</p> <p>A kifejezés kiértékelésére a ciklusmag végrehajtása előtt kerül sor, tehát ha az már induláskor hamis (false), akkor a ciklusmag egyszer sem hajtódik végre.</p>		<p>Az algoritmus ábrázoló eszközeinek ismerete, folyamatára értelmezése.</p> <p>összefüggések meglátása</p>

SZAKMAI GONDOLATMENET, MUNKAFÁZISOK	TANÁRI, TANULÓI TEVÉKENYSÉGEK, MUNKAFORMÁK, MÓDSZEREK	ÚJ FOGALMAK, ISMERETEK	ESZKÖZÖK ÉS FELTÉTELEK
<p>IV. Feladatok elől tesztelő ciklusokra</p> <p>Az új ismeretek beágyazása a meglévő ismeretrendszerbe</p> <p><i>Próbáljuk értelmezni, alkalmazni konkrét feladat kapcsán!</i></p> <p>1. feladat <u>El.pas</u>: Olvassunk be valós számokat, míg pozitívak, s számláljuk meg, hányat olvastunk be!</p> <p><i>Mi lesz a logikai kifejezés a ciklusban?</i> <i>Ahhoz, hogy vizsgálni tudjuk az a-t, mit kell tennünk?</i></p> <p>Gondolkodási műveletek mozgósítása</p>	<p>Frontális munka tanári irányítással A programok felkerülnek a táblára</p> <p><i>/sejtések megfogalmazása/</i> A ciklus működését megvilágítja a TESZT6.htm állomány.</p> <p><i>Hogy pozitív. $a > 0$;</i></p> <p><i>Beolvasni még a ciklus előtt.</i></p> <pre> program e1; uses crt; var db:integer; a:real; begin clrscr; writeln('Kérem a számokat!'); db:=0; readln(a); </pre>		<p>Netsupport rendszer</p> <p>Szövegértés, elvonatkoztatás, probléma meglátása</p>

SZAKMAI GONDOLATMENET, MUNKAFÁZISOK	TANÁRI, TANULÓI TEVÉKENYSÉGEK, MUNKAFORMÁK, MÓDSZEREK	ÚJ FOGALMAK, ISMERETEK	ESZKÖZÖK ÉS FELTÉTELEK
<p>2. feladat <u>E2.pas</u>: Kérjünk be 50-nél kisebb pozitív egészeket ,s jelenítsük meg ezeket négyzetgyökükkel együtt(2 tizedesjegy pontosságig)! A négyzetgyök adattípusaként melyiket érdemes választani?</p> <p>matematikai kompetencia fejlesztése</p>	<pre>while (a>0) do begin db:=db+1; readln(a); end; writeln('Beolvasottak száma:',db); readln; end.</pre> <p><i>/lényegkiemelés, algoritmikus gondolkodás fejlesztése/</i></p> <p><i>real, valós</i></p> <pre>program e2; uses crt; var a:integer; g:real; begin clrscr; writeln('Kérem a számokat!'); readln(a); while (a<50) do</pre>		<p>adattípusok ismerete, célszerű használata</p> <p>négyzetgyökfüggvény: sqrt</p>

SZAKMAI GONDOLATMENET, MUNKAFÁZISOK	TANÁRI, TANULÓI TEVÉKENYSÉGEK, MUNKAFORMÁK, MÓDSZEREK	ÚJ FOGALMAK, ISMERETEK	ESZKÖZÖK ÉS FELTÉTELEK
<p>3.feladat <i>E3.pas:</i> Kérjünk be egy számot 1 és 10 között-ellenőrizzük-, s írassuk ki a képernyő közepére! Melyik művelet fog ismétlődni, míg a feltétel igaz, mit kell tartalmaznia a ciklusmagnak? Hogyan adjuk meg a kettős feltételt, hogy igaz lehessen?</p> <p>Gondolkodási műveletek mozgósítása</p>	<pre>begin writeln(a,' ',sqrt(a):5:2); readln(a); end; end.</pre> <p><i>/ötletgazdagság, rugalmasság/</i></p> <p><i>A szám beolvasását.</i></p> <p><i>(a<1) or (a>10)</i></p> <pre>program e3; uses crt; var a:integer; begin clrscr; writeln('Mondj egy számot 1 és 10 között! '); readln(a); while (a<1) or (a>10) do begin</pre>	<p>kettős feltétel megadása (or, and)</p>	<p>alkalmazása</p> <p>valós értéket 2 tizedesjegy pontossággal megjelenít</p> <p>műveleti jelek helyes használata</p>

SZAKMAI GONDOLATMENET, MUNKAFÁZISOK	TANÁRI, TANULÓI TEVÉKENYSÉGEK, MUNKAFORMÁK, MÓDSZEREK	ÚJ FOGALMAK, ISMERETEK	ESZKÖZÖK ÉS FELTÉTELEK
<p>4. feladat <u>Művelet a ciklusban</u> <u>E4.pas</u>: Kérjünk be nem negatív számokat ,s határozzuk meg ezek összegét, a bekérést egy negatív szám zárja!</p>	<pre> write('1 és 10 közötti számot kérek! Mondj másikat!'); readln(a); end; clrscr; gotoxy(35,12); writeln('A számod :',a); readln; end. Önálló munka, ellenőrzés vizuál táblán /összefüggések meglátása, következtető képesség fejlesztése/ program e4; uses crt; var a,s:integer; begin clrscr; writeln('Kérem a számokat, ha vége, a negatív számot adj meg!'); </pre>	<p>újabb művelet a ciklusban</p>	<p>gotoxy függvény használata</p> <p>megoldás megtervezésének képessége ismeretek alkalmazásának képessége</p>

SZAKMAI GONDOLATMENET, MUNKAFÁZISOK	TANÁRI, TANULÓI TEVÉKENYSÉGEK, MUNKAFORMÁK, MÓDSZEREK	ÚJ FOGALMAK, ISMERETEK	ESZKÖZÖK ÉS FELTÉTELEK
<p><i>Ehhez hasonló feladatot oldottunk meg számlálós ciklusra, próbálkozzatok meg a megoldásával elől tesztelős ciklusra is!</i></p> <p>kezdeményezőképeség, vállalkozói kompetencia fejlesztése</p> <p>5. feladat <i>E5.pas: Írjunk programot, mely egy tanuló átlagát számítja ki! Az érdemjegyeket egyesével kérjük be, a bekérést a 0 zárja!</i></p> <p><i>Határozzuk meg a változókat és adattípusaikat!</i></p>	<pre>s:=0; readln(a); while (a>=0) do begin s:=s+a; readln(a); end; writeln('Az ”összegük: ',s); readln; end.</pre> <p>Frontális munka tanári irányítással</p> <p><i>/ismeretelemző-értékelő gondolkodás, kommunikációs képesség fejlesztése/</i></p> <pre>program e5; uses crt; var jegy,db,s:integer;</pre>		<p>logikus, problémamegoldó gondolkodás</p>

SZAKMAI GONDOLATMENET, MUNKAFÁZISOK	TANÁRI, TANULÓI TEVÉKENYSÉGEK, MUNKAFORMÁK, MÓDSZEREK	ÚJ FOGALMAK, ISMERETEK	ESZKÖZÖK ÉS FELTÉTELEK
<p><i>Az előző feladatokhoz hasonlóan, a ciklus előtt már beolvassuk a jegyet, ezt figyelembe véve mi legyen az s és mi a db kezdőértéke?</i></p> <p><i>A beolvasásokat figyelembe véve a db értékét változatlanul kiírathatjuk a ciklus lezárása után?</i></p> <p>Precíz, aprólékos, pontos munkára nevelés</p>	<pre> atl:real; begin clrscr; writeln('Kérem a jegyeket egyesével, ha vége, a nullát üsd le!'); s:=0; db:=1; write('Kérem a(z) ',db,'. jegyet: '); readln(jegy); while (jegy>0) do begin s:=s+jegy; write('Kérem a(z) ',db,'. jegyet: '); readln(jegy); db:=db+1; end; db:=db-1; if db=0 then writeln('0 db jegynek nincs átlaga!') else begin atl:=s/db; writeln('Az átlag: ',atl:2:2); end; readln; end. </pre>		<p>precizitásra való törekvés</p>

SZAKMAI GONDOLATMENET, MUNKAFÁZISOK	TANÁRI, TANULÓI TEVÉKENYSÉGEK, MUNKAFORMÁK, MÓDSZEREK	ÚJ FOGALMAK, ISMERETEK	ESZKÖZÖK ÉS FELTÉTELEK
Házi feladat <i>EHF.pas</i> : Kérjen be egy negatív számot (ellenőrizze), majd írassa ki a szám (-5)-szörösét!	A házi feladat célja az órán tanultak alapos elmélyítése, begyakorlása.		
V. Értékelés	A csoportra vonatkozóan A legügyesebbek egyéni kiemelése		

Óravázlat 6.

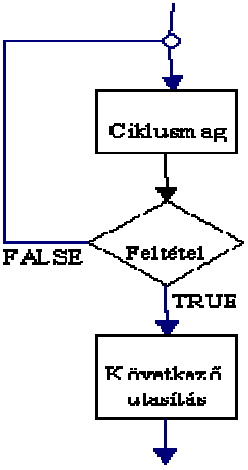
A tanítási óra anyaga:	A hátul tesztelő ciklusok fogalma, működési elve; alkalmazása
A tanítási óra típusa:	Új ismeret-szerző óra
Eszközök:	Vizuál tábla, számítógépek, Turbo Pascal program, Netsupport School rendszer, Molnár Csaba – Sági Gábor: Programozás Turbo Pascal nyelven
A tanulócsoporthoz előzetes ismeretei:	Az elől tesztelő ciklusok fogalma, működési elve; alkalmazása; A számlálós ciklus fogalma, szemantikája, alkalmazása
A tanítási óra céljai:	Az új ismeretek beágyazása a meglévő ismeretrendszerbe Az új ismeretek összefüggéseinek feltárása, gyakorlati jelentőségének megértése Gondolkodási műveletek mozgósítása, a tanulási folyamatok segítése A programozás tantárgy iránti érdeklődés felkeltése, illetve megerősítése Alapvető programozási fogalmak ismerete Az egzakt, pontos feladatmegoldásra való igény fejlesztése

Folyamatterv:

- I. Jelentés, szervezési feladatok
- II. Ismétlés, házi feladat ellenőrzése
- III. Hátul tesztelő ciklusok
- IV. Feladatok ciklusokra
- V. Értékelés

SZAKMAI GONDOLATMENET, MUNKAFÁZISOK	TANÁRI, TANULÓI TEVÉKENYSÉGEK, MUNKAFORMÁK, MÓDSZEREK	ÚJ FOGALMAK, ISMERETEK	ESZKÖZÖK ÉS FELTÉTELEK
I. Jelentés, szervezési feladatok	<p>A hetesek jelentése A csoport üdvözlése A napló beírása A számítógépek bekapcsolása <i>/Fegyelemre, udvariasságra, a szabályok betartására nevelés./</i></p>		napló, számítógépek
<p>II. Ismétlés, házi feladat ellenőrzése</p> <p>A házi feladattal kapcsolatos esetleges problémák megbeszélése</p> <p><i>EHF.pas:</i> Kérjen be egy negatív számot (ellenőrizze), majd írassa ki a szám (-5)-szörösét!</p>	<p>Egy diák már óra előtt felírja a táblára a programot. Megbeszéljük helyességét, s aki tévesztett, kijavítja hibáját.</p> <p>program ehf; uses crt; var a:real;</p> <p>begin clrscr; writeln('Kérem a negatív számokat!'); readln(a); while (a<0) do begin writeln(a*(-5):5:2); writeln('Kérem a negatív számokat!'); readln(a); end; end.</p>		elől tesztelő ciklus megismerése, alkalmazása

SZAKMAI GONDOLATMENET, MUNKAFÁZISOK	TANÁRI, TANULÓI TEVÉKENYSÉGEK, MUNKAFORMÁK, MÓDSZEREK	ÚJ FOGALMAK, ISMERETEK	ESZKÖZÖK ÉS FELTÉTELEK
Az ismétlés egy tanuló felelése folyamán valósul meg	Feleltetés az elmúlt óra anyagából <i>/Egzakt, pontos megfogalmazásra való nevelés; Kifejezőképesség, megértésen alapuló fejlesztés/</i>		
<p>III. Hátralétesztelő ciklusok</p> <p>Az új ismeretek beágyazása a meglévő ismeretrendszerbe</p>	<p>Az új ismeret felkerül a táblára. Az új fogalmakat diktálás során rögzítik a tanulók a füzetükbe.</p> <p><u>Repeat ..until (hátralétesztelő ciklus)</u></p> <p>REPEAT utasítások UNTIL logikai kifejezés</p> <p><i>Másképpen:</i></p> <p>repeat = ismétlés until = addig – amíg</p> <p>Ciklus a <i>ciklusmag</i> utasításai <i>amíg</i> feltétel Ciklus vége</p> <p>repeat utasítás (ciklusmag) until logikai kifejezés</p>	<p>repeat..until „hátralétesztelő”</p>	<p>alapvető programozási fogalmak ismerete</p>

SZAKMAI GONDOLATMENET, MUNKAFÁZISOK	TANÁRI, TANULÓI TEVÉKENYSÉGEK, MUNKAFORMÁK, MÓDSZEREK	ÚJ FOGALMAK, ISMERETEK	ESZKÖZÖK ÉS FELTÉTELEK
<p>Az utasítás folyamatábrája is felkerül a táblára annak érdekében, hogy az iteráció részei közötti összefüggések, valamint szemantikájának megértését segítse. Így az algoritmus ábrázoló eszközeinek alkalmazása révén értelmezik azt.</p> <p>Az új ismeretek összefüggéseinek feltárása</p>	<p>Az utasítás folyamatábrája:</p>  <pre> graph TD Start(()) --> LoopMag[Ciklusmag] LoopMag --> Feltétel{Feltétel} Feltétel -- FALSE --> Start Feltétel -- TRUE --> Utasitas[Következő utasítás] Utasitas --> End(()) </pre> <p>A ciklusmag végrehajtása mindaddig ismétlődik, míg a logikai – kifejezés értéke TRUE nem lesz. Mivel a kifejezés kiértékelése a ciklus végén történik, a ciklusmag legalább egyszer végrehajtódik. Ha a ciklusmagban több utasítás van, akkor sincs szükség begin...end kulcsszavakra, mivel a repeat ..until parancspár is egybefogja a tevékenységsorozatot.</p> <p>Fontos megemlíteni, hogy a while-do és a</p>		<p>Az algoritmus ábrázoló eszközeinek ismerete, folyamatábra értelmezése.</p>

SZAKMAI GONDOLATMENET, MUNKAFÁZISOK	TANÁRI, TANULÓI TEVÉKENYSÉGEK, MUNKAFORMÁK, MÓDSZEREK	ÚJ FOGALMAK, ISMERETEK	ESZKÖZÖK ÉS FELTÉTELEK
<p>IV. Feladatok ciklusokra</p> <p>Az új ismeretek beágyazása a meglévő ismeretrendszerbe</p> <p>1. feladat A múlt órán elől tesztelő ciklussal már megoldottuk a most következő feladatokat, lássuk ezeket hátul tesztelő ciklussal! <i>H1.pas</i>: Kérjünk be egy számot 1 és 10 között, s írassuk ki a képernyő közepére! <i>Ami fontos, hogy mindaddig ismétlődik a ciklusmag utasítása, míg hamis a feltétel, ami az until utasítást követi. Akkor áll le a ciklus, amikor</i></p>	<p>repeat-until utasításpároknál a ciklusmag olyan legyen, hogy hordozza magában a logikai kifejezés teljesülésének lehetőségét, mert ha nem teljesül, akkor végtelen ciklust hozhatunk létre.</p> <p>Frontális munka tanári irányítással A programok felkerülnek a táblára</p> <p><i>/problémaérzékenység fejlesztése/</i></p> <p><i>until (a>=1) and (a<=10)</i></p>		<p>Szövegértés, elvonatkoztatás, probléma meglátása</p>

SZAKMAI GONDOLATMENET, MUNKAFÁZISOK	TANÁRI, TANULÓI TEVÉKENYSÉGEK, MUNKAFORMÁK, MÓDSZEREK	ÚJ FOGALMAK, ISMERETEK	ESZKÖZÖK ÉS FELTÉTELEK
<i>igaz lesz! Ügyeljünk a feltétel meghatározásánál!</i>	<pre> program H1; uses crt; var a:integer; begin clrscr; repeat writeln('Mondj egy számot 1 és 10 között! '); readln(a); until (a>=1) and (a<=10); gotoxy(35,12); writeln('A számod :',a); readln; end. </pre>	kettős feltétel megadása (and)	
<p><u>H2.pas</u>: Előző feladatot dolgozzuk ki részletesebben! Jelezze a program, hogy a beírt szám kisebb vagy nagyobb a kértnél!</p> <p><i>Vizsgáljuk külön-külön a két határértéket!</i></p> <p>Gondolkodási műveletek mozgósítása</p>	<pre> program H2; uses crt; var a:integer; begin clrscr; writeln('Mondj egy számot 1 és 10 között! '); repeat readln(a); if a<1 then write('Ez túl kicsi, mondj másikat: '); if a>10 then write('Ez túl nagy, mondj </pre>		elvonatkoztatás képessége

SZAKMAI GONDOLATMENET, MUNKAFÁZISOK	TANÁRI, TANULÓI TEVÉKENYSÉGEK, MUNKAFORMÁK, MÓDSZEREK	ÚJ FOGALMAK, ISMERETEK	ESZKÖZÖK ÉS FELTÉTELEK
	<pre> másikat: '); until (a>=1) and (a<=10); clrscr; gotoxy(35,12); writeln('A számod :',a); readln; end. </pre> <p style="text-align: center;"><i>/rugalmasság, ötletgazdagság/</i></p>		
<p><i>H3.pas:</i> A program addig kér be számokat, míg 0-t nem adunk meg, s a bekért számoknak kiszámolja az összegét!</p> <p><i>Mit fog tartalmazni a ciklusmag? A feltétel?</i></p> <p>Az új ismeretre vonatkozó alapeladatok készségi szintre történő begyakorlása</p>	<p><i>beolvasást és összegzést a szám=0</i></p> <pre> program H3; uses crt; var szam,osszeg:integer; begin clrscr; szam:=0; osszeg:=0; repeat write('Kérem adja meg a következő számot:'); readln(szam); </pre>		<p>matematikai alpműveletek, alapvető programozási műveletek alkalmazása</p>

SZAKMAI GONDOLATMENET, MUNKAFÁZISOK	TANÁRI, TANULÓI TEVÉKENYSÉGEK, MUNKAFORMÁK, MÓDSZEREK	ÚJ FOGALMAK, ISMERETEK	ESZKÖZÖK ÉS FELTÉTELEK
<p><i>H4.pas</i>: Kérjünk be két tetszőleges számot, határozzuk meg a különbségüket, majd a végén kérdezzük meg, hogy legyen-e vége!</p> <p><i>Addig kér be újat, míg nem i-t vagy I-t ütünk le!</i></p>	<pre> osszeg:=osszeg+szam; until szam=0; write('A számok összege: ',osszeg); readln; end. </pre> <p>Új, ismeretlen probléma, együttes munka.</p> <p><i>/ötletgazdagság, együttműködési képesség fejlesztése/</i></p> <pre> program H4; uses crt; var a,b,k:integer; c:char; begin clrscr; repeat write('Kérem az első számot:'); readln(a); write('Kérem a második számot:'); readln(b); writeln; k:=a-b; writeln('A két szám különbsége:',k); </pre>	<p>readkey- várakoztatás billentyűzet leütéséig</p>	<p>algoritmikus, problémamegoldó gondolkodás</p> <p>matematikai alapműveletek, alapvető programozási műveletek alkalmazása</p>

SZAKMAI GONDOLATMENET, MUNKAFÁZISOK	TANÁRI, TANULÓI TEVÉKENYSÉGEK, MUNKAFORMÁK, MÓDSZEREK	ÚJ FOGALMAK, ISMERETEK	ESZKÖZÖK ÉS FELTÉTELEK
	<pre>write('Akarsz még egyet számolni? (i/n)'); c:=readkey; until(c<>'i') and (c<>'T'); readln; end.</pre>		
<p>Házi feladat <i>HHF.pas</i>: Kérjen be egy 100 és 200 közötti számot, tesztelje, és irassa ki a szám négyzetét, majd a végén kérdezzen rá, folytatja-e!</p>	<p>A házi feladat célja az órán tanultak alapos elmélyítése, begyakorlása.</p>		
<p>IV. Értékelés</p>	<p>A csoportra vonatkozóan A legügyesebbek egyéni kiemelése A csoport egészének értékelése. <i>Az elismerés mellett jelezzük, ha nem volt mindenkinek megfelelő a munkája. A tanulók érezzék, hogy mindenkire figyelünk.</i></p>		

Óravázlat 7.

A tanítási óra anyaga:

Elöl-és hátul tesztelő ciklusutasítások alkalmazásának gyakorlása

A tanítási óra típusa:

Gyakorló óra

Eszközök:

Vizuál tábla, számítógépek, Turbo Pascal program, Netsupport School rendszer,
Molnár Csaba – Sági Gábor: Programozás Turbo Pascal nyelven

A tanulócsoporthoz előzetes ismeretei:

A hátul tesztelő, az elöl tesztelő, a számlálós ciklus fogalma, szemantikája, alkalmazása

A tanítási óra céljai:

A korábban megszerzett ismeretek megerősítése

A megtanult fogalmak és összefüggések alkalmazás szintű elsajátítása

Gondolkodási műveletek mozgósítása, a tanulási folyamatok segítése

A tantárgyak közötti kapcsolatok szemléltetése

A programozás tantárgy iránti érdeklődés felkeltése, illetve megerősítése

Az alapeladatok megoldási algoritmusának készségszintre fejlesztése

Az egzakt, pontos feladatmegoldásra való igény fejlesztése

Folyamatterv:

- I. Jelentés, szervezési feladatok
- II. Ismétlés, házi feladat ellenőrzése
- III. Gyakorlás
- IV. Értékelés

SZAKMAI GONDOLATMENET, MUNKAFÁZISOK	TANÁRI, TANULÓI TEVÉKENYSÉGEK, MUNKAFORMÁK, MÓDSZEREK	ÚJ FOGALMAK, ISMERETEK	ESZKÖZÖK ÉS FELTÉTELEK
I. Jelentés, szervezési feladatok	<p>A hetesek jelentése A csoport üdvözlése A napló beírása A számítógépek bekapcsolása <i>/Fegyelemre, udvariasságra, a szabályok betartására nevelés./</i></p>		napló, számítógépek
<p>II. Ismétlés, házi feladat ellenőrzése</p> <p>A házi feladattal kapcsolatos esetleges problémák megbeszélése</p> <p><i>HHF.pas</i>: Kérjen be egy 100 és 200 közötti számot, tesztelje, és írassa ki a szám négyzetét, majd a végén kérdezzen rá, folytatja-e!</p>	<p>Az elől és a hátul tesztelő ciklusról tanultak összegzése, összevetése tanári irányítással.</p> <p>Egy diák már óra előtt felírja a táblára a programot. Megbeszéljük helyes-e, s aki tévesztett, kijavítja hibáját.</p> <pre> program hhf; uses crt; var a,k:real; c:char; begin clrscr; repeat writeln('Kérem a számot:'); readln(a); if (a>=100) and (a<=200) then begin k:=sqr(a); </pre>		<p>számok négyzete szorzással vagy sqr függvénnyel</p> <p>readkey használata</p> <p>folytatás lehetősége</p>

SZAKMAI GONDOLATMENET, MUNKAFÁZISOK	TANÁRI, TANULÓI TEVÉKENYSÉGEK, MUNKAFORMÁK, MÓDSZEREK	ÚJ FOGALMAK, ISMERETEK	ESZKÖZÖK ÉS FELTÉTELEK
Az ismétlés egy tanuló felelése folyamán valósul meg	<pre>writeln('A szám négyzete:',k:10:2); end; writeln('Akarsz még egyet számolni? (i/n)'); c:=readkey; until(c<>'i') and (c<>'T'); readln; end.</pre> <p>Feleltetés az elmúlt óra anyagából (<i>Egzakt, pontos megfogalmazásra való nevelés; Megértésen alapuló fejlesztés</i>)</p>		
<p>III. Gyakorlás (iterációk alkalmazása)</p> <p>Kezdeményezőképeség és a vállalkozói kompetencia fejlesztése</p> <p>Mi a véleményetek az alábbi utasításról? Mit csinál? Repeat</p> <p>Until (a<=1) and (a>=10);</p> <p><i>Akkor mi történhet, ha futtatjuk? Az ilyen ciklusokat nevezzük végtelen ciklusoknak. Végtelen ciklusnak nevezzük az olyan</i></p>	<p>Netsupport segítségével kivetítem a tanulók képernyőjére a feladatot. A diákok kapnak egy kis gondolkodási időt.</p> <p><i>Olyan a feltétel, amely egyetlen értékre sem teljesülhet soha. Sosem áll le.</i></p>	végtelen ciklus	<p>Netsupport rendszer</p> <p>lényeglátás, problémamegoldó gondolkodás</p>

SZAKMAI GONDOLATMENET, MUNKAFÁZISOK	TANÁRI, TANULÓI TEVÉKENYSÉGEK, MUNKAFORMÁK, MÓDSZEREK	ÚJ FOGALMAK, ISMERETEK	ESZKÖZÖK ÉS FELTÉTELEK
<p>ciklusokat, amelyben a feltételt-persze hibásan- úgy fogalmazzuk meg, hogy az soha ne tegye lehetővé a ciklusból való kilépést. Jól működő programokban a végtelen ciklusoknak természetesen nincs helyük, de elég gyakori a hiba, főleg kezdő programozóknál, ezért érdemes külön foglalkozni velük. <i>Hogyan lesz helyesen?</i></p> <p>S mi a vélemény az alábbi ciklusról?</p> <pre>Readln(a); Repeat Until (a>=1) and (a<=10);</pre> <p><i>Helyesen?</i></p> <p>Az alábbi utasítás?</p> <pre>s:=0; while i<=10 do</pre>	<p><i>Helyes:</i> Repeat Readln(a); Until (a>=1) and (a<=10);</p> <p><i>A ciklus elé tették az adatbekérést, és a ciklus üres.</i> <i>Helyes:</i> Repeat Readln(a); Until (a>=1) and (a<=10);</p>		<p>a programozás szakkifejezéseinek helyes használata</p> <p>lényeglátás, problémamegoldó gondolkodás</p>

SZAKMAI GONDOLATMENET, MUNKAFÁZISOK	TANÁRI, TANULÓI TEVÉKENYSÉGEK, MUNKAFORMÁK, MÓDSZEREK	ÚJ FOGALMAK, ISMERETEK	ESZKÖZÖK ÉS FELTÉTELEK
<pre>s:=s+i;</pre> <p>end;</p> <p>Mi a véleményetek az alábbi utasításról?</p> <pre>i:=0; s:=0; Repeat</pre> <p>Until i=10;</p> <p><i>S helyesen?</i></p>	<p><i>Az i, a ciklusváltozó kezdőérték-adása a cikluson belülrre került.</i></p> <p><i>Helyes:</i></p> <pre>i:=1; s:=0; while i<=10 do begin s:=s+i; i:=i+1; end;</pre> <p><i>A ciklusváltozó értéke nincs növelve a ciklus belsejében.</i></p> <p><i>Helyes:</i></p> <pre>i:=0; s:=0; Repeat i:=i+1; s:=s+i; Until i=10;</pre>		<p>a programozás szakkifejezéseinek helyes használata</p> <p>lényeglátás, problémamegoldó gondolkodás</p>

SZAKMAI GONDOLATMENET, MUNKAFÁZISOK	TANÁRI, TANULÓI TEVÉKENYSÉGEK, MUNKAFORMÁK, MÓDSZEREK	ÚJ FOGALMAK, ISMERETEK	ESZKÖZÖK ÉS FELTÉTELEK
<p><i>Az előbb látott hibákat jól jegyezzétek meg, gyakori tévesztési esetek!</i></p> <p>1. feladat <i>Találkozhatunk olyan feladatokkal, melyek akár mindhárom ciklusfajttával megoldhatóak, s rajtuk áll, melyiket alkalmazzuk a gyakorlatban.</i></p> <p>kezdeményezőképeség, vállalkozói kompetencia fejlesztése</p> <p><i>Példa.pas: Írassuk ki a számokat a képernyőre 20-tól 40-ig egymás alá!</i></p> <p><i>Kezdjük while ciklussal:</i></p> <p><i>For ciklussal:</i></p>	<p>Önálló munka, ellenőrzés vizuál táblán vagy Netsuppert segítségével. Helyes megoldás értékelése plusszal.</p> <pre> program elolt; uses crt; var i:integer; begin clrscr; i:=19; while i<40 do begin i:=i+1; writeln(i); end; readln; end. program szaml; uses crt; var i:integer; </pre>		<p>elvonatkoztatás, tanult ismeretek összevetése</p>

SZAKMAI GONDOLATMENET, MUNKAFÁZISOK	TANÁRI, TANULÓI TEVÉKENYSÉGEK, MUNKAFORMÁK, MÓDSZEREK	ÚJ FOGALMAK, ISMERETEK	ESZKÖZÖK ÉS FELTÉTELEK
<p><i>Repeat..until ciklussal:</i></p> <p>Hagyom, hogy önállóan próbáljanak rájönni a megoldásra, ezzel gondolkodásra készítem a diákokat.</p> <p>2.feladat <i>PL_E.pas:</i> A program addig kér be számokat, míg azok összege nagyobb nem lesz 1000-nél!</p> <p><i>Lássuk elől tesztelő ciklussal!</i></p>	<pre>begin clrscr; for i:=20 to 40 do writeln(i); readln; end. program hatult; uses crt; var i:integer; begin clrscr; i:=20; repeat writeln(i); i:=i+1; until i>40; readln; end.</pre> <p><i>/összefüggések feltárása, megoldás megtervezésének képessége, ill. ezek fejlesztése/</i></p>		<p>algoritmikus gondolkodás</p> <p>következtetési képesség</p>

SZAKMAI GONDOLATMENET, MUNKAFÁZISOK	TANÁRI, TANULÓI TEVÉKENYSÉGEK, MUNKAFORMÁK, MÓDSZEREK	ÚJ FOGALMAK, ISMERETEK	ESZKÖZÖK ÉS FELTÉTELEK
<p>matematikai kompetencia</p> <p>3.feladat <i>PL_H.pas</i>: Írjunk programot, mely egy tanuló átlagát számítja ki! Az érdemjegyeket egyesével kérjük be, a bekérést a 0 zárja.</p> <p><i>Korábban elől tesztelő ciklussal már megoldottátok ezt a feladatot, most lássuk hátul tesztelőssel!</i></p>	<pre> program PL_E; uses crt; var szam,osszeg:integer; begin clrscr; osszeg:=0; szam:=0; while osszeg<=1000 do begin write('Kérem adjon meg egy számot:'); readln(szam); osszeg:=osszeg+szam; end; write('A számok összege:',osszeg); readln; end. </pre> <pre> program Pl_H; uses crt; </pre>		<p>problémamegoldó gondolkodás</p> <p>átlagszámítás</p> <p>elvonatkoztatás képessége</p>

SZAKMAI GONDOLATMENET, MUNKAFÁZISOK	TANÁRI, TANULÓI TEVÉKENYSÉGEK, MUNKAFORMÁK, MÓDSZEREK	ÚJ FOGALMAK, ISMERETEK	ESZKÖZÖK ÉS FELTÉTELEK
vállalkozói kompetencia fejlesztése	<pre> var jegy,db,s:integer; atl:real; begin clrscr; writeln('Kérem a jegyeket egyesével, ha vége, a nullát üsd le!'); s:=0; db:=0; repeat db:=db+1; repeat write('Kérem a(z) ',db,'. jegyet: '); readln(jegy); until (jegy>=0) and (jegy<=5); s:=s+jegy; until jegy=0; db:=db-1; if db=0 then writeln('0 db jegynek nincs átlaga!') else begin atl:=s/db; writeln('Az átlag: ',atl:2:2); end; readln; end. </pre>		<p>kettős feltétel megadása</p> <p>elágazások alkalmazása</p> <p>valós szám kiírása</p>

SZAKMAI GONDOLATMENET, MUNKAFÁZISOK	TANÁRI, TANULÓI TEVÉKENYSÉGEK, MUNKAFORMÁK, MÓDSZEREK	ÚJ FOGALMAK, ISMERETEK	ESZKÖZÖK ÉS FELTÉTELEK
<p>4.feladat <i>Pelda2.pas</i>: Készítsünk olyan programot, amely egy adott tartományon belül, adott lépésközzel kilistázza a Fahrenheit fokokban adott hőmérséklet Celsius fokokban számolt értékét! $celsius=5*(fahr-32) \text{ div } 9$ <i>Mit kell megadnunk a deklaráció után?</i> <i>Annak kezdő- és végértékét, valamint?</i></p> <p><i>Először elől tesztelő ciklussal oldjuk meg!</i></p> <p>Ismeretek kiegészítése, bővítése</p> <p>Programozás és a való világ kapcsolata</p>	<p>Frontális munka tanári irányítással A program felkerül a táblára</p> <p><i>A tartományt. A lépésközt.</i></p> <p>program elolt; uses crt; var fahr,cels,e,v,n:integer;</p> <pre>begin clrscr; e:=0; v:=300; n:=20; fahr:=e; writeln; while (fahr<=v) do begin</pre>	<p>Fahrenheit fok és a Celsius fok viszonya</p>	<p>logikus gondolkodás</p> <p>egészrész osztás- div</p>

SZAKMAI GONDOLATMENET, MUNKAFÁZISOK	TANÁRI, TANULÓI TEVÉKENYSÉGEK, MUNKAFORMÁK, MÓDSZEREK	ÚJ FOGALMAK, ISMERETEK	ESZKÖZÖK ÉS FELTÉTELEK
	<pre>cels:=5*(fahr-32) div 9; writeln(fahr,' ',cels); fahr:=fahr+n; end; readln; end.</pre> <p>Szükség esetén tesztelés.</p>		
<p>Házi feladat Az előző feladat megoldása számlálás és hátul tesztelés ciklussal.</p>	<p>A házi feladat célja az órán tanultak alapos elmélyítése, begyakorlása.</p>		
<p>IV. Értékelés</p>	<p>A csoportra vonatkozóan A legügyesebbek egyéni kiemelése A szerzett plusszok beírása a kisnaplóba, ezúton a legügyesebbek egyéni kiemelése A magatartás, munkamorál minősítése</p>		

Óravázlat 8.

A tanítási óra anyaga:	Ciklusutasítások alkalmazásának gyakorlása, összefoglalás
A tanítási óra típusa:	Gyakorló óra
Eszközök:	Vizuál tábla, számítógépek, Turbo Pascal program, Netsupport School rendszer, Molnár Csaba – Sági Gábor: Programozás Turbo Pascal nyelven
A tanulócsoporthoz előzetes ismeretei:	A hátul tesztelő, az elől tesztelő, a számlálós ciklus fogalma, szemantikája, alkalmazása
A tanítási óra céljai:	A korábban megszerzett ismeretek megerősítése Önálló munkára, felelősségvállalásra, nyilvános szereplésre nevelés A megtanult fogalmak és összefüggések alkalmazás szintű elsajátítása Gondolkodási műveletek mozgósítása, a tanulási folyamatok segítése A tantárgyak közötti kapcsolatok szemléltetése A programozás tantárgy iránti érdeklődés felkeltése, illetve megerősítése

Folyamatterv:

- I. Jelentés, szervezési feladatok
- II. Ismétlés, házi feladat ellenőrzése
- III. Gyakorlás
- IV. Értékelés

SZAKMAI GONDOLATMENET, MUNKAFÁZISOK	TANÁRI, TANULÓI TEVÉKENYSÉGEK, MUNKAFORMÁK, MÓDSZEREK	ÚJ FOGALMAK, ISMERETEK	ESZKÖZÖK ÉS FELTÉTELEK
I. Jelentés, szervezési feladatok	A hetesek jelentése A csoport üdvözlése A napló beírása A számítógépek bekapcsolása <i>/Fegyelemre, udvariasságra, a szabályok betartására nevelés./</i>		napló, számítógépek
II. Ismétlés, házi feladat ellenőrzése A házi feladattal kapcsolatos esetleges problémák megbeszélése <i>Pelda2.pas</i> : Készítsünk olyan programot, amely egy adott tartományon belül, adott lépésközzel kilistázza a Fahrenheit fokokban adott hőmérséklet Celsius fokokban számolt értékét!	Egy diák már óra előtt felírja a táblára a programot. Ellenőrzés, javítás. program szamlal; uses crt; var fahr,cels,v,n,i,k:integer; begin clrscr; v:=300; n:=20; k:=v div n; fahr:=0; writeln; for i:=0 to k do begin cels:=5*(fahr-32) div 9; writeln(fahr,' ',cels); fahr:=fahr+n; end; readln; end.		Fahrenheit fok és a Celsius fok viszonya

SZAKMAI GONDOLATMENET, MUNKAFÁZISOK	TANÁRI, TANULÓI TEVÉKENYSÉGEK, MUNKAFORMÁK, MÓDSZEREK	ÚJ FOGALMAK, ISMERETEK	ESZKÖZÖK ÉS FELTÉTELEK
	<pre> program hatult; uses crt; var fahr,cels,e,v,n:integer; begin clrscr; e:=0; v:=300; n:=20; fahr:=e; writeln; repeat cels:=5*(fahr-32) div 9; writeln(fahr,' ',cels); fahr:=fahr+n; until (fahr>v); readln; end. </pre>		<p>tanult ismeretek felhasználása, alkalmazása</p>
<p>III.Gyakorlás 1. feladat <i>Gyak1.pas:</i> Írassuk ki 200-ig az 5-tel osztható számok közül azokat, melyek 25-tel nem oszthatók!</p> <p><i>Oldjátok meg hátul tesztelő ciklussal!</i></p>	<p>Önálló munka, ellenőrzés vizuál tábla segítségével Helyes megoldás értékelése plusszal.</p> <pre> program gyak1; uses crt; var i:integer; </pre>		<p>elágazás ciklusban</p>

SZAKMAI GONDOLATMENET, MUNKAFÁZISOK	TANÁRI, TANULÓI TEVÉKENYSÉGEK, MUNKAFORMÁK, MÓDSZEREK	ÚJ FOGALMAK, ISMERETEK	ESZKÖZÖK ÉS FELTÉTELEK
<p>Kezdeményezőképeség és a vállalkozói kompetencia fejlesztése</p> <p>Gondolkodási műveletek mozgósítása, a tanulási folyamatok segítése</p>	<pre>begin clrscr; i:=1; repeat begin if (i mod 5=0) and (i mod 25>0) then writeln(",i); i:=i+1; end; until (i=200); readln; end.</pre>		<p>kettős feltétel megadása</p> <p>önállóság, kreativitás</p>
<p>2. feladat</p> <p><i>Gyak2.pas:</i> Egy beolvasott számról döntse el a program, hogy -30 és 40 között van-e, s ha nem, kérjen újabb számot!</p> <p><i>Melyik ciklussal érdemes?</i></p> <p>Következtetési képesség fejlesztése</p> <p>Megoldás megtervezésének képessége</p>	<p><i>/lényeglátás, döntéshozó képesség fejlesztése/</i></p> <pre>program gyak2; uses crt; var a:real; i:integer; begin clrscr; repeat begin writeln('Kérem a számot!'); readln(a); end; until (a>=-30) and (a<=40); writeln('köztük van!'); readln; end.</pre>		<p>kettős feltétel megadása</p> <p>önállóság, kreativitás</p>

SZAKMAI GONDOLATMENET, MUNKAFÁZISOK	TANÁRI, TANULÓI TEVÉKENYSÉGEK, MUNKAFORMÁK, MÓDSZEREK	ÚJ FOGALMAK, ISMERETEK	ESZKÖZÖK ÉS FELTÉTELEK
<p>3. feladat <i>Gyak3.pas</i>: Számítsa ki a program 10 véletlen szám szorzatát és átlagát, melyek 1 és 15 közöttiek!</p> <p><i>Mivel a számok szorzatát is kéri a feladat, gondoljuk át, milyen adattípussal deklaráljuk! Az integer-nél tágabb tartományt kell választanunk.</i></p> <p>Új ismeret beágyazása a már meglévő ismeretkörbe</p> <p>Gondolkodási műveletek mozgósítása, a tanulási folyamatok segítése</p>	<p>Frontális munka tanári irányítással. A program a táblára kerül.</p> <p>Véletlenszám-generálás random() függvény segítségével történik. Számgenerálás tartománya 0 és 1 közötti (1 már nem tartozik bele), így az 1 és 15 közötti tartományt a random(15)+1 segítségével adható meg. Növelve az alsó határt, s belevéve a felsőt: $1 \leq x < 16$</p> <p><i>longint</i></p> <pre> program gyak3; uses crt; const n=10; var a,i,s:integer; sz:longint; atl:real; begin clrscr; randomize; sz:=1; s:=0; atl:=0; for i:=1 to n do </pre>	<p>véletlenszám-generálás, randomize, random()</p>	<p>longint adattípus</p>

SZAKMAI GONDOLATMENET, MUNKAFÁZISOK	TANÁRI, TANULÓI TEVÉKENYSÉGEK, MUNKAFORMÁK, MÓDSZEREK	ÚJ FOGALMAK, ISMERETEK	ESZKÖZÖK ÉS FELTÉTELEK
Összefüggések megláttatása	<pre> begin a:=random(15)+1; writeln(a); s:=s+a; sz:=sz*a; end; atl:=s/n; writeln('szorzatuk: ',sz); writeln('atlaguk: ',atl:5:2); readln; end. </pre>		

SZAKMAI GONDOLATMENET, MUNKAFÁZISOK	TANÁRI, TANULÓI TEVÉKENYSÉGEK, MUNKAFORMÁK, MÓDSZEREK	ÚJ FOGALMAK, ISMERETEK	ESZKÖZÖK ÉS FELTÉTELEK
<p>4. feladat <i>Gyak4.pas</i>: Írjuk ki az N-nél nem nagyobb páratlan számok összegét!</p> <p>A vállalkozói kompetencia fejlesztése</p>	<p>Önálló munka, ellenőrzés vizuál tábla segítségével Helyes megoldás értékelése plusszal.</p> <pre> program gyak4; uses crt; const n=100; var a,i,s:integer; begin clrscr; s:=0; i:=0; while (i<=n) do begin if (i mod 2>0) then s:=s+i; i:=i+1; end; writeln('osszeguk: ',s); readln; end. </pre>		<p>maradékos osztás- mod</p>

SZAKMAI GONDOLATMENET, MUNKAFÁZISOK	TANÁRI, TANULÓI TEVÉKENYSÉGEK, MUNKAFORMÁK, MÓDSZEREK	ÚJ FOGALMAK, ISMERETEK	ESZKÖZÖK ÉS FELTÉTELEK
<p>5. feladat <i>Gyak5.pas</i>: Egy 5 elemű vektor elemeinek szorzatát határozzuk meg!</p> <p>Önálló munkára, felelősségvállalásra, nyilvános szereplésre nevelés</p> <p>Gondolkodási műveletek mozgósítása, a tanulási folyamatok segítése</p>	<pre> program gyak5; uses crt; const n=5; var a:array[1..n] of integer; i:integer; sz:longint; begin clrscr; sz:=1; for i:=1 to n do begin readln(a[i]); sz:=sz*a[i]; end; for i:=1 to n do write(' ',a[i]); writeln(' szorzatuk: ',sz); readln; end. </pre>		<p>longint adattípus</p>

SZAKMAI GONDOLATMENET, MUNKAFÁZISOK	TANÁRI, TANULÓI TEVÉKENYSÉGEK, MUNKAFORMÁK, MÓDSZEREK	ÚJ FOGALMAK, ISMERETEK	ESZKÖZÖK ÉS FELTÉTELEK
<p>6. feladat <u>Gyak6.pas</u>: Egy 5 elemű vektor szomszédos elemeinek különbségeit határozzuk meg!</p> <p>A kreatív, problémamegoldó, algoritmikus gondolkodás fejlesztése</p>	<p>Igény szerinti tanári segítségnyújtás</p> <pre> program gyak6; uses crt; const n=5; var a:array[1..n] of integer; i,k:integer; begin clrscr; k:=0; for i:=1 to n do begin readln(a[i]); end; for i:=1 to n-1 do begin k:=a[i]-a[i+1]; writeln(' ',k); end; readln; end. </pre>		<p>önállóság, kreativitás</p>
<p>7. feladat (Szorgalmi feladat) <u>Gyak7.pas</u>: Egy 2x2-es mátrix elemeinek összegét és soronkénti összeget jelenítse meg a program!</p>	<p>Igény szerinti tanári segítségnyújtás</p> <p><i>/elemzőképesség fejlesztése/</i></p> <pre> program gyak7; uses crt; const n=2; </pre>		<p>logikus kreatív gondolkodás</p>

SZAKMAI GONDOLATMENET, MUNKAFÁZISOK	TANÁRI, TANULÓI TEVÉKENYSÉGEK, MUNKAFORMÁK, MÓDSZEREK	ÚJ FOGALMAK, ISMERETEK	ESZKÖZÖK ÉS FELTÉTELEK
A vállalkozói kompetencia fejlesztése	<pre> var a:array[1..n,1..n] of integer; i,j,s,ss:integer; begin clrscr; s:=0; ss:=0; for i:=1 to n do begin for j:=1 to n do begin readln(a[i][j]); ss:=ss+a[i][j]; s:=s+a[i][j]; end; writeln('sorosszeg:',ss); ss:=0; end; writeln('teljes osszeg:',s); readln; end. </pre>		összefüggések meglátása, problémafeltárás
Házi feladat <i>GYAKHF.pas</i> : Írassuk ki az első 10, 3-mal osztható szám felét!	A házi feladat célja az órán tanultak alapos elmélyítése, begyakorlása.		

SZAKMAI GONDOLATMENET, MUNKAFÁZISOK	TANÁRI, TANULÓI TEVÉKENYSÉGEK, MUNKAFORMÁK, MÓDSZEREK	ÚJ FOGALMAK, ISMERETEK	ESZKÖZÖK ÉS FELTÉTELEK
IV. Értékelés	A csoportra vonatkozóan A szerzett plusszok beírása a kisnaplóba, ezúton a legügyesebbek egyéni kiemelése A magatartás, munkamorál minősítése		

Az utolsó gyakorló, összefoglaló órán megmutatkozik, hogy van-e olyan terület, melyen hiányosságok észlelhetők és szükség van ezek pótlására. Amennyiben igen, a szabadon felhasználható órakeret terhére ez megvalósítható, viszont ha mindenki számára begyakoroltak az ismeretek, alkalmazások, a 9. órán következhet a dolgozatírás.

Összegzés

Így a dolgozat utolsó soraiban ismét hangsúlyozom, hogy az itt leírtak valójában a ciklusok alapjainak tanítását fogták át. Mivel a programozás fő elemei az utasítások, így a diákok tanulmányaik során számos változatos, összetett feladatot oldanak majd meg az iterációk alkalmazásával. Ily módon például a sztringek, függvények, eljárások, állományok, stb. kezelése, programozása révén.

Az óravázlatok kidolgozása folytán igyekeztem szem előtt tartani, hogy a tantervben, s a tanmenetben megfogalmazott célok, feladatok megvalósítására lehetőséget találjak, és annak lépéseit követve eljuthassanak a tanulók olyan fejlettségi szintre, hogy a meghatározott követelményeknek eleget tudjanak tenni. A napjainkban teret hódító kompetencia alapú oktatás kívánalmait szintén szem előtt tartottam. Nagyon fontos mindaz, amit ezen oktatás mivolta közvetít: az ismeretekbe ágyazott képességfejlesztés megvalósítását, az egységre törekvést, az összefüggések megláttatását a tanult ismeretek között. Mindennapi munkánk során, sosem szabad elfelejtenünk, hogy a diákot, a gyermeket egészében oktatjuk, tanítjuk, fejlesztjük, neveljük. Nem csupán információt, ismeretet közlünk vele, hanem egész személyiségére kiható, nagy felelősséggel járó munkát végzünk. Tudásvágyát kielégítve egyben megnyitjuk számára a világot. Amellett, hogy megtanítjuk tanulni, az elsajátított ismeretek közötti összefüggéseket is meg kell láttatnunk velük, s erre minden lehetséges alkalmat meg kell ragadnunk. Nekünk, pedagógusoknak óriási lehetőség, s egyben óriási felelősség van a kezünkben: kíváncsi, érdeklődő diákokból tudatos, tenni vágyó felnőtteket nevelni, függetlenül attól, mely tantárgy keretei között tesszük is, de fel kell vállalnunk, minden esetben meg kell ragadnunk a kínálkozó lehetőséget.

Irodalomjegyzék

1. Molnár Csaba – Sági Gábor: Programozás Turbo Pascal nyelven
BBS-E Bt. 1999, 2001.
2. Programozási feladatok I. , Kossuth Kiadó 1997
A kötetet összeállította: Zsakó László
3. Benkő Tiborné - Benkő László - Tóth Bertalan - Varga Balázs: Programozzunk
Turbo Pascal Nyelven!
ComputerBooks Budapest, 2004
4. <http://fppti.hu/szakteruletek/informatika/tanulmanyok/cikkek/helyitantervekrol.pdf>
5. www.zipernowsky.hu/~naszlaci/kov/tanmenetek
6. Programozás1. Jegyzet dr. Juhász István előadása alapján
Készítette: Csordás Annamária