

# OOP tanítása vizualizációs eszközök segítségével

LOGISZTIKA – INFORMATIKA – MENEDZSMENT

volume 4 • number 1 • 2019 január • pp: 31-42

DOI: 10.29177/LIM.2019.1.31

## Összefoglaló

A programozás oktatása nehéz feladat, főleg a lecsökkent óraszámok miatt, részben a diákok felkészültsége miatt. Egy a gyakorlatban felvetődött pedagógiai problémára szerettem volna a cikkben megoldási módszert adni. Erre a vizualizáción és gamifikáción alapuló megoldást találtam ki és egy egyszerű alkalmazás segítségével igazoltam, hogy az oktatás jelentősen hatékonyabbá válik egy ilyen eszköz segítségével. A tanulók a módszer segítségével jobban elmélyítették OOP gyakorlati tudásukat. A játékos elemek használatával elértem, hogy a tanulók úgy tekintsenek az oktató - vizualizáló programra, mint egy egyszerű kis játékra, s közben elsajátították a tananyagot is.

A program csak a főbb elemeket vizsgálta. A programot szeretném továbbfejleszteni, egyrészt webes megoldást létrehozni, valamint a vizualizációt sokkal izgalmasabban megoldani és több gamifikációs elemet felhasználni. Emellett szeretném jobban megmutatni az OOP együttműködési lehetőségeit, a különböző láthatósági elemeket bemutatni, valamint a virtuális és az osztály attribútumokat, ill. metódusokat vizualizálni.

## Absztrakt

A cikk a felvidéki magyar középiskolák OOP tanításával foglalkozik. Két felmérés eredményeit foglalja össze. Az első felmérés segítségével felmértem a középiskolás tanulók általános informatika tudásszintjét, valamint az informatikai ismereteit. A második felmérésben a tanulók objektum-orientált programozás ismereteire fókuszáltam. Két oktatási módszer hatékonyságát vizsgáltam. Összehasonlítottam a jellegzetes előadás módszer és az általam kifejlesztett vizualizációs alkalmazás használatának eredményességét az oktatási folyamatban. A kifejlesztett alkalmazás és a bevezetett módszer segíti a programozás gyakorlati oktatását és növeli a tanulók programozási készségeit.

## Abstract

The article deals with OOP teaching in Hungarian High Schools in Slovakia. It contains the results of two surveys. With the help of the first survey I observed the basic knowledge surrounding Informatics. In the second survey I focused the object-oriented programming primarily. I observed two teaching methods effectiveness. I compared the effectiveness of using typical presentation method and the use of the visualization application developed by me in the educational process. The developed application and introduced method helps the teaching of practical programming and increases the readiness of the students programming.

## 1. Bevezetés

A mai operációs rendszerek és fejlesztőkörnyezetek az objektum orientált programozás (OOP) elvét alkalmazzák és a mai fejlesztések szinte elképzelhetetlenek OOP használata nélkül. A gyakorló programozók, programfejlesztők számára az OOP és a hozzákapcsolódó rendszerfejlesztési módszerek ismerete elengedhetetlen. Ebből következik, hogy a számítástechnika és különösen a programozás oktatása az OOP-n kellene, hogy alapuljon. Az oktatási tapasztalatok alapján ismert, hogy a hagyományos szekvenciális programozás oktatása is nagy nehézségekbe ütközik, ugyanakkor a valósághoz közelebb álló OOP oktatása még nehezebb. Ez főleg abból ered, hogy további elméleti, modellezési ismeretekre van szükség, ugyanakkor a hagyományos szekvenciális programozási elveket is meg kell tanítani. Informatika tanárként sokszor felmerülnek olyan kérdések, mint például milyen szinten is van a szlovákiai informatika, illetve a programozás oktatása, milyen alapokkal érkeznek a tanulók az általános iskolából az érettségivel végződő szakközépiskolákba, gimnáziumokba, hogy értékelik maguk a tanulók általános informatikatudásukat, milyen számítógép kezelői ismeretekkel fejezik be középiskolai tanulmányaikat, mire támaszkodhatnak egyetemi tanulmányaik során, valamint munkahelyeiken, és még sok más. Ezekre a kérdésekre nem egyszerű a válasz. A programozás és azon belül az OOP jelenlegi tanítása nagy nehézségekbe ütközik és ennek okait fel kell tárni. Ezek alapján fogalmazódott meg a doktori értekezésem témája, hogy mérjük fel Szlovákiában az általános középiskolai informatikaoktatás aktuális helyzetét. A tanulók milyen informatika ismeretekkel rendelkeznek. A felmérés eredményei alapján, majd megtudhatjuk, milyen oktatási módszereket használjunk a jobb eredmények elérésére. A javasolt oktatási módszerek alkalmazásával kurzusokat készítettem az OOP oktatására, valamint irodalomkutatást végeztem az OOP tanításával és vizualizációjával kapcsolatosan. Az oktatás során felhasználtam egy általam kifejlesztett vizualizációs alkalmazást, melynek hatékonyságát szerettem volna vizsgálni. A kurzusok végén egy újabb felmérés segítségével bizonyítottam, hogy a tanulók objektum-orientált programozás fogalmainak megértése, elsajátítása és a programozás oktatás hatékonysága a kifejlesztett vizualizációs alkalmazással jelentősen növelhető. Továbbá a tanulók feladat és megoldás modellezési készsége javítható az új vizualizációs és gamifikációs módszer bevezetésével, a kifejlesztett alkalmazás és a bevezetett módszer segíti a programozás gyakorlati oktatását és a tanulók programozási készségeinek növelését.

## 2. Irodalomkutatás

A szakirodalom feldolgozását a szisztematikus irodalomkutatás módszerével végeztem el [1]. Kutatásom során első lépésként átnéztem több nemzetközi és hazai tudományos és egyetemi folyóiratokat, melyek az objektum-orientált programozás tanításával és szemléltetésével foglalkoznak.

Tartalmuk szerint a tanulmányok következőképpen csoportosíthatók:

- módszertan,
- algoritmus vizualizáció,
- OOP fogalmi vizualizáció.

A kutatásom során elméleti jellegű tanulmányokkal nem találkoztam. Az objektum-orientált programozás elméleti hátterével csak szakkönyvek, ill. középiskolai és egyetemi oktatók jegyzetei és elektronikus tananyagok foglalkoznak.

## 2.1. Módszertani jellegű cikkek

Wing [2] az elsők között fogalmazta meg a számítógépes gondolkodás fogalmát, mint rendszerek tervezése és emberi viselkedés megértése, a számítástechnikai alapfogalmak segítségével. Ugyanezt a fogalmat az International Society for Technology in Education és a Computer Science Teachers Association [3] a következő módon fogalmazta meg:

„A problémák oly módon történő meghatározása, hogy lehetővé tegye számunkra a számítógép és egyéb eszközök használatát, hogy segítsen megoldani azokat; az adatok logikus megszervezése és elemzése; az adatokat a absztrakciókkal szemléltesse, modellek és szimulációk segítségével; algoritmikus gondolkodással automatizálja a megoldásokat (rendezett lépések sorozata); azonosítsa, elemezze, és végrehajtsa a lehetséges megoldásokat elegendő és hatékony lépések és erőforrások kombinációjával; általánosítsa és átranzformálja ezt a probléma megoldási folyamatot különböző problémákra. A számítástechnikai gondolkodás az emberi vagy gépi folyamat valamelyikén alapul. A módszerek és a számítógépes modellek segítik a problémák megoldását és a rendszerek tervezését.”

José-Manuel Sáez-López, Marcos Román-González és Esteban Vázquez-Cano [4] tanulmányukban egy kétéves kutatás eredményeit összegzik. Az általános iskolai programozásba bevezették a Scratch programozási nyelvet, mivel szükségét látták, hogy a tanulók már nagyon fiatalon elsajátítsák a programozás alapfogalmait. Úgy gondolták, hogy a későbbi programozás tanulást sokkal hatékonyabbá tehetik, ha a tanulók már korábban is találkoztak vele. Ez a programozási környezet lehetővé teszi a fiatalok számára, hogy saját interaktív történeteket, játékokat és szimulációkat hozzanak létre, majd megosszák alkotásaikat egy online közösséggel, más fiatal programozókkal a világ minden tájáról. A gyermekek megtanulnak kreatívan gondolkodni, és közösen használják a Scratch-t. A kódolás könnyebb ezen a felületen, mint a hagyományos programozási nyelvek esetében, mivel a gyerekek játszanak, és a színes blokkok segítségével szkripteket hoznak létre. Megértik a ciklus, elágazás és adat bevitel fogalmakat, valamint megtanulják a szerkesztési dizájnt, párhuzamos folyamatokat és egérbillentyű utasításokat.

Andrew P. Black tanulmányában leírja [5], hogy 50 évvel ezelőtt Dahl és Nygaard [6] a SIMULA programozási nyelv létrehozásakor meghatározták az objektum-orientált programozás legfontosabb öt alapfogalmát, melyek a következők: objektum, mint adatszerkezet, és mint eljárásilag egységbezárt absztrakció, öröklődés definíciója, aktív objektumok és objektumok, mint modulok.

Horváth Roman és Javorský Stanislav cikkükben [7] bemutatják egy kutatás eredményeit, melyet a szlovákiai Nagyszombati Egyetem Pedagógia karán végeztek. A cikk a programozás módszertani problémáival foglalkozik, pontosabban az új pedagógiai eljárások alkalmazásával, a Java új kifejlesztett osztályainak tanításakor. Az osztályok implementálva lettek egy mikrokörnyezetbe és kihasználták a Logo grafikai környezetét. A kutatás eredményeként a következő módosításokat végezték el:

- Megváltoztatták a tanítási módszereket, melyek tartalmazták az oktatók leegyszerűsített előadását, a tananyag vizualizációját, több teret adtak az osztály elemzésére.
- Felülvizsgálták a tananyagot.
- Kifejlesztettek egy Java osztály csoportot Graphical Robot néven.
- Nagyobb hangsúlyt adtak a házi feladatokra, ahol analizálni és módosítani kellett az adott programokat.

- Tananyagokat fejlesztettek.
- Minden fejezet előtt kötelezően bevezették a vetélkedőket.
- A tanulókat folyamatosan ösztönözték a programozás tanulására.

Stelios Xinogalos, Maya Satratzemi és Vassilios Dagdilelis cikkükben [8] bemutatják az objectKarel objektum-orientált programozási környezetet, [9] amelyet kezdő programátorok meglátásai alapján szerkesztettek meg. A beépített elektronikus tananyagok, a gyakorlati tevékenységek, a könnyen kezelhető program szerkesztő-fejlesztő ablak, az alkalmazást animáló ablak, magyarázatok és a hibajelentések a programkörnyezetet egyedivé tették. A tanulmány továbbá bemutatja a programozási környezet diákok általi értékelését, valamint a program alkalmazásának eredményeit a tanítási folyamatban.

Robert Biddle és Ewan Tempero [10] az újrafelhasználhatóság elve alapján történő programozás tanításról szóló tanulmányukban egy modellt mutatnak be az újrafelhasználhatóság támogatásáról, amely egy új megközelítés a programozás oktatásában. A modell magába foglalja a környezetet és a komponenseket, valamint a közöttük levő függőségeket. Különösen koncentrálva arra, hogy néhány függőséget használjunk többszöri felhasználásra, testreszabás és változékonyság támogatásával. Az újrafelhasználhatóság elvével lehet magyarázni a programozási nyelvek legfontosabb jellemzőit, mint például az eljárásokat, a felhasználó által definiált típusokat, továbbá a bonyolultabb funkciókat, mint az egységbezárást, öröklődést és polimorfizmust.

## 2.2. Algoritmust vizualizáló jellegű cikkek

Végh László és Stoffa Veronika tanulmányukban [11] leírják, hogy a programozás tanítása folyamán a tanulók számára problémát okoz a rendezőalgoritmusok működésének megértése. Tanulmányukban egy saját fejlesztésű oktató játékot mutatnak be, majd kimutatják a szoftver hatékonyságát az oktatás szempontjából.

James H. Cross II, T. Dean Hendrix, Jhilmil Jain, és Larry A. Barowski tanulmányuk [12] elején bemutatják a jGRASP alkalmazást, mely képes vizualizálni a Javában megírt forráskódokat. A szerzők bemutatják a Java keretrendszer gyűjteményben megtalálható ArrayList, LinkedList, TreeMap, és HashMap adatszerkezeteket. Ezután közlik felmérésük eredményeit. Hallgatók között készítettek egy felmérést, mely a bináris fák ismeretére összpontosult. Két csoportban vizsgálták a hallgatók tudás szintjét. Eredményeik azt bizonyítják, hogy az a csoport ért el jobb eredményeket, melynek tagjai tanulmányaik során használták a jGRASP vizualizációs alkalmazást.

Az elmúlt évtizedekben többen próbálkoztak animációkat használni az algoritmusok megmagyarázására, de ezek csalódást okoztak. A legtöbb esetben érdekes algoritmus animációkat terveztek, de formális, szisztematikus értékelést nem végeztek. Bizonyítékokkal támasztották alá, hogy az algoritmus-animációk oktatási fölénye nem jött létre. Ebben a kontextusban olyan kutatási programot indítottak el, amely oktatási szempontból hatékony algoritmus vizualizációkat fejlesztett ki. Ez a program azon a feltételezésen alapult, hogy az animációkat olyan tudásba és kontextusba kell beágyazni, amely hipermedia környezetet biztosít ahhoz, hogy hatékonyan kihasználhassák a tanulás javítására. A Designing Educationally Effective Algorithm Visualizations [13] című cikkben a szerzők a HalVis Hypermedia Algorithm Visualization rendszer architektúráját írják le. Négy empirikus tanulmányt írtak le a HalVis-szel kapcsolatban, amelyek azt mutatják, hogy a HalVis-t használó diákok által mutatott tanulás mértéke szignifikánsan nagyobb volt, mint a hagyományos oktatás eszközeit vagy tipikus algoritmus-animációkat használó diákoké.

### 2.3. OOP fogalmait vizualizáló jellegű cikkek

Az irodalom kutatása során, az OOP alapfogalmainak valamilyen vizualizációs eszköz felhasználásával való bemutatásával foglalkozó cikkekkel nem találkoztam. Néhány tanulmányban UML segítségével szemléltették az alapfogalmakat. Meglátásom szerint, az általános és a középiskolai oktatás során az UML használata korai, mivel a tanulóknak még nincs tapasztalatuk az algoritmizációval, és így nehéz megérteniük.

## 3. Szlovákiai magyar középiskolák informatika tudásszintjének elemzése

Az első kutatásom során szerettem volna felmérni a tanulók programozási készségeit. A felmérést on-line módszerrel végeztük, mivel szeretnénk volna minél több tanuló véleményét megtudni. Ezt legegyszerűbben on-line kérdőív segítségével érthetjük el. A kérdőívet eljuttattuk Szlovákia majdnem összes magyar nyelvű iskolájába, tehát a válaszokat a teljes magyarlakta területről gyűjtöttük össze. A kérdőív összeállításakor nem csak a programozás tanításával kapcsolatos kérdésekkel foglalkoztunk, hanem szeretnénk volna felmérni a tanulók általános számítógép használati szokásait is.

A felmérés eredményei alapján kimondhatjuk [14], hogy a szlovákiai magyar középiskolákban az OOP tanítása nagyon kezdetleges állapotban van. Ennek okait következő pontokban foglalhatjuk össze:

- A nemzeti kerettanterv nem követeli meg az OOP tanítását, valamint eleve kevés óraszámot biztosít a strukturális programozás tanítására.
- A tanárok nem rendelkeznek megfelelő OOP ismeretekkel.
- A tanárok nem rendelkeznek megfelelő didaktikai szemléltető eszközökkel.
- A tanulók nem érdeklődnek az OOP iránt, nagyon bonyolultnak - absztraktnak tartják.
- A tanulók nagy része nem rendelkezik megfelelő logikus gondolkodással.

Az OOP ismeretének javítására a játékosítás és a kombinált oktatás módszereit javaslom.

## 4. Az OOP-t segítő alkalmazás hatásának vizsgálata a szlovákiai középiskolákban

Az előző kutatásban az OOP-ismeretek felmerésekor az osztály, objektum, objektumorientált program, metódus és öröklődés fogalmak ismeretét is vizsgáltam. A válaszok értékelése során nagyon gyenge eredményt kaptam [15]. A felmérés eredményei alapján megállapítottam, hogy a szlovákiai magyar középiskolákban az OOP tanítása nagyon kezdetleges állapotban van. A gyenge eredmények okainak vizsgálatkor rájöttem, hogy nem elég az informatika tanításának heti óraszámát növelni, hanem ki kell dolgozni a stratégiát (és hozzá egy módszertant, módszereket), valamint ezeket a tanítási módszereket a következő két fő elem bevonásával érdemes kiegészíteni: játékosítás és e-learning.

Ezek mellé azonban szükséges egy jól felépített alkalmazás [16], amelynek segítségével még hatékonyabb lesz az oktatás. A kutatásom is e módszer és az alkalmazás hatékonysága köré épül. A korábbi kutatáshoz képest továbblépésként az alkalmazás használata mellett lemértem, hogy mennyit fejlődött az OOP értése a diákok körében, valamint bizonyítottam hipotéziseimet.

A második kutatásomhoz készítettem egy alkalmazást. Az elkészített program nem végleges, hanem egy próbaverzió, amely a megfogalmazott hipotézisek igazolására, illetve elvetésére szolgál. A kutatásnak nem egy komplex eszköz létrehozása volt a célja, hanem, hogy megmutassa, milyen eszközt kellene kifejleszteni és mely korosztály számára.

A programot elsődlegesen középiskolákba szánom, ahol a tanár irányításával fogják a diákok használni.

#### 4.1 A program hatékonyságának és alkalmazhatóságának felmérése az oktatásban

A tanulók tudásszintje a felmérés előtt megközelítette a nullát, mivel az előző tanulmányaik során nem tanultak objektum orientált programozást. A kutatás érdekében a diákok számára tanfolyamot szerveztünk. A diákokat két csoportba osztottuk: míg a diákok fele az OOP fogalmaival vizuális szemléltetőeszközök nélkül ismerkedett meg, a másik csoport a szemléltető alkalmazást is használta. A tudásfelmérő tesztekben részt vevő diákokat a vizsgált csoportokban különböző felkészültségű gyerekekből állítottuk össze, azért, hogy megvizsgálhassuk mennyire segíti a program az oktatást. Ezért mindkét csoportban voltak gyengébb, közepes és jó képességű diákok közelítve a reprezentatív mintához.

A felmérést a galántai Kodály Zoltán Gimnáziumban és az érsekújvári Jedlik Ányos Elektrotechnikai szakközépiskolában készítettük, mivel Szlovákiában nagyon kevés középiskolában folyik objektum-orientált programozás tanítása. A felmérést 167 tanuló segítségével végeztük, melyből 79 használta az alkalmazást, míg 88 nem használta.[17]

A felmérés során feltettünk elméleti és gyakorlati kérdéseket, melyek segítségével jobban fel tudtuk mérni a tanulók tudását.

#### 4.2 A kísérleti és a kontroll csoportok átlag eredményeinek összehasonlítása

A kísérleti csoport tanulóit a Tanulók cellában 1-essel jelöltem, míg a kontroll csoport tanulóit 0-val.

| Tanuló |                | 1. Mit jelent szerinted az objektum? | 2. Tudnál példát mondani az objektumokra? | 3. Miért kell modellezni? | 4. Milyen összefüggés van a modell és a valóság között? | 5. Tudnál modellre példát mondani tanulmányaidból? |
|--------|----------------|--------------------------------------|-------------------------------------------|---------------------------|---------------------------------------------------------|----------------------------------------------------|
| 0      | Mean           | 2,375                                | 2,852                                     | 2,932                     | 2,864                                                   | 2,341                                              |
|        | N              | 88                                   | 88                                        | 88                        | 88                                                      | 88                                                 |
|        | Std. Deviation | 1,0539                               | ,8241                                     | ,7239                     | ,7608                                                   | 1,0921                                             |
| 1,0    | Mean           | 4,329                                | 4,823                                     | 4,696                     | 4,734                                                   | 4,759                                              |
|        | N              | 79                                   | 79                                        | 79                        | 79                                                      | 79                                                 |
|        | Std. Deviation | 1,2682                               | ,6357                                     | ,8527                     | ,6926                                                   | ,8041                                              |
| Total  | Mean           | 3,299                                | 3,784                                     | 3,766                     | 3,749                                                   | 3,485                                              |
|        | N              | 167                                  | 167                                       | 167                       | 167                                                     | 167                                                |
|        | Std. Deviation | 1,5150                               | 1,2327                                    | 1,1819                    | 1,1859                                                  | 1,5479                                             |

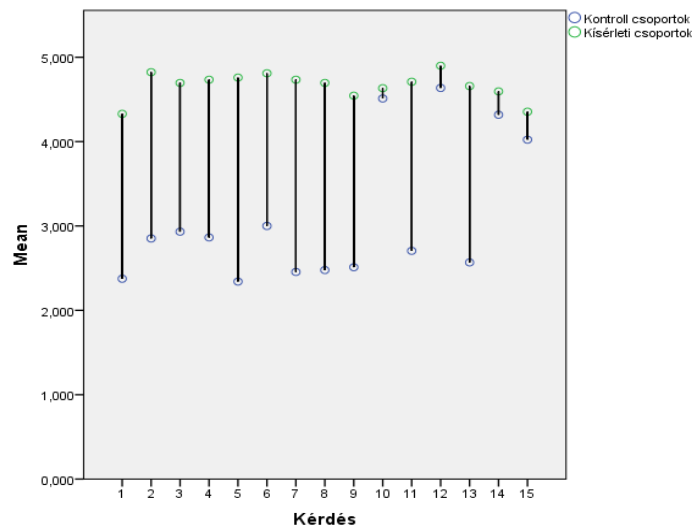
1. táblázat: A két csoport eredményeinek összehasonlítása (1 - 5. kérdések)

| Tanuló |                | 6. Mit jelenthet az általánosítás? | 7. Tudnál ismereteid alapján példát adni rá? | 8. Mit értünk osztály alatt? | 9. Tudnál példát mondani az objektum és osztály közötti különbségre? | 10. Milyen tulajdonság lehet az amikor egy speciálisabb elem megkapja az általánosabb valamelyik tulajdonságát! |
|--------|----------------|------------------------------------|----------------------------------------------|------------------------------|----------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------|
| 0      | Mean           | 3,000                              | 2,455                                        | 2,477                        | 2,511                                                                | 4,511                                                                                                           |
|        | N              | 88                                 | 88                                           | 88                           | 88                                                                   | 88                                                                                                              |
|        | Std. Deviation | ,8305                              | 1,0273                                       | ,9587                        | 1,0613                                                               | 1,1744                                                                                                          |
| 1,0    | Mean           | 4,810                              | 4,734                                        | 4,696                        | 4,544                                                                | 4,633                                                                                                           |
|        | N              | 79                                 | 79                                           | 79                           | 79                                                                   | 79                                                                                                              |
|        | Std. Deviation | ,6217                              | ,8429                                        | ,9655                        | 1,0476                                                               | 1,0763                                                                                                          |
| Total  | Mean           | 3,862                              | 3,533                                        | 3,527                        | 3,473                                                                | 4,569                                                                                                           |
|        | N              | 167                                | 167                                          | 167                          | 167                                                                  | 167                                                                                                             |
|        | Std. Deviation | 1,1508                             | 1,4799                                       | 1,4678                       | 1,4637                                                               | 1,1273                                                                                                          |

2. táblázat: A két csoport eredményeinek összehasonlítása (6 - 10. kérdések)

| Tanuló |                | 11. Tudnál példát mondani? | 12. Lehet-e egy osztályból származó két objektum különböző, más paraméterértékekkel? | 13. Mit jelent ez a gyakorlatban? | 14. Az alábbi programrészlet mit definiál? | 15. Mit nevezünk absztrakciónak? |
|--------|----------------|----------------------------|--------------------------------------------------------------------------------------|-----------------------------------|--------------------------------------------|----------------------------------|
| 0      | Mean           | 2,705                      | 4,636                                                                                | 2,568                             | 4,318                                      | 4,023                            |
|        | N              | 88                         | 88                                                                                   | 88                                | 88                                         | 88                               |
|        | Std. Deviation | 1,1055                     | 1,1565                                                                               | 1,1527                            | 1,5127                                     | 1,2127                           |
| 1,0    | Mean           | 4,709                      | 4,899                                                                                | 4,658                             | 4,595                                      | 4,354                            |
|        | N              | 79                         | 79                                                                                   | 79                                | 79                                         | 79                               |
|        | Std. Deviation | ,8646                      | ,6324                                                                                | ,9592                             | 1,2144                                     | 1,0626                           |
| Total  | Mean           | 3,653                      | 4,760                                                                                | 3,557                             | 4,449                                      | 4,180                            |
|        | N              | 167                        | 167                                                                                  | 167                               | 167                                        | 167                              |
|        | Std. Deviation | 1,4139                     | ,9519                                                                                | 1,4914                            | 1,3826                                     | 1,1528                           |

3. táblázat: A két csoport eredményeinek összehasonlítása (11 - 15. kérdések)



1. ábra: A kísérleti és a kontroll csoportok átlag eredményeinek összehasonlítása

Az objektum fogalmának vizsgálatokor azon tanulók, melyek nem találkoztak az alkalmazásunkkal (kontroll csoport), azok 2,375 átlagot értek el, míg a másik vizsgált csoport 4,329-es átlagot. Gyakorlati tudásuk vizsgálatánál a tanulóknak példát kellett adni az objektumokra, ahol 2,852 és 4,823 átlagokat értek el és azon tanulók értek el jobb eredményt, melyek használták az alkalmazásunkat. A válaszok javításakor nagyon lehetett látni, hogy a tanár milyen példán mutatta be az objektum fogalmát.

„Miért kell modellezni?“ kérdésre a kísérleti csoportok tanulói 4,696 átlagot értek el, míg a kontroll csoportok tanulói 4,696 átlagot. A valóság és a modell közötti összefüggést az alkalmazást használó tanulók (kísérleti csoportok tanulói) 4,734 átlagra tudták és a kontroll csoportok tanulói 2,864-re. Kevesebb tanuló tudott gyakorlati példát mondani, mint az elméleti kérdésre válaszolni. Ebben az esetben 4,759 és 2,341 átlag született. A vizualizáló alkalmazás használatával, mely tartalmaz játékos elemeket is, jelentősen növelhető a tanulóknál a modellezés gyakorlati felhasználásának az ismerete.

Az általánosítás fogalmának ismeretének felmérésekor a kísérleti csoportok tanulói 4,810 átlagot értek el, a kontroll csoportok tanulói 3,000. Mikor példát kértünk az általánosításra, a helyes válaszok esetében nagyobb visszaesést tapasztalhatunk. Itt 4,734 és 2,455 volt az átlag.

Az osztály fogalmának vizsgálatokor a csoportok 4,696, illetve 2,477 átlagokat értek el. A következő kérdésnél a tanulóknak példát kellett adni az osztály és objektum közti különbségre. A kísérleti csoportok tanulói jobban teljesítettek (4,544), mint a kontroll csoportok tanulói (2,511).

„Milyen tulajdonság lehet az amikor egy speciálisabb elem megkapja az általánosabb valamelyik tulajdonságát?“ A tanulóknak előző kérdésre az öröklődés választ kellett volna válaszolni, melyet 4,633, ill. 4,633 átlagra teljesítettek. Ezután példát kellett adni az öröklődésre. „Lehet-e egy osztályból származó két objektum különböző, más paraméterértékekkel?“ kérdésre igen-nem válaszok közül választhattak, ahol az eredményekben elhanyagolható különbség lett. Jelentőségét a gyakorlatban a kísérleti csoportok tanulói jelentősen jobban indokolták.

A programrészlet felismerésénél az eredményekben elhanyagolható különbség tapasztalható.

Az absztrakció fogalmának vizsgálatokor a kísérleti csoportok tanulói 4,354 átlagra tudták, a kontroll csoportok tanulói 4,023-ra.



Az eredmények vizsgálatakor kimutatható, hogy azon tanulók csoportja, mely használta a szemléltető alkalmazásunkat, minden kérdésnél jobban teljesített.

|                                                                                                                 |                     | Tanulók |
|-----------------------------------------------------------------------------------------------------------------|---------------------|---------|
| 1. Mit jelent szerinted az objektum?                                                                            | Pearson Correlation | ,646**  |
|                                                                                                                 | Sig. (2-tailed)     | ,000    |
|                                                                                                                 | N                   | 167     |
| 2. Tudnál példát mondani az objektumokra?                                                                       | Pearson Correlation | ,801**  |
|                                                                                                                 | Sig. (2-tailed)     | ,000    |
|                                                                                                                 | N                   | 167     |
| 3. Miért kell modellezni?                                                                                       | Pearson Correlation | ,748**  |
|                                                                                                                 | Sig. (2-tailed)     | ,000    |
|                                                                                                                 | N                   | 167     |
| 4. Milyen összefüggés van a modell és a valóság között?                                                         | Pearson Correlation | ,790**  |
|                                                                                                                 | Sig. (2-tailed)     | ,000    |
|                                                                                                                 | N                   | 167     |
| 5. Tudnál modellekre példát mondani tanulmányaidból?                                                            | Pearson Correlation | ,782**  |
|                                                                                                                 | Sig. (2-tailed)     | ,000    |
|                                                                                                                 | N                   | 167     |
| 6. Mit jelenthet az általánosítás?                                                                              | Pearson Correlation | ,783**  |
|                                                                                                                 | Sig. (2-tailed)     | ,000    |
|                                                                                                                 | N                   | 167     |
| 7. Tudnál ismereteid alapján példát adni rá?                                                                    | Pearson Correlation | ,771**  |
|                                                                                                                 | Sig. (2-tailed)     | ,000    |
|                                                                                                                 | N                   | 167     |
| 8. Mit értünk osztály alatt?                                                                                    | Pearson Correlation | ,757**  |
|                                                                                                                 | Sig. (2-tailed)     | ,000    |
|                                                                                                                 | N                   | 167     |
| 9. Tudnál példát mondani az objektum és osztály közötti különbségre?                                            | Pearson Correlation | ,696**  |
|                                                                                                                 | Sig. (2-tailed)     | ,000    |
|                                                                                                                 | N                   | 167     |
| 10. Milyen tulajdonság lehet az amikor egy speciálisabb elem megkapja az általánosabb valamelyik tulajdonságát! | Pearson Correlation | ,054    |
|                                                                                                                 | Sig. (2-tailed)     | ,488    |
|                                                                                                                 | N                   | 167     |
| 11. Tudnál példát mondani?                                                                                      | Pearson Correlation | ,710**  |
|                                                                                                                 | Sig. (2-tailed)     | ,000    |
|                                                                                                                 | N                   | 167     |
| 12. Lehet-e egy osztályból származó két objektum különböző, más paraméterértékekkel?                            | Pearson Correlation | ,138    |
|                                                                                                                 | Sig. (2-tailed)     | ,075    |
|                                                                                                                 | N                   | 167     |
| 13. Mit jelent ez a gyakorlatban?                                                                               | Pearson Correlation | ,702**  |
|                                                                                                                 | Sig. (2-tailed)     | ,000    |
|                                                                                                                 | N                   | 167     |
| 14. Az alábbi programrészlet mit definiál?                                                                      | Pearson Correlation | ,100    |
|                                                                                                                 | Sig. (2-tailed)     | ,197    |
|                                                                                                                 | N                   | 167     |
| 15. Mit nevezünk absztrakciónak?                                                                                | Pearson Correlation | ,144    |
|                                                                                                                 | Sig. (2-tailed)     | ,063    |
|                                                                                                                 | N                   | 167     |

4. táblázat: Pearson-féle korreláció eredményei

Második lépésként külön-külön megvizsgáltam, hogy van-e kapcsolat a kísérleti és a kontroll csoportok által adott válaszok eredményei között. A vizsgálatkor a Pearson-féle korrelációs együtthatót határozom meg - amely két metrikus változó közötti kapcsolat erősségét méri - az IBM SPSS Statistics 22 statisztikai szoftver segítségével.

1 - 9., 11. és a 13. kérdések esetében az alábbi táblázatból megállapítható, hogy a kísérleti csoportba tartozók (azaz az alkalmazást használók esetében) a kérdésekre adott magasabb érték kapcsolódik. Az értékek eléggé erős pozitív kapcsolatot jeleznek. A hipotézisvizsgálat is megerősíti a kapcsolat létét, hiszen a szignifikancia 0,000. Azaz, ezen kérdések esetében a tanulók jobb válaszokat adtak, mint a kontroll csoport esetében, akik nem használták.

10. kérdés esetében (Milyen tulajdonság lehet az amikor egy speciálisabb elem megkapja az általánosabb valamelyik tulajdonságát!) nincs szignifikáns eltérés a kísérleti (az alkalmazást használók) és a kontroll csoport esetében.

12. és a 14 – 15. kérdések esetében nagyon kismértékű kapcsolat érzékelhető a csoportba tartozás és az adott válaszok között. Habár a kérdések jellegéből előre várható volt ez az eredmény.

### 4.3 A kutatás tapasztalatainak kiértékelése, elemzése

Egy új szemléletű képzés kialakításának érdekében és a hipotéziseink alátámasztása érdekében egy felmérést készítettünk, amely azt vizsgálta, hogy milyen az informatikaoktatás színvonala a szlovákiai magyar középiskolákban. A felmérésünk alapján látszik, hogy az alap- és a középiskolai számítástechnika oktatás színvonala, óraszámja nem megfelelő. A tanulók sok esetben saját maguk tapasztalatai alapján fejlődnek, nem kapnak megfelelő iskolai támogatást. A felmérésünk alapján kiderült, hogy a programozási ismeret, ami egyébként a kreatív informatikusok képzésében nagyon fontos, igen alacsony szintű és az újabb szemléletek csak elvétve jelennek meg a képzésekben.

A felmérés alapján megerősítést nyertünk abban, hogy igen nagy szükség van egy olyan oktatási módszer kialakítására, amelyik a Z-generáció tagjainak hatékony programozási és OOP képzését segíti. A továbbiakban ezt az e-learning, a játékosítás és egy alkalmazás segítségével szeretnénk megoldani [17].

A szlovákiai középiskolákban az OOP hatékonyabbá tétele érdekében egy kísérletet, mely egy program segítségével történt, és egy felmérést végeztünk. A kimutatott eredményeink alapján a program segítséget nyújt az osztály, objektum, öröklődés fogalmak gyors megértéséhez, így a minimális programozási alapokkal rendelkező tanulók is elmélyíthetik tudásukat a vizualizáció segítségével. A felmérés eredményeiből jól látható, hogy az elméleti fogalmaknál nem változtak különösebben az eredmények, inkább a gyakorlatiasság javult, ezáltal a programozási, modellezési készség javult. Azon tanulók, akiknek lehetőségük volt egyéni gyakorlásra az alkalmazással, tisztábbá vált az objektum és osztály közti különbség. Ezek az eredmények különösen azért fontosak, mert a programozás gyakorlati feladat, mely csoportosan, de mégis egyénileg történik, így a gyakorlati készség különösen fontos szerepet játszik. Kijelenthetjük, hogy szemléltető eszközök segítségével, sokkal hatékonyabban lehet oktatni az objektum orientált programozást.

## References

- [1] Gubán Á. – Kása R. (2013). *A Literature Based Review of Business Process Amelioration Methods and Techniques Regarding Service Orientation*. Journal of Advanced Management Science (Joams) 1:(2) pp. 230-235.
- [2] Wing, J. (2006). *Computational thinking*. *Communications of the ACM*, 49 (3), 33 - 35. <http://dx.doi.org/10.1145/1118178.1118215>.
- [3] International Society for Technology in Education and the Computer Science Teachers Association. (2011). *Operational definition of computational thinking for K12*. <http://csta.acm.org/Curriculum/sub/CurrFiles/CompThinkingFlyer.pdf>.
- [4] J.-M. Sáez-López et al. (2016). *Visual programming languages integrated across the curriculum in elementary school – A two year case study using Scratch in five schools*. *Computers & Education* 97, 129-141 p.
- [5] Andrew P. Black, (2013). *Object oriented programming – Some history and challenges for the next fifty years*. *Information and Computation* 231, pp. 3 – 20.
- [6] K. Nygaard, O. - J. Dahl, (1981). *The development of the SIMULA languages*, in: R. L. Wexelblat (Ed.), *History of Programming Languages I*, ACM, New York, NY, USA, pp. 439 – 480.
- [7] R. Horváth - S. Javorský, (2013). *New Teaching Model for Java Programming Subjects*, 5th World Conference on Educational Sciences - WCES 2013. *Procedia - Social and Behavioral Sciences* 116 (2014), pp. 5188 – 5193.
- [8] S. Xinogalos - M. Satratzemi , - V. Dagdilelis, (2006). *An introduction to object-oriented programming with a didactic microworld: objectKarel*, *Computers & Education* 47, pp. 148–171.
- [9] Bergin, J. - Stehlik, M. - Roberts, J. - R. Karel J. *Robot a gentle introduction to the art of object oriented programming in Java*. Unpublished manuscript. <http://csis.pace.edu/~bergin/KarelJava2ed/Karel++JavaEdition.html>
- [10] R. Biddle, E. Tempero, (1998). *Teaching programming by teaching principles of reusability*, *Information and Software Technology* 40, pp. 203–209.
- [11] L. Végh - V. Stoffová, (2016). *An interactive animation for learning sorting algorithms: How students reduced the number of comparisons in a sorting algorithm by playing a didactic game*. *TMCS 2016/14/1 (4)* - pp. 45-62.
- [12] Cross, J. H., T. D. Hendrix, J. Jain and L. A. Barowski (2007). *Dynamic object viewers for data structures*, in: *Proceedings of the SIGCSE 2007 Technical Symposium*, pp. 4–8.
- [13] Hansen, S. R., N. H. Narayanan and M. Hegarty (2002). *Designing educationally effective algorithm visualizations: Embedding analogies and animations in hypermedia*, *Journal of Visual Languages and Computing* 13, pp. 291–317.
- [14] Udvaros J., Gubán M. (2016). *Szlovákiai magyar középiskolások informatika-tudásszintjének elemzése az objektumorientált programozás oktatásának kialakítására*, In: *Alkalmazott Tudományok III. Fóruma*, Editor Csillag Sára, Budapesti Gazdasági Egyetem, 2016. pp. 851-879. ISBN:978-963-7159-23-7, pp. 851-879.
- [15] Udvaros J., Gubán M. (2016). *Demonstration the class, objects and inheritance concepts by software*. *ACTA DIDACTICA NAPOCENSIA* 9:(1) Paper 3., ISSN 2065-1430
- [16] Avornicului M. (2016). *Bevezetés a számítógépek programozásába*, Ábel kiadó Kolozsvár, ISBN 978-973-114-220-3

- [17] Udvaros J. (2016). *The investigation of OOP helper application effects in Slovakian secondary schools*. Journal of Logistic – informatics - management 2016, volume 2016/1, ISSN 2498-9037.