

Programlama Eğitiminde Görselleştirme ile ACE Döngüsü

İbrahim Çetin¹

Ercan Top²

Özet

Öğrenciler bilgisayar programlamayı öğrenirken çeşitli zorluklarla karşılaşmaktadır. Araştırmacılar öğrencilere bu zorlukları aşmada yardımcı olmak için çalışmalar yapmışlardır. Bu çalışmada matematik eğitimi bağlamında geliştirilmiş olan ACE döngüsü isimli öğretim ortamı programlama eğitimine PACE döngüsü olarak uyarlanmıştır. Çalışmanın amacı PACE döngüsünün etkililiğini nicel ve nitel boyutlarıyla araştırmaktır. Çalışmanın örneklemini 62 makine mühendisliği öğrencisi oluşturmaktadır. Öğrenciler deney ve kontrol gruplarına rastgele atanmıştır. Deney grubunda PACE döngüsü kontrol grubunda ise 'geleneksel öğretim' uygulanmıştır. Öğretim sürecinden önce ve sonra nicel ve nitel yöntemler kullanılarak veri toplanmıştır. Elde edilen sonuçlara göre kontrol ve deney gruplarının başarı ve tutum puanları arasında anlamlı bir fark bulunamamıştır. Fakat sadece bu sonuca bağlı olarak PACE döngüsünün etkililiği yeterince değerlendirilemez. Nitel veriden elde edilen bulgular kontrol grubundaki öğrencilerin verilen öğretimi yenilikçi bir ortam olarak tanımladıklarını göstermektedir. Bu da 'geleneksel öğretim' ortamı tarifinin bağlamsal olarak değişebileceğine işaret eder. Öğrencilerin öğretim ortamı algısı ve deneyimleri çalışmanın sonuçlarını etkilemiş olabilir.

Anahtar Kelimeler: ACE döngüsü, programlama eğitimi, PACE döngüsü, karma desen

Abstract

Students experience difficulties in learning computer programming. Researchers have conducted several studies with different perspectives to help students learn programming. The ACE cycle constructed in the context of mathematics education was adapted to programming education and called PACE cycle. The aim of the study was to test effectiveness of the PACE cycle by using both quantitative and qualitative measures. The sample of the study included 62 mechanical engineering students. The students were randomly assigned to control and experimental groups. Experimental group was instructed by using PACE cycle whereas control group was instructed by using 'traditional instruction'. Both quantitative and qualitative data were gathered before and after the treatment. It was found that there is no significant difference between achievement and attitude scores of experimental and control group students. Nevertheless, it is hard to announce PACE cycle as ineffective depending only on this result. Qualitative data showed that students, in the control group, did not perceive

¹ Yrd. Doç. Dr. İbrahim Çetin, Abant İzzet Baysal Üniversitesi, Eğitim Fakültesi, Ortaöğretim Fen ve Matematik Alanları Eğitimi, ibretin@ibu.edu.tr

² Doç. Dr. Ercan Top, Abant İzzet Baysal Üniversitesi, Eğitim Fakültesi, Bilgisayar ve Öğretim Teknolojileri Eğitimi, ercantop@gmail.com

their instruction as traditional one. This might show us that the definition of 'traditional instruction' can change depending on the context. It might be the case that students' perception and experience related to their instruction affected the results of the study.

Key Words: ACE cycle, programming education, PACE cycle, mixed-method design

1. Giriş

Öğrencilerin bilgisayar programlama konusunda güçlü bir alt yapı oluşturmaya yardımcı olmak Bilgisayar Bilimleri ve Bilgisayar Mühendisliği bölümlerinin temel hedeflerindedir (Zyda, 2009). Daha başka bir çok bölüm için de programlama öğretilmesi gereken temel beceriler arasında yer alır. (Christensen, Rundus, Fujinoki & Davis, 2002; Hodge & Steele, 2002). Bilgisayar Bilimleri ülkelerin ekonomilerine önemli katkılar sağlayan alanlardan bir tanesidir (Zyda, 2009). Araştırmacılar bu bilim dalının öğrenciler tarafından öğrenilmesi üzerine değişik bakış açılarıyla çalışmalar yapmışlardır. Bunlara örnek olarak öğrencilerin kavram yanılgıları, öğrencilerin programlama konularını kavrayışı, usta-çırak karşılaştırması, eşik kavramlar (threshold concepts) ve öğretim metodu karşılaştırma çalışmaları verilebilir (Eckerdal & Thune, 2005; Fleury, 2000; Haberman & Averbuch, 2002; Khalife, 2006; Winslow, 1996). Bakış açısından bağımsız olarak araştırmacılar öğrencilerin programlama öğrenirken önemli zorluklar ile karşılaştığı konusunda hemfikirdirler (Cetin, 2013).

Döngüler öğrencilerin zorluk çektiği önemli konuların arasında yer alır. Pea (1986) öğrencilerin döngü dışındaki ifadeleri döngü içindeymiş gibi çalıştırdıklarını gözlemlemiştir. DuBoulay (1986) programlamaya yeni başlayan öğrencilerin kontrol değişkeninin, döngünün her yinelenişinde arttığını anlamada güçlük çektiklerini belirtmiştir. Benzer bir şekilde Sleeman, Putnam, Baxter ve Kuspa (1986) öğrencilerin kontrol değişkeninin for döngüsü içerisinde değer aldığını farketmediklerini rapor etmiştir. Soloway, Bonar ve Ehrlich (1983) öğrencilerin, oku/işle ve işle/oku olmak üzere, iki farklı döngü stratejisi geliştirdiğini kaydetmişlerdir. Oku/işle stratejisi ikinci elemanın ikinci yinelemede okunup işlenmesine dayanırken işle/oku stratejisinde ikinci eleman işlenir ve bir sonraki değer ikinci yinelemede okunur. Oku/işle stratejisinin öğrenciler tarafından daha kolay benimsendiği ve daha iyi uygulandığı anlaşılmıştır. Buna ek olarak, Ginat (2004) öğrencilerin iç içe döngüler kavramını anlamakta güçlük çektiğini göstermiştir.

Cetin (Yayında) farklı bir yol izleyerek öğrencilerin döngüler konusunu nasıl kavradıklarını araştırmak için APOS (Action, Process, Object, Schema) teorisini kullanmıştır. Bu çalışmada da aynı teori esas alınacağı için Cetin'in (Yayında) elde ettiği sonuçlar vurgulanmadan önce APOS teorisinin detaylandırılmasında fayda vardır. APOS teorisi (Asiala ve ark., 1996) matematik eğitimi bağlamında geliştirilmiş olan oluşturmacı bir çerçevedir. APOS teorisine (Dubinsky, 1991) göre matematiksel öğrenme mekanizması yansıtıcı soyutlamalardır (Beth & Piaget, 1966). Bu mekanizma kullanarak zihinsel yapılar oluşturulur. Bu yapılar eylem (action), süreç (process), nesne (object) ve şemadır (schema). Teoriye göre bir kavramın yapılandırılması var olan nesnelere (mental veya fiziksel) dönüştürülmesiyle (transformation) başlar. Bu ilk dönüştürülme işlemine eylem ismi verilir.

Bu seviyedeki dönüştürmeler öğrenen için henüz dışsaldır. Bunların gerçekleştirilmesi için dışsal ipuçlarına ihtiyaç duyulur.

Öğrenen bir eylemi tekrar ederek ve onun üzerinde düşünerek o eylemi içselleştirebilir (interiorization) (Dubinsky, 1991). Bu içselleştirme sonucunda elde edilen yapıya süreç ismi verilir. İçselleştirme sayesinde öğrenenler artık dışsal ipuçlarına ihtiyaç duymadan dönüştürmeleri gerçekleştirebilir. Süreç öğrenen için içseldir. Öğrenen bu süreçteki her bir adımı gerçekten yapmadan tasavvur edebilir. Süreç elde etmenin tek yolu içselleştirme değildir. Süreçler başka süreçler ile eşgüdüm içine girdiğinde veya bir sürecin tersi alındığında yeni süreçler elde edilebilir.

Eğer öğrenen süreci bütüncül olarak görür ve bu süreç üzerinde başka eylem ve/veya süreçlerin yapılabileceğini farkedirse, süreci nesneleştirir (encapsulation) (Dubinsky, Weller, Stringer & Vidakovic, 2008). Sonuç olarak elde edilen zihinsel yapıya nesne ismi verilir. Süreç öğrenenin yaptığı bir dönüştürme iken, nesne öğrenenin dönüştürmeye uğrattığı statik yapıdır. Çoğu zaman nesne yeniden oluştuğu sürece dönüştürülerek (de-encapsulation) problem çözümü durumlarında kullanılır.

Eylemler, süreçler ve nesnelere şema ismi verilen zihinsel yapılarda organize edilebilir. Bir şema içerisindeki yapılar az yada çok tutarlıdır (Asiala vd., 1996). Şemalar da nesneleştirilerek (thematization) başka şemalar içerisinde bulunabilir. Şemalar bir kere yapılandırıldıklarında matematiksel durumları anlamlandırmada kullanılabilirler. APOS araştırmalarında öğrencilerin zihinsel yapılarını ve bu yapıları elde etmek için kullanılan mekanizmaları tanımlamak için genetik çözümleme (genetic decomposition) aracı kullanılır (Asiala vd., 1996). Bu araç ile öğrencilerin bir konuyu öğrenirken hangi zihinsel yapıları oluşturdukları (eylem, nesne, süreç ve/veya şema) tanımlanmaya çalışılır.

Cetin'in (Yayında) yaptığı çalışma sonucunda döngüler konusunun genetik çözümlemesini oluşturmuştur. Buna göre öğrenciler döngüler konusunu öğrenirken dört aşamalı bir süreçten geçebilir. Bunlardan ilki eylem öncesi kavrayıştır. Bu aşamada öğrenenler döngülerin sözdizimine hâkim değildir. Bir görevin tekrarlı bir şekilde yapılması gerektiğini bilmelerine rağmen döngüleri kullanarak gerekli kodu yazamaz. Bundan sonraki aşama eylem kavrayışı olarak adlandırılır. Eylem kavrayışında olan öğrenenler döngülerin sintaksını (söz dizimi) bilir ve onu kullanarak görece basit problemleri çözebilir. Fakat bu aşamada döngüler öğrenenler için dışsaldır. Döngünün her bir yinelenmesi açık bir şekilde düşünülür veya ifade edilir. Bu eylemleri tekrar ederek ve üzerinde düşünerek, öğrenciler üçüncü aşama olan süreç kavrayışına geçebilir. Bu aşamada öğrenciler döngü gövdesindeki komutları açık bir şekilde düşünmek veya çalıştırmak zorunda değildir. Onların çalıştırıldığını hayal edip döngü adımlarını atlayabilirler. Son aşamaya geçebilmek için öğrencilerin bahsi geçen süreci nesneleştirilmesi gerekir. Nesne kavrayışında öğrenenler döngüyü bir fonksiyon gibi görürler. Kontrol değişkeninin sınırları bu fonksiyonun girdisidir. Döngünün test ifadesinin doğru olduğu müddetçe yinelemenin devam etmesi fonksiyon sürecidir. Döngü bitimindeki döngü ile ilgili ifadelerin çalıştırılması ise çıktı olarak görülür. Döngü bittikten sonra çalıştırılan ifadeler döngü içerisinde olmasa da burada

yapılan işlemler döngü içerisindeki değişkenler ile ilintili olduğundan bu kısımda nesne kavrayışı içerisine dâhil edilir. Bu aşamada öğrenciler farklı iki döngünün çalışma zamanlarını karşılaştırabilir.

Öğrencilerin döngüler konusunu öğrenirken oluşturdukları yapılar ve bu yapıları elde etmek için kullandıkları mekanizmalar bir kez ortaya konulduğunda bunlar kullanılarak öğrencilere döngüler konusunu öğrenmede yardımcı olunabilir. Bu amaçla Asiala ve arkadaşları (1996) ACE döngüsünü geliştirmişlerdir. Bu yaklaşıma göre klasik öğretim biçimi olan ders, soru çözüme ve ev ödevi sırası etkinlikler (Activities), sınıf tartışmaları (Class discussion) ve ev ödevi (Exercises) şeklinde değiştirilir. Öğrenciler ilk olarak bilgisayar laboratuvarında buluşurlar ve verilen etkinlikler üzerinde grupça çalışırlar. Bu etkinliklerin amacı öğrencilerin genetik çözümlemede tarif edilen yansıtıcı soyutlamaları gerçekleştirmelerine yardımcı olmaktır. Daha sonra öğrenciler sınıf ortamında biraraya gelirler ve laboratuvarında yaptıkları etkinlikler üzerinde tartışmalar gerçekleştirirler. tartışmalar grupça ve/veya sınıfça yapılabilir. Bu tartışmaların amacı da bahsi geçen zihinsel yapıların oluşturulmasına yardımcı olmaktır. Bunun yanında öğrenciler sınıfta yeni etkinlikler yapabilir ve nadiren de olsa öğretmen konu anlatıp kavramları isimlendirebilir. Sınıf tartışmalarının ardından öğrencilere bir sonraki hafta teslim edilmek üzere ev ödevleri verilir. Bu ödevlerin amacı öğrencilerin laboratuvarında ve sınıfta öğrendiklerini pekiştirmelerini sağlamaktır. ACE döngüsü Arnon ve arkadaşları (2014) ile Asiala ve arkadaşlarının (1996) çalışmalarında detaylı olarak anlatılmıştır.

Bu çalışmada ACE döngüsü programlama eğitimine uyarlanmıştır. Bu amaç için yazılım görselleştirme aracı kullanılacaktır. Yazılım görselleştirme programlama eğitiminde öğrencilerin öğrenmesini kolaylaştırmak için kullanılan önemli araçlardan bir tanesidir (Stasko, Kehoe & Taylor, 2001; Urquiza-Fuentes & Valezquez-Iturbide, 2009). Yazılım görselleştirme (YG) algoritma görselleştirme (AG) ve program görselleştirme (PG) olmak üzere iki alt alandan oluşur. AG’de yazılım ile ilgili yüksek seviyeli soyutlamalar görselleştirilirken PG’de gerçek program kodları ve veri yapıları görselleştirilir (Price, Baecker & Small., 1998). İnternet üzerinden yüzlerce görselleştirme aracına ulaşmak mümkündür (Shaffer ve ark., 2010).

Hundhausen ve arkadaşları (2002) AG’nin etkililiğini değerlendirmek için 24 deneysel çalışmayı incelemiştir. Bu çalışmalardan bazıları AG’nin eğitimsel etkinliğine vurgu yaparken, bazıları etkili bir yöntem olmadığını, bazıları ise kısmen etkili olduğunu vurgulamıştır. Bunun üzerine araştırmacılar yapılan çalışmaları pedagojik teorilerine göre sınıflandırmışlardır. Algoritma görselleştirmenin öğrencilerin aktif olarak bilgilerini inşa ettikleri araştırmalarda etkili olduğu bulunmuştur. Bunun aksine uzmanların algoritma hakkındaki zihinsel modellerini sunarak öğrencilere aktarmaya çalışan araştırmalarda AG’nin eğitimsel olarak etkili olmadığını gözlemişlerdir.

Naps ve arkadaşları (2002) öğrencilerin aktif olarak inşa sürecine katıldıklarında AG’nin daha etkili olduğu sonucundan yola çıkarak AG araçlarının katılım taksonomisini oluşturmuştur. Bu taksonomi öğrencilerin pasif olmasından aktif olmasına doğru giden bir skalada 6 kategori içerir. Bunlar sırasıyla: (i) görsel olmadan, (ii) sadece görerek, (iii) yanıtlayarak, (iv) değiştirerek, (v) oluşturarak ve (vi) sunarak. İlk kategori olan “görsel

olmadan” herhangi bir görselleştirme teknolojisi içermez. İkinci kategori olan “sadece görerek” ise öğrencinin görselleştirme aracını izlemesini içerir. Üçüncü kategori “yanıtlayarak” olarak adlandırılmıştır. Burada öğrencilerden görselleştirme ile ilgili sorulan soruları yanıtladığı beklenir. Dördüncü kategoriye ise “değiştirerek” ismi verilmiştir. Burada öğrencilerin aktif katılımı biraz daha artar ve onlardan görselleştirme üzerinde değişiklikler yapması beklenir. Beşinci kategori olan “oluşturarak”da öğrencilerin verilen algoritmayı kendilerinin görselleştirmeleri beklenir. Son kategori olan “sunarak”da ise öğrencilerin kendi oluşturdukları algoritma görselleştirmesini sunmaları beklenir. Burada sunumu yapan öğrenciler diğerlerinden dönüt alırlar ve görselleştirmeyi tartışırlar. Myller, Bednarik, Sutinen ve Ben-Ari (2009) katılım taksonomisine PG ile ilgili maddeler ekleyerek YG araçlarının taksonomisini oluşturmuştur. Sonuç itibarıyla yeni taksonomi şu on kategoriye içerir: (i) görsel olmadan, (ii) sadece görerek, (iii) kontrollü izleyerek, (iv) değer girerek, (v) yanıtlayarak, (vi) değiştirerek, (vii) modifiye ederek, (viii) oluşturarak, (ix) sunarak ve (x) değerlendirerek.

Bu çalışmanın amacı görselleştirme yardımı ile ACE döngüsünü programlama eğitimi bağlamında döngüler konusuna uyarlayarak, tasarlanan öğretim ortamının etkililiğini araştırmaktır. Bu çalışmada ACE döngüsünün programlamaya uyarlanmış hali PACE (Programlamada ACE) olarak adlandırılacaktır.

2. Yöntem

Bu çalışmada PACE döngüsünün öğrencilerin programlama başarılarına ve programlamaya karşı tutumlarına etkilerini incelemek ve elde edilen sonuçları açıklamak için karma araştırmaya yöntemi kullanılmıştır. Araştırmanın ilk aşamasında deneysel yöntem, ikinci aşamasında ise birinci aşamada elde edilen sonuçların incelenmesinde kullanılmak üzere nitel yöntemlere başvurulmuştur. Cresswell ve Clark (2007) elde edilen nicel sonuçları açıklamak için nitel yöntemlerin kullanılmasına tamamlayıcı açıklama modeli (follow-up explanation) ismini vermiştir. Bu modelde ağırlık nicel aşamadır, nitel kısım nicel kısımda elde edilen sonuçları açıklamakta kullanılır.

2.1. Katılımcılar

Bu çalışmaya bir üniversitenin makine mühendisliği bölümünde okuyan 62 üniversite öğrencisi katılmıştır. Ortalaması 21 olmak üzere öğrencilerin yaş aralığı 19 ile 28 arasındadır. Çalışmanın nicel aşaması için rastgele atama yöntemiyle deneysel grup ve kontrol grubu oluşturulmuştur. Öğrencilerin 31’i deneysel gruba geriye kalan 31’i ise kontrol grubuna atanmıştır. Çalışma başladığında öğrenciler Programlamaya Giriş dersine devam etmekteydiler. Bu ders öğretim programlarının ikinci sınıfında verilmektedir ve öğrencilerin aldıkları ilk programlama dersidir. Çalışma başlamadan önce öğrenciler bu derste veri yapıları, koşul ifadeleri ve C dili için bir programlama ortamına giriş yapmışlardır.

2.2. Veri Toplama Aracı

Bu çalışmanın verileri açık uçlu sorular içeren bir Başarı Testi (BT), Bilgisayar Programlama Tutum Ölçeği (BPTÖ) ve yarı yapılandırılmış görüşmeler vasıtası ile toplanmıştır.

Başarı Testi

BT bu çalışmanın araştırmacıları tarafından dersin amaçlarına uygun olarak tasarlanmıştır. Dersin amaçlarına göre, bu dersin sonunda öğrenciler (i) for döngüsünün söz dizimini kavrar, (ii) while döngüsünün söz dizimini kavrar, (iii) do-while döngüsünün söz dizimini kavrar, (iv) koşullara bağlı olarak hangi döngü çeşidini kullanacağını ayırdına varır, (v) başkaları tarafından yazılmış döngü komutlarını takip eder, (vi) döngüler kullanılarak yazılmış bir koddaki mantık ve söz dizimi hatalarını bulup ayıklar, (vii) döngüleri kullanarak günlük hayattaki problemlere çözüm bulunabileceğini farkeder, (viii) günlük hayattaki problemleri çözmek için döngüleri kullanır. BT, iki farklı uzman tarafından, kullanılan ifadelerin uygunluğu ve içeriğin dersin amaçlarına uygunluğu açılarından incelenmiş ve BT'nin içerik geçerliliği sağlanmıştır. Testin amacı öğrencilerin döngüler konusunu kavrayışını ölçmektir. Kontrol ve deney grubu öğrencilerine çalışmanın nicel aşamasında ön-test ve son-test olarak uygulanmıştır. Öğrencilerin ön-test ve son-test cevapları bu çalışmanın bir araştırmacısı ve bağımsız bir araştırmacı tarafından notlandırılmıştır. Değerlendirici uyumu (inter-rater agreement) ön-test ve son-test için 0,89 ve 0,92 (Pearson r) olarak bulunmuştur. BT üç çeşit soru içermektedir. Bunlardan ilki sintaks ve mantık hatalarıyla ilgilidir ve öğrencilerin verilen koddaki sintaks ve mantık hatalarını bulmasını gerektirir. Aşağıdaki soru bu tipe örnek olarak verilebilir:

Örnek Sınav Sorusu 1. Aşağıda bir kod parçası ve bu kod parçasının görevi verilmiştir. Verilen koddaki (eğer varsa) hata veya hataları bulup düzeltiniz. Eğer hata yoksa, cevap olarak hata yoktur yazınız.

Görev: 50 ile 100 arasındaki sayıları toplamak.

```
int i,top=0;
for(i=50;i<=100;i++);
    {top=top+i;}
```

İkinci tip soru ise öğrencilere gene bir kod parçası verir ve öğrencilerden bu kod parçasının çıktısını bulmalarını ister. Öğrencilerin verilen kodu anlamlandırmaları ve/veya kodu takip ederek sonucu bulmaları gerekmektedir. Aşağıdaki soru bu tip için bir örnektir:

Örnek Sınav Sorusu 2: Aşağıdaki kod parçasının ekran çıktısı ne olur?

```
int i,j,k,m=0;
for (i=1;i<=5;i++){
  for (j=i;j<=5;j++){
    m=m+i;
  }
  for (k=5;k>=i;k--){
    m=m+i;
  }
}
printf("m=%d\n",m);
```

Üçüncü tip soruda ise verilen bir problemin programlama vasıtası ile çözülmesi istenmektedir. Öğrenciler çözümü C dilini kullanarak yazacaklardır. Aşağıdaki soru bu tip için bir örnektir:

Örnek Sınav Sorusu 3: $a^2 + b^2 \leq 25$ eşitsizliğini sağlayan a, b pozitif tam sayılarını bularak ekrana yazdıran C programını yazınız.

Bilgisayar Programlama Tutum Ölçeği

Öğrencilerin bilgisayar programlamaya karşı tutumlarını ölçmek için kullanılan BPTÖ Cetin ve Ozden (Değerlendirmede) tarafından geliştirilmiştir. Ankette duyuş, biliş ve davranış olmak üzere üç alt boyut bulunmaktadır. BPTÖ her bir alt boyutta altı soru olmak üzere toplamda 18 soru içermektedir. Sorular 5=”Kesinlikle katılıyorum”, 1= “Kesinlikle katılmıyorum” u gösterecek şekilde 5’li Likert tipinde hazırlanmıştır. Anketten alınabilecek en yüksek puan 90 iken en düşük puan 18’dir. Anketin geçerlik ve güvenilirlik çalışmaları Cetin ve Ozden (Değerlendirmede) tarafından daha önce yapılmıştır. Buna göre duyuş alt boyutunun Cronbach alfa güvenilirlik katsayısı 0.90, biliş alt boyutu için 0.80 ve davranış alt boyutu için 0.90 bulunmuştur. Anketin tamamı için güvenilirlik katsayısı 0.93 olarak hesaplanmıştır. Bu çalışmada ise ön-test için bulunan güvenilirlik katsayıları anketin tamamında 0.90, duyuş için 0.86, biliş için 0.82 ve davranış için 0.80 olmuştur. Son-test için bulunan güvenilirlik katsayıları ise anketin tamamında 0.94, duyuş için 0.93, biliş için 0.85 ve davranış için 0.87 olmuştur. Aşağıda herbir alt boyut için birer madde örnek olarak verilmiştir.

- Program yazmak sıkıcıdır. (Duyuş)
- Programlama insanın problem çözme becerisini geliştirir. (Biliş)
- İyi bir programcı olmak için araştırma yaparım. (Davranış)

Görüşmeler

Bu çalışmada diğer bir veri toplama aracı ise yarı yapılandırılmış görüşmelerdir. Yarı yapılandırılmış görüşmelerin amacı çalışmanın nicel boyutundan elde edilen sonuçları açıklamaya yöneliktir. Görüşmeler bu çalışmanın araştırmacılarından birisi tarafından gerçekleştirilmiştir. Görüşmelerde öğrencilere ders, laboratuvar uygulamaları ve grup çalışması ile ilgili görüşleri hakkında sorular sorulmuştur. Sorular, animasyonlar ve sözde kodlar ile ilgili olanlar dışında, deney ve kontrol grubu için aynıdır. Aşağıdaki iki soru görüşme sorularına örnek olarak verilebilir:

Örnek Görüşme Sorusu 1: Programlamaya giriş dersinde animasyonlar üzerinden uygulamalar yaptınız. Animasyonlar hakkında ne düşünüyorsunuz? (Deney grubu için)

- Animasyonlar hoşuna gitti mi? Neden?
 - a) Neler hoşuna gitti, nelerden sıkıldın?
- Animasyonlardan ne öğrendin?
 - a) Animasyonlar döngüler konusunu öğrenimde yardımcı oldu mu? Neden?

Örnek Görüşme Sorusu 2: Bu dersin geliştirilmesi ile ilgili önerilerin var mı? (Her iki grup için)

Bireysel olarak gerçekleştirilen görüşmeler ortalama 30 dakika sürmüştür. Yarı yapılandırılmış görüşmelerin analizi için içerik analizi kullanılmıştır. “İçerik analizinde temel yapılan işlem, birbirine benzeyen verileri belirli kavramlar ve temalar çerçevesinde bir araya getirmek ve bunları okuyucunun anlayabileceği bir biçimde düzenleyerek yorumlamaktır.” (Yıldırım ve Şimşek, 2006, s. 227). Görüşmelerin tamamlanmasının ardından ses kayıtlarının deşifreleri yapılmıştır. Deşifreler bir kelime işlemci programı kullanılarak iki kolonlu bir dökümanın sol kolonuna yerleştirilmiştir. Bu veri, kendi içerisinde anlamlı bir bütün oluşturan bölümlere ayrılmıştır. İkinci kolon kullanılarak her bir bölüm etiketlenmiş ve numaralandırılmıştır. Etiketler kısaca ait olduğu bölümde ne anlatıldığını özetler. Numaralar ise daha sonra bölümlere daha kolay ulaşılmasını sağlamak içindir. Bu süreç araştırmacıların veri ile ilk temas ettiği noktadır. Deşifrelerin tamamı okunmuştur ve araştırmacılar veriye aşina olmuşlardır. Bunun yanında deşifrelerin ileriki analiz aşamalarında daha rahat ele alınabilmesi için düzenlenmiştir. Bunun ardından her bir deşifre yeniden dikkatlice okunmuştur ve bu sefer kodlanmıştır. Bu kodlama işlemi sonucunda kavramlar listesi elde edilmiştir. Kavramlar listesi öğrencilerin ders, laboratuvar uygulamaları ve grup çalışmaları hakkındaki fikirlerini yansıtacak şekilde oluşturulmuştur. Kavramlar yeniden gözden geçirilerek kategori tablosu oluşturulmuştur. Kategori “... elde edilen kavramların birbirleriyle belirli bir tema altında sınıflandırılmasıdır.” (Yıldırım ve Şimşek, 2006, s. 228). Bu tablo her bir öğrenci için kategorileri, kategorilerin kavramlarını, ve kavramların ait olduğu bölümün numarasını içerir.

Araştırmacılar arasındaki tutarlılığın sağlanması için verinin kodlanması üç aşamada gerçekleştirilmiştir. Araştırmacılar verinin bir kısmını ayrı ayrı kodlamıştır. Daha sonra

araştırmacılar bir araya gelip kodlarını karşılaştırmışlardır. Karşılaştırma sonucunda %71 oranında bir birlikteliğin olduğu gözlenmiştir. Araştırmacılar farklılıkları gidermek için bunlar üzerinde düşünüp tartışmıştır. Sürecin ikinci aşamasında her bir araştırmacı verinin %10'una tekabül eden kısmını ayrı ayrı kodlamıştır ve kodlama sonucunda %92 oranında bir birlikteliğin olduğu gözlenmiştir. Bunun üzerine araştırmacılar yeniden farklılıklar üzerinde tartışarak bir görüş birliği oluşturmuşlardır. Kodlamanın son aşamasında ise araştırmacılar verinin tamamını kodlamıştır.

Bu çalışmanın verileri açık uçlu sorular içeren bir Başarı Testi (BT), Bilgisayar Programlama Tutum Ölçeği (BPTÖ) ve yarı yapılandırılmış görüşmeler vasıtası ile toplanmıştır.

2.3. Öğretim Ortamı

Bu bölümde deney ve kontrol grubundaki öğretim ortamları tarif edilecektir. Her iki grup haftada toplamda dört saat olmak üzere üç haftalık bir uygulamaya tabi tutulmuştur. Her iki grupta da dersin iki saatlik kısmı sınıfta ve geriye kalan iki saatlik kısmı ise laboratuvarda yapılmıştır. Kontrol grubu haftaya iki saat süren sınıf aşaması ile başlayıp arkasından iki saatlik laboratuvar aşamasına geçmiştir. Buna karşın deney grubu haftaya iki saatlik laboratuvar aşaması ile başlayıp bunun ardından iki saatlik ders aşamasına geçmiştir. Her iki grupta da bu dört saatlik öğretimin ardından bir sonraki haftaya kadar çözülmek üzere ev ödevi verilmiştir. Çalışma başlamadan önce, kontrol ve deney grubunda, genellikle üçer kişiden oluşan çalışma grupları oluşturulmuştur. Öğrencilerden üç hafta boyunca sınıf ve laboratuvarda yapılan ilgili çalışmaları ve ev ödevlerini grupça yapmaları istenmiştir.

Kontrol Grubu

Kontrol grubunda 'geleneksel öğretim' yaklaşımına uygun bir öğretim ortamı oluşturulmaya çalışılmıştır. Fakat bulgular kısmında görüleceği üzere araştırmanın nitel kısmından elde edilen veri 'geleneksel öğretim' kavramının sorgulanmasına sebep olmuştur. Bu yüzden öğretim ortamının açık bir şekilde tarifi yapılacaktır.

Öğrenciler ve dersin öğretim üyesi her hafta öncelikle sınıfta buluşup iki saat ders işlemiştir. Öğretim üyesi derse genellikle bir soru sorarak başlamıştır. Bu soru öğrencilerin geçmiş bilgilerini tespit etmek ve öğretilen yeni konular için bir temel oluşturmak amacıyla sorulmuştur. Öğretim üyesi temel konuları anlatmış ve anlatılan konular hakkında pratik kod örnekleri çözmüştür. Örnekler bazen tahtada bazen ise derleyici kullanılarak yapılmıştır. Konu anlatımında veya örnek çözümünde öğrencilere zaman zaman bireysel düşünmeleri ve grup veya sınıf olarak tartışmaları için sorular sorulmuştur. Bu sorular genellikle öğrencilerin dikkatlerinin dağıldığı zamanlarda yöneltmiştir. Konu anlatımı ve örnek çözümünün ardından tüm sınıfa tartışmaları için bir programlama problemi verilmiştir. Bu problem öğrenciler tarafından tartışılarak çözülmüştür. Öğretim üyesi, gerektiği hallerde, bazen ipucu vererek bazen de soruyu tamamen çözerek öğrencilere yardımcı olmuştur. Konu sonlarında öğrencilere çözmeleri için programlama problemleri

verilmiştir. Öğrenciler bunlar üzerine bireysel olarak çalışmıştır. Fakat gerekli hallerde dersin öğretim üyesi veya öğrencilerden biri tahtada problemi çözmüştür.

2 saatlik dersin ardından öğrenciler her hafta 2 saat bilgisayar laboratuvarında buluşmuştur. Laboratuvar başlangıcında öğretim üyesi kısaca konu tekrarı yapmıştır. Bunu takiben öğrencilere programlama problemleri içeren laboratuvar föyleri verilmiştir. Öğrenciler problemlerin çözümü için 2 veya 3 kişiden oluşan gruplar halinde çalışmışlardır. Öğrencilere gruplar arası fikir alış veriş için izin verilmiştir. Öğretim üyesi laboratuvarda öğrencilere sorularında yardımcı olmak için bulunmuştur. Öğrencilerin soruları bazen doğrudan yanıtlanırken bazı durumlarda ise öğrencilerin sorunun çözümünü üzerinde düşünceleri için fırsatlar yaratmıştır. Öğretim üyesinin laboratuvarda bulunmasının bir diğer amacı ise öğrencilerin grup çalışmasını uyumlu bir şekilde yapmalarına yardımcı olmaktır. Öğretim üyesi bireysel ve grup olarak dönütler vermiş ve grupları konu üzerinde tartışmaları için cesaretlendirmeye çalışmıştır.

Laboratuvarın ardından öğrencilere her hafta grup olarak çözmeleri için ev ödevleri verilmiştir. Ev ödevleri öğrencilerin derste ve laboratuvarda öğrendiği konuların pekiştirmeleri amacı ile verilmiştir. Öğrencilerden ev ödevlerini bir sonraki hafta laboratuvar saatinden önce teslim etmeleri istenmiştir. Geç gelen ödevler kabul edilmemiştir.

Deney Grubu

Bu çalışmada ACE döngüsü programlama eğitimine adapte edilip PACE döngüsü ismi verilmiştir. ACE döngüsünde öğretimin geleneksel sırası öğrencilere aktif, anlamlı ve işbirlikçi bir öğretim ortamı sunmak için değiştirilir. Dolayısıyla bu çalışmada PACE döngüsüne, geleneksel öğretim ortamının aksine, laboratuvar etkinlikleri ile başlanmıştır. Öğrenciler, her hafta iki saat olmak üzere, grup olarak genetik çözümleme çerçevesinde geliştirilmiş etkinlikler üzerinde çalışmışlardır. Burada amaç öğrencilerin genetik çözümlemeye tarif edilmiş yansıtıcı soyutlamaları gerçekleştirmesine yardımcı olmak ve sınıf aşamasındaki tartışmalar için konu ile ilgili deneyimler elde etmelerini sağlamaktır. Laboratuvardan sonra ikinci aşamaya geçilmiş ve öğrenciler sınıfta buluşmuşlardır. Sınıf aşaması her hafta iki saat sürmüştür. Burada öğrenciler laboratuvar aşamasında elde ettikleri bilgiler ve sınıfta yaptıkları yeni etkinlikler üzerinde grup olarak tartışmışlardır. Bu tartışmalar, gerek görüldüğü durumlarda, sınıf olarak da yapılmıştır. Öğretim üyesinin görevi süreç boyunca öğrencilere kılavuzluk etmek olmuştur. Fakat gerekli durumlarda tanımlar vermiş ve doğrudan açıklamalarda bulunmuştur. Öğretim üyesinin bir diğer görevi ise öğrencilerin grupça çalışmasını sağlamaktır. Bu amaçla hem laboratuvarda hem de sınıfta öğrencilere kişisel ve grup olarak dönütler verip onların konu ile ilgili tartışmalarını cesaretlendirmiştir. Öğretim periyodu öğrencilere bir sonraki haftaya kadar grupça çözülmek üzere verilen ev ödevi ile son bulmuştur.

ACE döngüsünün programlama eğitimine adaptasyonu iki temel engeli beraberinde getirmiştir. Bunlar sintaks ve bilişsel yük olarak adlandırılabilir. Bu engeller döngünün ilk adımı olan etkinlikler ile ilintilidir. Bu çalışmada PACE döngüsünün geriye kalan adımları literatürde (Arnon vd., 2014; Asiala vd., 1996) yaygın bir şekilde tarif edildiği gibi

uygulandığından, bu bölümde etkinlikler kısmının nasıl gerçekleştirildiği detaylı bir şekilde anlatılacaktır.

PACE döngüsü bilgisayar laboratuvarı ile başlamaktadır. Öğrencilerin derslerde döngülerin sintaksını öğrenmeden bilgisayar laboratuvarında döngüler ile ilgili etkinlikler yapmaları gerekmektedir. Bu, dilin sintaksını bilmeden o dil ile ilgili etkinlikler yapmak anlamına gelir. Bu durum bu çalışmanın araştırmacıları tarafından sintaks engeli olarak adlandırılmıştır. Öğrencilerin sintaks problemini ortadan kaldırılmak için ilk birkaç laboratuvar da derste kodlar yerine Türkçeden türetilmiş sözde kodlar (pseudo code) kullanılmıştır. Öğretim üyesi öğrencilere örnek sözde kodlar sağlasa da her öğrencinin anlaşılır olmak kaydı ile kendi sözde kodunu üretmesine izin verilmiştir. Dersin ileriki aşamalarında, sınıf tartışmalarında öğretim üyesi bu sözde kodların C dilindeki karşılığını göstermiştir.

Bilişsel yük olarak adlandırılan ikinci engel yine öğretimin dizilişinden kaynaklanmaktadır. Öğrenciler bahsi geçtiği üzere derslerden önce bilgisayar laboratuvarında grup içinde etkinlikler yapmışlardır. Literatürde de belirtildiği gibi döngüler öğrenciler için öğrenilmesi zor konulardandır (Ginat, 2004; Pea, 1986). Her ne kadar öğretim üyesi, öğrenciler etkinlikleri yaparken, onları desteklemiş olsa da laboratuvar da öğrenciler genel olarak etkinlikleri grup içerisinde kendileri yapmışlardır. Bu durum öğrencilerin kendilerini bir anda programlamanın önemli ve zor konularından biri olan döngüler ile ilgili etkinlik yaparken bulmalarına sebep olur. Döngüler ile ilgili etkinlikler, DuBoulay'ın (1986) ortaya koyduğu ve diğer birçok programlama konusunda olduğu gibi, birkaç farklı bileşenden oluşmaktadır. Mayer (2009) çokluortamın birkaç bileşenden oluşan öğrenilmesi zor konularda öğrencilere yardımcı olmak için kullanılabileceğini söylemiştir. Benzer bir şekilde görselleştirme araçlarının öğrencilerin aktif ve anlamlı öğrenmelerine yardımcı olmak amacı ile kullanılabilmesi iddia edilmiştir (Shaffer vd., 2010). Bahsi geçen bilişsel yükün azaltılması için Flash yazılımı kullanılarak oluşturulan animasyonlarda program görselleştirme öğeleri içerilmiştir. Bu animasyonlar sözde kodu veya gerçek kodu yazılması istenilen programların görselleştirmesini sağlamaktadır. Öğrencilerin görselleştirmeleri seyredirken bir yandan da kod ile ilgili sorulara cevap vermeleri gerekmektedir. Bu yönüyle görselleştirme animasyonlarının YG katılım taksonomisinin beşinci adımında olduğu söylenebilir. Bunlar yardımı ile öğrencilerin laboratuvar da yapacakları aktiviteler sonucunda oluşacağı tahmin edilen bilişsel yük azaltılmak istenmiştir.

Görselleştirme animasyonlarının bir diğer amacı ise öğrencilere genetik çözümleme anlatılan zihinsel yapıları oluşturmada yardımcı olmaktır. Örneğin öğrencilerin döngülerin eylem aşamasından süreç aşamasına geçebilmeleri için eylemleri tekrar edip onlar üzerine düşünmeleri gerekir. Şekil 1'de öğrencilerin döngülerin eylem aşamasından süreç aşamasına geçmesine yardımcı olmak için oluşturulmuş animasyonlardan bir tanesinin başlangıç kısmı görülmektedir. Bu animasyon, kullanıcının girdiği bir sayının tersinin ekrana yazılmasıyla ilgilidir. Animasyon, diğerlerinde olduğu gibi, iki çerçeveden

oluşmaktadır. Soldaki çerçeve program ile kullanıcı arasındaki iletişimi göstermektedir. Burada kullanıcı bir sayı girmiştir ve program bu sayının tersini yazdıracaktır. Buna ek olarak siyah renk ile yazılmış kısımlar ise yazılımın görselleştirilmesidir ve istenilen görevin adım adım nasıl gerçekleştirildiğini gösterir. Sağ çerçevede ise sol çerçevede olanlar ile ilgili soru veya sorular çıkacaktır. Bunun yanında animasyon içerisinde ileri veya geri hareket edebilmek için buton veya butonlar görülecektir.



Şekil 1. Animasyonun başlangıcı

Bu animasyonda sayının tersinin yazılması ile ilgili süreçler adım adım gösterilir (sol çerçevedeki siyah renkli yazılar) ve animasyonun sonunda Şekil 2’de görülen pencere belirir. Sağ pencerede programın görevinin ne olduğu sorulmaktadır. Öğrenciler ileri tuşuna bastığında aşağıdaki sorular sırayla sağ pencerede belirir.

- i. Bu görev kaç adımda gerçekleştirilir,
- ii. Tekrarın her bir adımında yapılan işlem veya işlemleri açıkça yazınız,
- iii. Tekrarın keyfi seçilmiş bir adımında yapılan işlemleri genel olarak tarif ediniz,
- iv. Bu programda kullanıcının gireceği değeri önceden bilmediğinizi varsayarak, programın sözde kodunu yazınız.



Şekil 2. Animasyonun devamı

Öğrenciler bu soruları grup içerisinde tartışarak çözmüşlerdir. Soru çözümünde zorlandıkları durumlarda öğretim üyesinden yardım istemişlerdir. Birçok gruptan benzer soruların geldiği durumlarda öğretim üyesi bu soruları sınıf bazında tartışmıştır. İlk iki soru öğrencileri açık bir şekilde her bir döngü adımı üzerinde düşünmeye sevk eder. Bunun için öğrencilerin döngülerin eylem aşamasında olması gerekir. Üçüncü soruda ise öğrencilerin döngü adımlarına daha genel bir çerçeveden bakmaları gerekir. Burada öğrenciler her bir adımı açıkça gerçekleştirmez ve bu soru öğrencileri süreç aşamasında düşünmeye yönlendirir. Dördüncü soruda öğrencilerden basamak sayısı bilinmeyen bir sayı için program yazmaları istenmiştir. Basamak sayısı bilinmediği için öğrenciler programın her bir adımını açıkça gerçekleştiremez ve böylece bu adımları tasavvur etmesi beklenir. Bu animasyon öğrencilerin eylem aşamasından süreç aşamasına geçmelerine yardımcı olmak için tasarlanmıştır. Toplamda 13 adet animasyon bulunmaktadır. İlk beş animasyon döngülerin eylem aşaması ile daha sonraki beş animasyon döngülerin süreç aşaması ile ve son üç animasyon ise döngülerin nesne aşaması ile ilgilidir. Her bir animasyon grubu için bir hafta ayrılmıştır.

2.4. Uygulama Süreci

Çalışma nicel ve nitel olmak üzere iki boyuttan oluşmaktadır. Nitel boyut nicel boyutu takiben gerçekleştirilmiştir. Çalışmanın başında rastgele atama yöntemiyle 62 öğrenci deney (31 öğrenci) ve kontrol (31 öğrenci) gruplarına atanmıştır. Her iki gruba da, öğretim uygulamaları başlamadan önce, BT ve BPTÖ ön-test olarak verilmiştir. Ön-testin ardından deney ve kontrol grubunda Öğretim Ortamı alt başlığında detaylarının anlatıldığı öğretim uygulanmıştır. Uygulama üç hafta boyunca devam etmiştir. Her hafta iki saati ders iki saati laboratuvar olmak üzere öğrenciler toplamda dört saatlik bir öğretime tabi tutulmuşlardır. Üç haftalık periyodun ardından BT ve BPTÖ son-test olarak her iki gruba da verilmiştir. Bu noktada çalışmanın nitel boyutu başlamıştır. Her bir gruptan rastgele örneklem yöntemiyle

sekiz öğrenci (toplamda 16) nitel boyut için seçilmiştir. Bu öğrenciler ile yarı yapılandırılmış görüşmeler gerçekleştirilmiş ve görüşmeler kaydedilmiştir.

3. Bulgular

3.1. Çalışmanın Nicel Boyutu

Çalışmanın başlangıcında deney ve kontrol grubu öğrencilerinin başarı puanları arasında anlamlı bir fark olup olmadığını test etmek amacıyla BT ön-test olarak uygulanmış ve sonuçlar bağımsız gruplar için t testi ile irdelenmiştir. Elde edilen sonuçlar Tablo 1’de verilmiştir.

Tablo1. BT ön-test sonuçları

Öğrenciler		N	X	SS	Sd	T	P
Başarı puanları	Deney	31	4.16	4.51	1.10	1.40	0.166
	Kontrol	31	2.61	4.17			

Tablo 1’de görüldüğü üzere deney grubundaki öğrencilerin başarıları ile kontrol grubundaki öğrencilerin başarı düzeyleri arasında istatistiksel olarak anlamlı bir fark yoktur ($p>0.05$). Tablo 1 incelendiğinde, öğrencilerin BT puanlarının aritmetik ortalamasının kontrol grubu için 2.61 ve deney grubu için 4.16 olduğu ve BT’ den alınabilecek en yüksek puanın 100 olduğu düşünüldüğünde, bu ortalamaların çok düşük düzeyde olduğu söylenebilir.

Uygulamanın sonucunda deney ve kontrol grubunda bulunan öğrencilerin başarı puanları arasında anlamlı bir fark olup olmadığını test etmek için BT son-test olarak uygulanmış ve elde edilen sonuçlarla bağımsız gruplar için t testi yapılmıştır. Deney grubundaki öğrencilerin başarı puanlarının aritmetik ortalaması kontrol grubundaki öğrencilerin başarı puanlarının aritmetik ortalamasından fazla olmasına rağmen, aradaki farkın anlamlı olmadığı görülmüştür ($p>0.05$). Bu testin sonuçları Tablo 2’ de gösterilmiştir.

Tablo 2. BT son-test sonuçları

Öğrenciler		N	X	SS	Sd	T	P
Başarı puanları	Deney	31	57.6	26.7	6.80	-0.94	0.349
	Kontrol	31	51.2	26.6			

Öğrencilerin uygulama öncesi tutumları arasında bir fark olup olmadığını ölçmek için BPTÖ çalışma öncesinde ön-test olarak uygulanmıştır. Tutum ölçeğinin duyuş, biliş ve davranış alt boyut puanları üzerinde yapılan MANOVA sonuçları deney ve kontrol grubu öğrencilerinin bu boyutlar bakımından anlamlı farklılık göstermediğini ortaya koymuştur [Wilks Lamda ($\lambda=0.97$), $F(3,58)=0.53$, $p=0.66$].

Çalışma sonunda uygulanan yöntemin öğrencilerin tutumları üzerindeki etkisini görmek amacıyla BPTÖ son-test olarak tekrar uygulanmış ve MANOVA sonuçları deney ve kontrol

grubu öğrencilerinin biliş, duyuş ve davranış alt boyut puanları arasında anlamlı bir fark olmadığını göstermiştir [Wilks Lamda ($\lambda=0.92$), $F(3,58)=1.67$, $p=0.18$].

Tutum ölçeğinin üç alt boyutuna ilişkin ortalama ve standart sapma değerleri ile öğretim yöntemine göre alt boyut bazında yapılan tek yönlü ANOVA sonuçları Tablo 3'te gösterilmiştir.

Tablo 3. BPTÖ son-test sonuçları

Değişken	Grup	N	X	S	Sd	F	P
Duyuş	Deney	31	21.4	4.50	1-60	0.001	0.98
	Kontrol	31	21.5	5.94			
Biliş	Deney	31	23.0	3.79	1-60	2.818	0.10
	Kontrol	31	24.6	3.93			
Davranış	Deney	31	18.5	4.41	1-60	0.036	0.85
	Kontrol	31	18.7	4.89			

Tablo 3'ten de görüldüğü gibi deney ve kontrol grupları arasında hiçbir alt boyut için anlamlı bir fark bulunmamıştır [Duyuş: $F(1,60)=0.001$, $p>0.05$; Biliş: $F(1,60)=2.818$, $p>0.05$; Davranış: $F(1,60)=0.036$, $p>0.05$].

3.2. Çalışmanın Nitel Boyutu

Çalışmanın nitel aşamasının amacı, nicel kısımda elde edilen sonuçların detaylı bir şekilde irdelenmesini sağlamaktır. Bu amaca paralel olarak öğrencilere ders, laboratuvar uygulamaları ve grup çalışması hakkında sorular sorulmuştur. Öncelikle grup çalışması ile ilgili elde edilen sonuçlar üzerinde durulacaktır. Çalışmanın nitel boyutundan elde edilen veri özet olarak Tablo 4'de sunulmuştur.

Tablo 4. Nitel verinin özeti

Öğretim Bileşenleri		Değerlendirme	Kategori	Deney Grubu	Kontrol Grubu
Grup Çalışması	Okul	Verimli		8	7
			Farklı bakış açıları	3	1
			Fikir alış-verişi	4	3
			Bilenin bilmeyene anlatması	3	5
	Okul dışı	Verimsiz		-	1
			Pasif rol	-	1
				1	1
				7	7
		Zaman	5	5	
		Mekan	4	3	
		Görev paylaşımı	-	2	

Tablo 4'ün devamı

Laboratuvar Uygulamaları	Animasyonlar	Verimli	7	-	
			Mantık öğrenme	5	-
			Kod yazma	4	-
			Aktif katılım	4	-
			Görsellik	1	-
			Verimsiz	1	-
Geleneksel Öğretim	Verimli	-	7		
		Pekiştirme	-	5	
		Pratik etme	-	3	
		Bilgi paylaşımı	-	2	
		Disiplin	-	2	
		Verimsiz	-	1	
Ders	PACE Döngüsü ve Geleneksel Öğretim	Verimli	7	7	
			Derse katılım	7	7
			Öğretim üyesi – Öğrenci ilişkisi	6	5
			Dikkat	3	4
			Verimsiz	1	1
			Ders ve Laboratuvar	Sözde kod	Verimli
Mantık öğrenme	5	-			
Öğrenmeyi kolaylaştırma	2	-			
Verimsiz	1	-			
Zaman kaybı	1	-			

Grup çalışması okul ve okul dışı olmak üzere iki farklı başlıkta incelenecektir. Okul grup çalışmalarından kastedilen laboratuvar veya sınıf ortamında yapılan grup çalışmalarıdır. Okul dışı grup çalışmalarından kastedilen ise öğrencilerin ödev yapmak için ders sonrasında biraraya gelip yaptıkları çalışmalarıdır. Deney ve kontrol grubundaki öğrenciler okul grup çalışması ile ilgili büyük oranda benzer açıklamalarda bulunmuşlardır. Deney grubundaki öğrencilerin tamamı (sekiz öğrenci) ve kontrol grubundaki sekiz öğrenciden yedisi okul grup çalışmasının verimli olduğunu ve dersi öğrenmelerinde yardımcı olduğunu belirtmiştir. Her iki gruptaki öğrenciler bu durumu benzer sebeplerle açıklamıştır. Öğrencilerden bazıları açıklamalarında birden fazla sebep kullanmıştır. Deney grubundaki sekiz öğrenciden dördü ve kontrol grubundaki sekiz öğrenciden üçü okul grup çalışmalarındaki fikir alışverişinin onlar için faydalı olduğunu belirtmişlerdir. Benzer bir şekilde deney grubundaki üç öğrenci ve kontrol grubundaki beş öğrenci bilenlerin bilmeyenlere anlattığı için okul grup çalışmasının verimli olduğunu belirtmiştir. Aşağıdaki alıntıda kontrol grubundan Öğrenci 11 derste görece olarak daha iyi olan grup arkadaşından faydalanmasının dersi daha iyi anlamasını sağladığını anlatmaktadır.

Öğrenci 11: “Dersteki grup çalışması verimli oluyor. Sonuçta herkes derse geliyor ve bir araya geliyor. Hani mesela dersi dinliyorum, hocayı anlıyorum, kendim çalışıyorum. Ama yapamadığım noktalar oluyor. Bu benim hani anlama kapasitemden kaynaklı da olabilir, onu bilmiyorum ama. Mesela grup arkadaşım benim çok iyi. O olmasa belki bu kadar da

dersi anlamayabilirdim. Laboratuvarlarda bana çok yardımcı oldu. Anında hemen sorunu halledebiliyoruz.”

Deney grubundaki üç öğrenci ve kontrol grubundaki bir öğrenci farklı bakış açılarını görmenin onların öğrenmesine katkıda bulunduğunu söylemiştir. Diğer öğrencilerin aksine kontrol grubundaki bir öğrenci okul grup çalışmasından fayda göremediğini belirtmiştir. Bunun sebebi olarak ise grup çalışmasında grup arkadaşlarının daha aktif kendisinin ise daha pasif bir rol üstlenmesini göstermiştir. Bu öğrenci bireysel çalışma yapmış olması durumunda zorunluluktan ötürü daha fazla çalışıp daha fazla öğreneceğine inanmaktadır.

Okul dışı grup çalışması ile ilgili de kontrol ve deney grubu öğrencileri benzer yanıtlar vermiştir. Fakat okul grup çalışmasının aksine, her iki gruptan da sekiz öğrenciden yedisi okul dışı grup çalışmasının verimli olmadığını belirtmiştir. Öğrencilerden bazıları açıklamalarında birden fazla sebep kullanmıştır. Deney ve kontrol grubundaki yedi öğrenciden beşi ortak zaman ayarlayamadıkları için biraraya gelemediklerini bildirmişlerdir. Benzer şekilde deney grubundaki yedi öğrenciden dördü ve kontrol grubundaki yedi öğrenciden üçü ortak mekan ayarlama sıkıntısı çektiklerini ve bu yüzden biraraya gelemediklerini belirtmiştir. Son olarak kontrol grubundan iki öğrenci grupta bazı üyelerin ödev yapmak istemediğini ve ödevi diğer üye veya üyelerin yapmak zorunda kaldığını söylemiştir. Aşağıdaki alıntıda deney grubundan Öğrenci 6 zaman ve mekan problemlerinden dolayı bir araya gelemediklerini anlatmaktadır.

Öğrenci 6: *“Ödevi yapmak için bir araya gelmekte sıkıntılar yaşadık. Benim zamanım onun zamanına uymuyor. Sürekli ertelemeler oluyor. Sıkıntı oluyor o yüzden. Benim dersim başka saatte arkadaşım dersi başka saatte. Benim grubumdaki arkadaşlarımın ikisi okuldaki yurttaki kalıyor. Ben merkezde kalıyorum. Ben diyorum ki, kütüphaneye geldiğimde 5:30’da buluşalım diyorum mesela. Fakat herkes başka bir yerde olduğu için buluşamıyoruz. Erteliyoruz, perşembeye yapalım. Erteliyoruz, cumaya yapalım. Bir arada olsak belki yine kolay olur ama ayrı yerlerde olduğumuz için zor oluyor.”*

Grup çalışmasının yanında öğrencilerden laboratuvar uygulamalarını değerlendirmeleri istenmiştir. Deney grubu öğrencileri laboratuvar uygulamalarını değerlendirirken animasyonlar üzerine yoğunlaşmışlardır. Sekiz öğrenciden yedisi animasyonların dersi öğrenme açısından faydalı olduğunu belirtmiştir. Bu öğrencilerden beşi animasyonların programların mantığını anlamada faydalı olduğunu iddia etmiştir. Aşağıdaki alıntı bu öğrencilere örnek olarak verilebilir.

Öğrenci 2: *“...Örneğin ben döngüleri kullanmayı bilmiyordum. Başlarda soru çözerken döngü yerine if’ leri kullanmaya çalışıyordum. Animasyonlar ilk başta bunu anlamada yardımcı oldu... Örneğin bir üçgen çizme problemi vardı. Animasyonlar bize bunun mantığını anlamada yardımcı oldu. İlk başta bize, bu sorunun çözümünü hoca bize anlatsaydı bu kadar anlamayabilirdik. Animasyonlarda ilk başta biz kendimiz uğraştık. Hani, nasıl bir mantık yürütebileceğimizi öğrenmiş olduk. Kafamıza oturuyordu yani.”*

Bunun yanında yedi öğrenciden dördü animasyonlardaki program görselleştirme öğelerinin kod yazarken onlar için faydalı olduğunu söylemiştir. Aşağıdaki alıntıda Öğrenci 7 animasyonlardaki program görselleştirmelerin aktivitelerde nasıl yardımcı olduğundan bahsetmektedir.

Öğrenci 7: *“Bir sorun çıkıyordu grup olarak aramızda. Hani bunu nasıl yapacağız, nasıl edeceğiz. Animasyonlarda altta ipuçları vardı (program görselleştirme kısmından bahsediyor). O çok faydalı oluyordu. O ipuçları bize yol gösteriyordu, çok iyi oluyordu.”*

Dört öğrenci ise animasyonların derse aktif katılım sağladığı için faydalı olduğunu belirtmiştir. Aşağıdaki alıntıda Öğrenci 4 derse katılım sayesinde konuların daha kalıcı olduğunu iddia etmiştir.

Öğrenci 4: *“Animasyonlarda kendimiz emek harcadığımız için daha kalıcı oldu. Hoca göstermeden biz kendi fikirlerimizle yapıyoruz (animasyon aktivitelerinden bahsediyor). Üç kişi gelip biraraya tartışıyoruz. Onları bizim kendimiz anlamamız gerekiyor. Eğer önce hoca anlatsaydı ve sonra animasyon gelseydi, biz hocanın yaptığını kopyalayıp yapıştırabilirdik.”*

Animasyonların faydalı olduğunu düşünen öğrencilerden bir tanesi animasyonlardaki görselliğin kendisi için çekici olduğunu iddia etmiştir. Son olarak sekiz öğrenciden biri, diğerlerinin aksine, dersten önce animasyonların uygulanmasının kendisini zorladığını ve klasik yöntemin, yani önce ders daha sonra laboratuvar uygulaması yaklaşımının, daha verimli olacağını söylemiştir.

Kontrol grubundaki sekiz öğrenciden yedisi bilgisayar laboratuvarlarının döngüler konusunu öğrenmede yardımcı olduğunu ifade etmiştir. Bu yedi öğrencinin beşi laboratuvar uygulamalarının derste öğrendikleri şeyleri pekiştirmede etkili olduğunu iddia etmiştir. Bunun yanında üç öğrenci ise derste görülen teorik bilginin laboratuvar da pratik edilmesinin öğrenmelerine katkıda bulunduğunu belirtmiştir. Bahsi geçen yedi öğrencinin ikisi ise laboratuvar da arkadaşları ile bilgi paylaşımı yapmasının döngüler konusunu öğrenmelerine yardımcı olduğunu belirtmiştir. Aşağıdaki alıntıda Öğrenci 9 laboratuvar uygulamalarının neden kendisi için faydalı olduğundan bahsetmektedir. Bu öğrenci hem pekiştirme hem de bilgi paylaşımı kategorilerine örnek olarak gösterilebilir.

Öğrenci 9: *“Laboratuvar uygulamaları döngüler konusunu öğrenmemde yardımcı oldu. Çünkü, hani dersten sonra onu yapmak yani uygulama yapmak kesinlikle faydalı oldu. Derste gördüğümüz şeyleri sonuçta program yazarak kullanmaya çalıştık. Bu bize pekiştirme sağladı. Akılda daha kalıcı oldu. Bir de ödev yaparken mesela tek başımayım. Bir yerde takıldığım zaman bırakabiliyorum. Fakat laboratuvar da arkadaşlarım olsun hoca olsun soru sorabiliyorum.”*

Buna ek olarak iki öğrenci laboratuvarın onlar için disiplin sağladığını, eğer laboratuvar uygulamaları olmasa eve gidip o haftaki konular üzerinde çalışmalarının zor olduğunu belirtmişlerdir. Son olarak ise, bir öğrenci laboratuvar uygulamaları grup olarak yapıldığı için faydalanmadığını iddia etmiştir.

Öğrencilerin laboratuvar ve grup çalışması ile ilgili görüşlerinin ardından ders ile ilgili görüşleri de sorulmuştur. Öğrenciler programlama dersini aldıkları diğer bölüm dersleri ile kıyaslamışlardır. Bu kıyaslama sonucunda ilginç sonuçlar ortaya çıkmıştır. Elde edilen sonuçlar, Tartışma, Sonuç ve Öneriler bölümünde tartışılacağı üzere, ‘geleneksel öğretim’ yaklaşımının ne ima ettiğinin sorgulanmasını sağlamıştır. Hem kontrol hem de deney grubundaki sekiz öğrenciden yedisi derslerin kendileri için fazlasıyla verimli geçtiğini belirtmiştir. Aşağıdaki alıntı kontrol grubundan bir öğrenciye aittir ve bu duruma örnek olarak verilebilir.

Öğrenci 10: *“Dersi gerçekten anladığımı düşünüyorum. Açıkçası ders daha iyi anlatılmaz halinde.”*

Öğrenciler programlama dersinin neden verimli geçtiğini anlatırken diğer bölüm dersleriyle kıyaslamalarda bulunmuştur. Deney grubundan altı, kontrol grubundan ise yedi öğrenci bölüm derslerinde öğrencilerin derse katılımının çok düşük seviyede olduğunu iddia etmiştir. Bununla bağlantılı olarak, her iki grupta da altı öğrenci bölüm derslerinde öğretim üyesi ve öğrenci arasındaki iletişimin miktarının gereğinden az olduğunu belirtmiştir. Yine bununla ilintili olarak deney grubundan dört ve kontrol grubundan üç öğrenci bölüm derslerinde dikkat sıkıntısı çektiklerini ve belirli bir zamandan sonra derslerde dikkat kaybı yaşadıklarını ifade etmiştir. Bunun aksine deney ve kontrol grubundaki sekiz öğrenciden yedisi programlama dersinde öğrencilerin derse katılım sağlaması için gerekli olanakların yaratıldığını belirtmiştir. Aşağıdaki alıntıda kontrol grubundan Öğrenci 12 bölüm derslerini ve programlama dersini kıyaslayarak niçin programlama dersinin daha verimli geçtiğini anlatmaktadır.

Öğrenci 12: *“Diğer derslerde hoca dersi ya slaytlardan işliyor ya da direk anlatıyor. Kendisi anlıyor ama biz bir şey anlamıyoruz. Bu derste ise (programlama dersini kastediyor) teorik olarak sadece çıkıp hoca anlatmıyor. Biz de etkin durumda olduğumuz için soru filan da çözüyoruz. Bu nokta çok hoşuma gidiyor. Bir tek hoca anlatmıyor. Biz derslere katkı sağlıyoruz. Bildiğimiz bilmediğimiz noktaları, tahtaya falan kalkıyoruz çok hoşuma gidiyor.”*

Benzer şekilde deney grubundan altı ve kontrol grubundan beş öğrenci programlama dersinde öğrenci ile öğretim üyesi arasındaki iletişimin yeterli seviyede olduğunu belirtmişlerdir. Son olarak deney grubundan üç ve kontrol grubundan dört öğrenci programlama derslerinde derse takip ederken genellikle dikkat sıkıntısı yaşamadıklarını iddia etmişlerdir. Aşağıdaki alıntıda deney grubundaki Öğrenci 5 programlama derslerinde derslerden kopmadığını anlatmaktadır.

Öğrenci 5: *“Belli bir noktadan sonra da insanın ders dinleyesi gelmiyor açıkçası. Ben kendi üzerimden konuşayım. En fazla beş dakika dinleyebilirim. Fakat bu derste sürekli biz de etkiniz hoca da etkin. Bu şekilde olunca dersten kopmamaya başlıyoruz. Soruları cevaplıyoruz. Bu şekilde olunca insanın derse katılası, soruları cevaplayası geliyor.”*

Yöntem bölümünde anlatıldığı üzere deney grubundaki aktivitelerde önce sözde kod arkasından ise C kodları kullanılmıştır. Kontrol grubunda ise sadece C kodları kullanılmıştır. Görüşmelerde deney grubundaki öğrencilerden sözde kodun kullanımının değerlendirmeleri istenmiştir. Sekiz öğrenciden yedisi sözde kodu faydalı bulurken geriye kalan bir öğrenci ise sözde kodun zaman kaybettiğini ve doğrudan C kodlarıyla başlanması gerektiğini bildirmiştir. Sözde kodu faydalı bulan yedi öğrenciden beşi sözde kodlar sayesinde ezberden çok kodların mantığını anlamaya yöneldiklerini söylemiştir. İki öğrenci ise sözde kodların öğrenmeyi kolaylaştırdığını iddia etmiştir. Aşağıdaki alıntıda Öğrenci 6 sözde kodlardan nasıl faydalandığını anlatmaktadır.

Öğrenci 6: *“Olayın mantığını kavramada en basiti bildiğin dildir. İşin mantığını kavramada bence önemli. Mesela iken (while komutuna karşılık olarak kullandığı sözde kod), şu sayı böyle iken diyebiliyoruz. Fakat while ile anlatamıyoruz. Bence C’de sözlü olarak anlatım da önemli. Yani sözde kodlar sayesinde programı okumayı öğrendim. Yoksa while, for şeklinde olsa ezberleyip geçilecek bir konu olabilirdi. Böylece kendimizce düşünme, yorumlama, mantığını kavrama şansımız oldu.”*

4. Tartışma ve Sonuç

Günlük yaşamdaki problemleri bilgisayar programlama yardımıyla çözmek için sıklıkla tekrar yapıları kullanılır. Döngüler konusu programlama eğitiminde ilk öğrenilen önemli kavramlardan bir tanesidir. Araştırmacılar öğrencilerin bu konuyu öğrenirken çeşitli sıkıntılar yaşadığını göstermişlerdir (Ginat, 2004; Pea, 1986; Soloway, Bonar & Ehrlich, 1983). Cetin (Yayında) öğrencilerin döngüler konusunu nasıl öğrendiğini APOS teorisi ışığında araştırmış ve dört aşamadan oluşan genetik çözümleme sunmuştur. Bu çalışmada, bahsi geçen genetik çözümleme ve programlama öğretiminde yaygın olarak kullanılan görselleştirme marifetiyle, PACE döngüsü isimli öğretim ortamı oluşturulmuş ve etkililiği nicel ve nitel olarak test edilmiştir. Çalışmanın birinci aşamasında BT ve BPTÖ öğrencilere ön-test olarak verilmiştir. Bunu takiben, 3 hafta boyunca, deney grubundaki öğrencilere PACE öğretim döngüsü ve kontrol grubundaki öğrencilere ‘geleneksel öğretim’ uygulanmıştır. Öğretim sürecinin ardından BT ve BPTÖ öğrencilere son-test olarak verilmiştir. Nicel kısımdan elde edilen sonuçları daha detaylı açıklayabilmek için çalışmanın nitel boyutunda her iki gruptan seçilen öğrencilerle yarı-yapılandırılmış görüşmeler yapılmıştır.

Öğrencilerin ön-test BT puanları göz önüne alındığında, deney ve kontrol grubu öğrencilerinin çok düşük düzeyde bir ortalamaya sahip oldukları ve gruplar arasında programlama başarısı açısından anlamlı bir farkın bulunmadığı tespit edilmiştir. Deney ve kontrol grubu öğrencilerinin son-test BT puanları karşılaştırıldığında, deney grubundaki öğrencilerin başarı puanlarının aritmetik ortalaması kontrol grubundaki öğrencilerden fazla olmasına rağmen bu farkın anlamlı olmadığı gözlemlenmiştir. Yalnızca bu sonuca dayanarak PACE döngüsünün başarısız olduğunu söylemek zordur. Bunun yerine PACE döngüsünün istenen başarıyı oluşturmada yetersiz kaldığını veya deney ve kontrol grubuna uygulanan öğretim ortamlarının bu iki grup arasında programlama başarısı açısından anlamlı bir fark yaratacak kadar farklılaşmadığını söylemek daha doğru olur. Bu durum farklı argümanlarla açıklanabilir. Bunlardan ilki YG katılım taksonomisi ile ilintilidir

(Myller vd., 2009). PACE döngüsünde uygulanan görselleştirme animasyonları on kategoriden oluşan YG katılım taksonomisinin beşinci adımı olan Yanıtlayarak'a denk gelmektedir. Hundhausen ve arkadaşları (2002) görselleştirmenin öğrencilerin aktif olarak katılım gösterdiği durumlarda daha etkili olduğunu iddia etmiştir. Beşinci adım öğrencilerin derste pasif kaldıkları bir ortamı tarif etmez. Fakat daha ileriki adımlarda öğrencilerin animasyonları kullanırken daha aktif olmaları gerekir. Naps ve arkadaşları (2002) AG katılım taksonomisinde üst adımlara gidildikçe oluşturulan görselleştirme araçlarının öğrencilerin başarısını arttırmada daha etkili olacağını iddia etmiştir. Bu sonuçlardan yola çıkarak, öğrencilerin PACE döngüsünde kullandıkları animasyonların oluşturularak, sunarak ve değerlendirerek gibi ileri seviye adımlara göre yeniden inşa edilmesinin öğrencilerin başarılarına katkı sağlayabileceği söylenebilir.

Benzer sonuçlar öğrencilerin programlama tutumları için de geçerlidir. Öğrencilerin ön-test BPTÖ puanları gözönüne alındığında, deney ve kontrol grubu öğrencileri arasında BPTÖ'nin duyuş, biliş ve davranış alt boyut puanları arasında anlamlı bir farkın bulunmadığı gözlemlenmiştir. Öğrencilerin son-test BPTÖ puanları gözönüne alındığında, deney ve kontrol grupları arasında hiçbir alt boyut için anlamlı bir fark bulunmamıştır. Bu verilerden yola çıkarak PACE döngüsünün öğrencilerin programlamaya karşı tutumlarını arttırmada etkisiz olduğu söylenemez. PACE döngüsü bu konuda geleneksel öğretim ortamı kadar başarılı olmuştur fakat beklenen etkinin gerisinde kalmıştır denilebilir. Bu sonucu farklı bakış açıları ile açıklamak mümkündür. Bunlardan ilki uygulanan öğretim ortamının süresidir. Üç hafta PACE döngüsünün etkisini göstermesi için kısa bir süre olabilir. Papanastasiou ve Papanastasiou (2004) tutum değişiminin uzun bir süreç gerektirebileceğini belirtmiştir.

ACE döngüsü ve dolayısıyla PACE döngüsü teorik kısım ile başlayan klasik öğretimin sırasını laboratuvar etkinlikleri, ders etkinlikleri ve ev ödevleri olarak değiştirir. PACE döngüsü programlamaya yeni başlayan öğrenciler için, Yöntem bölümünde daha detaylı olarak anlatılan, sintanks ve bilişsel yük engellerini birlikte getirir. Bu engellerin aşılamamış olması deney grubundan beklenen başarının elde edilememesini açıklayabilir. Bu yüzden çalışmanın nitel kısmında deney grubu öğrencilerine sintaks ve bilişsel yük ile ilgili sorular sorulmuştur. Bu engellerden sintaks aşmak için sözde kodlar kullanılmıştır. Deney grubundaki sekiz öğrenciden yedisi sözde kod kullanımının faydalı olduğunu belirtmiştir. En popüler sebep olarak ise sözde kodun programlamanın söz diziminden ziyade mantığına odaklanmalarını sağladığı belirtilmiştir. Davies (1993) bu çalışmaya benzer bir bağlamda öğrencilerin programlamanın dile ait özelliklerine odaklanmak yerine programlama ile problem çözümüne odaklanmalarını önermiştir. Bu sonuçlardan yola çıkarak sintaks engelinin PACE döngüsünün etkililiğini negatif yönde etkilediğini söylemek güçtür.

Öğrencilere verilen animasyonların iki temel amacından bir tanesi programın görselleştirilmesi vasıtasıyla oluşabilecek bilişsel yükün azaltılmasıdır. Çalışmanın nitel kısmında öğrencilere animasyonlar ile ilgili sorular sorulmuştur. Deney grubundaki sekiz

öğrenciden yedisi animasyonları faydalı bulmuş ve animasyonlar ile ilgili pozitif düşüncelerini belirtmiştir. Yine bu yedi öğrenciden dördü animasyonlardaki görselleştirmenin kendilerine kod yazımında kolaylık sağladığını belirtmiştir. Bu verilerden yola çıkılarak program görselleştirmenin beklenen etkiyi gösterdiği iddia edilebilir. Dolayısı ile bilişsel yükün negatif yönde bir etkiye sahip olduğunu söylemek güçtür.

PACE döngüsünde geleneksel eğitimin klasik öğretim sırasını değiştirmedeki temel amaç aktif ve öğrencilerin zihinsel yapılarını oluştururken onlara destek olan bir öğretim ortamı oluşturmaktır. Bu aşamada sorgulanması gereken nokta acaba PACE döngüsü gerçekten bu amaca hizmet etmiş midir. Deney grubundaki sekiz öğrenciden beşi animasyonların programların mantığını anlamada yardımcı olduğunu belirtmiştir. Dört öğrenci ise animasyonların derse aktif katılım sağladığını söylemiştir. Buna ek olarak deney grubundaki görüşmelere katılan sekiz öğrenciden yedisi genel olarak dersi verimli bulduğunu söylemiştir. Buradan yola çıkarak oluşturulan öğretim ortamının öğrencilerin zihinsel yapılarını oluştururken onlara destek olmadığı ve aktif öğrenmeyi desteklemediği söylenemez.

Deney grubundaki öğrencilerin büyük çoğunluğu okulda yapılan grup çalışmalarının verimli olduğunu belirtmesine rağmen okul dışındaki grup çalışmalarının verimli olmadığını söylemiştir. Fakat okul dışı grup çalışmalarındaki verimsizliği beklenen etkinin oluşmamasına sebep olarak göstermek doğru olmaz. Çünkü benzer koşullar deney grubunda da oluşmuştur. Deney grubundaki öğrencilerin büyük bir kısmı okul içi grup çalışmalarını verimli bulurken okul dışı grup çalışmalarının verimsiz olduğunu belirtmiştir. Burada dikkat edilmesi gereken başka bir nokta ise dersin öğretim üyesinin bulunduğu ortamlarda grup çalışmasının verimli, bulunmadığı ortamlarda ise verimsiz olmasıdır. Slavin (1995) öğrencileri sadece bir araya getirerek grup öğrenme ortamlarının oluşturulmasının mümkün olmadığını iddia etmiştir. Dersin öğretim üyesinin okul içi grup çalışmaları boyunca öğrencilere bireysel ve grup olarak dönüt vermesinin, onları etkinlikleri yapması için cesaretlendirmesinin, onlara etkinliklerde yardımcı olmasının ve etkinlik boyunca öğrencileri gözleyip aralarında dolaşmasının öğrencilerin grup çalışmasının verimini etkilediği görülmektedir. Fakat öğrenciler haftalık ev ödevlerini yapmak için biraraya gelmekte sıkıntı çekmişlerdir. Buna çözüm olarak (Johnson & Johnson, 1999) öğrencilerin ev ödevlerinde hem bireysel hem de grupça yapılması gereken kısımlar oluşturulabilir. Öğretim üyesi öğrencilerin ev ödevlerini yapmaları için yer ve zaman planlaması yapabilir. Öğrencilerin grup ödevlerini öğretim üyesine grupça teslim etmesi sağlanıp teslim etme esnasında rastgele olmak kaydıyla öğrencilere çözümleri hakkında sorular sorulabilir.

Nicel ve nitel verilerden yola çıkarak PACE döngüsünün tamamının veya bir kısmının kullanışsız olduğunu söylemek mümkün görünmediğine göre gözlerin kontrol grubuna çevrilmesi gerekir. Bu durumda kontrol grubundaki 'geleneksel öğretim' ortamının sorgulanması gerekir. Bu amaç için şimdiye kadar sorduğumuz soru tersinden sorulmalıdır: nasıl olmuştur da 'geleneksel öğretim' ortamı PACE döngüsü kadar etkili bir öğretim ortamı sunmuştur. Bu araştırma için kontrol grubunda oluşturulan öğretim ortamı genel olarak öğretmen merkezlidir ve düz anlatım ve soru sorma tekniklerini içerir. Hake (1997)

ve Sungur ve Tekkaya' nın (2006) 'geleneksel öğretim' tanımı ile kontrol grubundaki öğretim ortamı örtüşmektedir. Fakat çalışmanın nitel aşamasında kontrol grubundaki öğrencilerin dersi değerlendirmesi istendiğinde ilginç sonuçlarla karşılaşmıştır. Öğrencilerden bir tanesinin söylediği "bu şimdikiye kadar aldığım en iyi ders" cümlesi dikkate değerdir. Kontrol grubundaki sekiz öğrenciden yedisi dersin kendileri için çok verimli geçtiğini belirtmiştir. Bu durumun sebepleri irdelendiğinde öğrencilerin programlama dersi ile diğer bölüm derslerini karşılaştırarak değerlendirmede bulunduğu gözlemlenmiştir. Öğrencilerin bölümlerinde aldığı derslerde genellikle öğrenci ve öğretim üyesi arasındaki iletişimin düşük seviyede olduğu, derslerin fazlasıyla öğretim üyesi merkezli olduğu ve öğrencilerin derslerde dersi takip etmekte dikkat açısından sıkıntı yaşadıkları ortaya çıkmıştır. Bu bize öğretim üyesinin dersin başından itibaren tahtaya sırtını dönüp konuyu anlattığı veya slaytları sunduğu, nadiren soru sorduğu ve nadiren öğrencilerle iletişime geçtiği bir öğretimi tarif etmektedir. Bu araştırma için oluşturulan kontrol grubundaki öğretim ortamında dersin öğretim üyesi her ne kadar dersi tahtada anlatmış ve derste otorite konumunda olsa da; (i) öğretim üyesi ders başlarında öğrencilerin geçmiş bilgilerini gözlemlemek için soru sorup onların dikkatini derse toplamıştır, (ii) öğrenciler laboratuvar etkinliklerinde kendi aralarında tartışmış ve dersin öğretim üyesine soru sorma fırsatı yakalamıştır, (iii) öğrenciler derslerde ve laboratuvarlarda öğretim üyesinin sorduğu soruları bireysel veya tüm sınıf olarak tartışmıştır, (iv) öğretim üyesi öğrencilere kod yazmada yardımcı olmuştur ve (v) öğretim üyesi laboratuvar başlarında konu tekrarı yapmıştır. Aslında bu beş maddenin tamamı Hake (1997) ve Sungur ve Tekkaya' nın (2006) 'geleneksel öğretim' ortamı tanımı ile uyumludur. Farklı bir şekilde söyleyecek olursak bu maddeler eğitim fakülteleri ve eğitim araştırmaları bağlamında 'geleneksel öğretimi' tarif eder. Fakat görüşme verilerinden anlaşıldığı üzere kontrol grubu öğrencileri için bu ortam yenilikçi bir öğretim ortamı olarak algılanmıştır. Bu durum sorduğumuz yeni sorunun cevabı olabilir.

5. Öneriler ve Sınırlılıklar

ACE döngüsü matematik eğitimi bağlamında oluşturulmuş (Asiala vd., 1996) ve ilk defa bu çalışma ile programlama eğitimine adapte edilerek PACE döngüsü ismi verilmiştir. Bu çalışma ile PACE döngüsünün etkililiği araştırılmak istenmiştir. Fakat elde edilen bulgulara göre PACE döngüsünün etkili bir yöntem olduğunu söylemek ne kadar zor ise etkisiz bir yöntem olduğunu söylemek de o oranda zordur. PACE döngüsünün etkililiğini araştırmak için araştırmacılar tarafından yeni çalışmaların yapılması gerekmektedir. Bu araştırmadan çıkan sonuçlara göre makine mühendisliği bölümü bağlamındaki 'geleneksel öğretimin' eğitim araştırmacıları tarafından tarif edilen öğretimden farklı olduğu anlaşılmaktadır. Bu sonuca dayanarak aslında mutlak bir 'geleneksel öğretim' ortamının olmadığı ve farklı bağlamlarda farklı 'geleneksel öğretim' tanımlarının çıkma ihtimalinin olduğunu söylemek yanlış olmaz. Dolayısı ile araştırmacıların araştırmalarında genel bir 'geleneksel öğretim' tanımı yapmak yerine çalışmayı gerçekleştirdikleri kurumda sıklıkla uygulanan öğretim tekniklerini inceleyip kendi uygulayacakları teknikleri çıkan sonuçlara göre oluşturmaları daha uygun bir yöntem olacaktır.

Bu çalışmanın çeşitli sınırlılıkları bulunmaktadır. Çalışma küçük bir örneklem grubu ile gerçekleştirilmiştir ve PACE döngüsü sadece bu çalışmada kullanılmıştır. Dolayısı ile bu çalışmadan yapılacak genellemeler sınırlıdır. Daha büyük örneklem grubu ile ve farklı konular üzerinde çalışmalar yapıp genellemelerin buraldan çıkacak sonuçlara göre yapılması daha uygun olacaktır.

ACE Cycle in Programming Education by Using Visualization

Extended Abstract

Loops are among the concepts that students experience difficulties in programming education. Sleeman, Putnam, Baxter, and Kuspa (1986) suggested that students are unable to see that control variable has a value inside a for loop. Moreover, Ginat (2004) showed that students have difficulties with nested loops. Cetin (In Print) explored students' understanding of loops concept by using APOS theory. It was suggested that students construct four phases (or mental constructions) when learning loops concept.

In APOS studies, researchers design a learning environment called ACE (Activities, Class Discussion, Exercises) (Asiala et al., 1996) cycle after figuring out phases of students' learning of a concept to teach the concept. ACE cycle is constructed in the context of mathematics education. In this study, ACE cycle will be adapted, by using program visualization, to the programming education. It will be called PACE (Programming ACE) cycle. The purpose of the study was to test effectiveness of PACE cycle. Hundhausen et al. (2002) showed that visualization is educationally effective if it helps students construct their own understandings by playing an active role in AV activities. Moreover, Myller, Bednarik, Sutinen, and Ben-Ari (2009) constructed the engagement taxonomy including: no viewing, viewing, controlled viewing, entering input, responding, changing, modifying, constructing, presenting, and reviewing.

In this study, follow-up explanation model of mixed-method design was utilized. At the beginning of the study 62 mechanical engineering students were randomly assigned to experimental and control groups. Then they were given an achievement test (AT) as a pre-test constructed by the researchers and its content validity was provided by two experts in the field. Moreover, they were given a computer programming attitude scale (CPAS) (Cetin & Ozden, In Review) as a pre-test.

Following the pre-tests, students in the experimental group were instructed by PACE cycle and students in the control group were given 'traditional instruction' for three weeks. In the PACE cycle, regular sequence of the instruction is replaced with a sequence: computer laboratory, class meetings, and homework. This could cause two learning obstacles for students. These are cognitive load and syntax obstacle. Syntax obstacle is that students have to study on programming problems without knowing syntax of programming language. Cognitive load is that students have to concentrate on different aspects of programming before they are formally given the concepts. Pseudo-codes and visualization were planned to be used to overcome these obstacles. Flash animations were designed to help students to construct mental mechanisms defined by Cetin (In Print). In contrast to experimental group, instructor mainly used direct instruction and questioning methods in the control group.

Following the instruction both group were given AT and CPAS as post-test. Then 16 students were randomly selected for semi-structured interviews. Interviews were transcribed and analyzed by using content analysis (Yildirim and Simsek, 2006).

Results showed that there was no significant difference between experimental and control groups with respect to AT before the treatment ($p>0,05$). Although the mean score of the experimental group was higher than the mean score of control group with respect to AT, there was no significant difference between experimental and control groups after the treatment ($p>0,05$). In a similar way, there was no significant difference between scores of experimental and control group students in cognitive, affective, and behavior dimensions of CPAS before the treatment [Wilks Lamda ($\lambda=0,97$), $F(3,58)=0.53$, $p=0.66$]. Moreover, MANOVA results on post-test CPAS showed that the difference between the experimental and control group with respect to cognitive, affective, and behavior dimensions of CPAS were not significant after the treatment [Wilks Lamda ($\lambda=0,92$), $F(3,58)=1.67$, $p=0.18$].

Qualitative dimension of the study showed that the majority of the students who attended to interviews in both groups see group work in school as an efficient method whereas they see after-school group activities inefficient. Majority of the students who attended to interviews in the experimental group stated that animations help them to understand loops concept. Similarly, majority of the students who attended to interviews in the control group believed that laboratory activities helped them to learn loops concept. Moreover, most of the students in both groups stated that the course was efficient enough to help them to understand loops concept.

No significant difference was detected between two groups with respect to post-AT scores. This might be due to the fact that the animations provided to students can be classified at the fifth level of the Program visualization engagement taxonomy, namely responding. It might be said that if the animations was designed at the higher levels of the taxonomy, then they would be more helpful (Hundhausen et al., 2002). Similarly, it was seen that there was no significant difference between two groups with respect to post-CPAS scores. The limited time dedicated to the treatment in the study might explain this result. Papanastasiou and Papanastasiou (2004) suggested that attitude change can be a long time process.

The results from the qualitative part of the study indicate that students in the control group perceived the 'traditional instruction' as a novelty although instructor mainly used direct instruction and questioning in their group. Although this instruction was defined as traditional one by the researchers in the field of education (Hake, 1997; Sungur & Tekkaya, 2006), control group students did not perceive it as traditional. It was understood that, in the courses they take from their department, students have very limited interaction with their instructors, and instructors rarely utilize questioning in their classes, and students have attention problems in their courses. Nevertheless, in the control group, instructor used questioning method, there was some interaction between instructor and students and among students, and students in general did not have attention problems in the treatment. This might explain the results related to achievement and attitude acquired through quantitative data.

It can be said that 'traditional instruction' does not have absolute meaning and it can change depending on the context. Researchers should take into consideration what the traditional is in the context they are doing research. Moreover, from the results of this study, it is hard to conclude that PACE cycle is an ineffective method. Further studies should be done to explore the effectiveness of PACE cycle.

Kaynaklar/References

- Arnon, I., Cottrill, J., Dubinsky, E., Oktac, A., Fuentes, S. R., Trigueros, M., & Weller, K. (2014). *APOS theory: A framework for research and curriculum development in mathematics education*. Newyork/London: Springer/ Heidelberg Dordrecht.
- Asiala, M., Brown, A., Devries, D. J., Dubinsky, E., Mathews, D., & Thomas, K. (1996). A framework for research and curriculum development in undergraduate mathematics education. In J. Kaput, A. H. Schoenfeld & E. Dubinsky (Eds.), *Research in Collegiate Mathematics Education II* (pp. 1-32). Providence, RI: American Mathematical Society.
- Beth, E. W., & Piaget, J. (1966). *Mathematical epistemology and psychology*. Dordrecht: Reidel.
- Cetin, I. (2013). Visualization: A tool for enhancing students' concept images of basic object-oriented concepts. *Computer Science Education*, 23(1), 1-23.
- Cetin, I. (Yayında). Students' understanding of loops and nested loops in computer programming: An APOS theory perspective. *Canadian Journal of Science, Mathematics and Technology Education*.
- Cetin, I., & Ozden, M. Y. (Değerlendirmede). Development of computer programming attitude scale.
- Christensen, K., Rundus, D., Fujinoki, H., & Davis, D. (2002). A crash course for preparing students for a first course in computing: did it work. *Journal of Engineering Education*, 91(4), 409–413.
- Creswell, J. W., & Clark, V. L. P. (2007). *Designing and conducting mixed methods research*. California: Sage Publications.
- Davies, S. P. (1993). Models and theories of programming strategy. *International Journal of Man-Machine Studies*, 39, 237–267.
- Dubinsky, E. (1991). Reflective abstraction in advanced mathematical thinking. In D. Tall (Ed.), *Advanced Mathematical Thinking* (pp. 95-126). Boston: Kluwer.
- Dubinsky, E., Weller, K., Stringer, C., & Vidakovic, D. (2008). Infinite iterative process: the tennis ball problem. *European Journal of Pure and Applied Mathematics*, 1(1), 99-121.
- DuBoulay, B. (1986). Some difficulties of learning to program. *Journal of Educational Computing Research*, 2(1), 57–73.
- Eckerdal, A., & Thune, M. (2005). Novice Java programmers' conceptions of "object" and "class", and variation theory. In *ITiCSE '05: Proceedings of the 10th Annual SIGCSE Conference on Innovation and Technology in Computer Science Education* (pp. 89–93). New York: ACM Press.
- Flcury, A. E. (2000). Programming in java: student constructed rules. *SIGCSE Bulletin*, 32(1), 197–201.
- Ginat, D. (2004). On novice loop boundaries and range conceptions. *Computer Science Education*, 14(3), 165-181.
- Haberman B., & Averbuch H., (2002). The case of base cases: Why are they so difficult to recognize? – Students difficulties with recursion. *SIGCSE Bulletin*, 34(3), 84-88.
-

- Hake, R. (1998). Interactive engagement versus traditional methods: A six-thousand student survey of mechanics test data for introductory physics courses. *American Journal of Physics*, 66 (1), 64-74.
- Hodge, B. K., & Steele, W. G. (2002). A survey of computational paradigms in undergraduate mechanical engineering education. *Journal of Engineering Education*, 91(4), 415-417.
- Hundhausen, C., Douglas, S., & Stasko, J. (2002). A meta-study of algorithm visualization effectiveness. *Journal of Visual Languages and Computing* 13(3), 259-290.
- Johnson, D. W., & Johnson, R. T. (1999). *Learning together and alone: Cooperative, competitive, and individualistic learning* (5th Ed.). Boston, MA: Allyn & Bacon.
- Khalife, J. T. (2006). Threshold for the introduction of programming: Providing learners with a simple computer model. In Romero, P., Good, J., Acosta, E., & Bryant, S. (Eds.), *Proceedings of the 18th Workshop of the Psychology of Programming Interest Group* (pp. 244-254).
- Mayer, R. E. (2009). *Multimedia Learning*. Cambridge: New York, NY.
- Myller, N., Bednarik, R., Sutinen, E., & Ben-Ari, M. (2009). Extending the engagement taxonomy: software visualisation and collaborative learning. *ACM Transactions on Computing Education*, 9(1), 1-27.
- Naps, T. L., Rößling, G., Almstrum, V., Dann, W., Fleischer, R., Hundhausen, C., Korhonen, A., Malmi, L., McNally, M., Rodger, S., & Velázquez-Iturbide, J. Á. (2002). Exploring the role of visualization and engagement in computer science education. In *ITiCSE-WGR '02: Working group reports from ITiCSE on Innovation and technology in computer science education*, (pp. 131-152). New York, NY, USA: ACM Press.
- Papanastasiou, C., & Papanastasiou, E. C. (2004). Major influences on attitudes toward science. *Educational Research and Evaluation*, 10(3), 239-257.
- Pea, R. D. (1986). Language-independent conceptual “bugs” in novice programming. *Journal of Educational Computing Research*, 2(1), 25-36.
- Price, B., Baecker, R. M., & Small, I. (1998). An Introduction to software visualization. In J. Stasko, J. Domingue, M. Brown, & B. Price (Eds.), *Software Visualization: Programming as a Multimedia Experience* (pp. 3-27). MIT Press.
- Shaffer, C. A., Cooper, M. L., Alon, A. J. D., Akbar, M., Stewart, M., Ponce, S., & Edwards, S. H. (2010) Algorithm visualization: The state of the field. *ACM Transactions on Computing Education*, 10, 1-22.
- Slavin, R. E. (1995). *Cooperative learning: Theory, research, and practice* (2nd Ed.). Boston: Allyn & Bacon.
- Sleeman, D., Putnam, R. T., Baxter, J., & Kuspa, L. (1986). Pascal and high-school students: A study of misconceptions. *Journal of Educational Computing Research*, 2(1), 5-23.
- Soloway, E., Bonar, J., & Ehrlich, K. (1983). Cognitive strategies and looping constructs: an empirical study. *Communications of the ACM*, 26(11), 853-860.
-

- Stasko, J., Kehoe, C., & Taylor, A.(2001). Rethinking the evaluation of algorithm animations as learning aids: An observational study. *International Journal of Human Computer Studies*, 54(2), 265–284.
- Sungur, S., & Tekkaya, C. (2006). Effects of problem-based learning and traditional instruction on self-regulated learning. *The journal of Educational Research*, 99 (5), 307-320.
- Urquiza-Fuentes, J., & Velázquez-Iturbide, J. Á. (2009). A Survey of successful evaluations of program visualization and algorithm animation systems. *TOCE*, 9(2), 1-21.
- Winslow, L.E. (1996). Programming pedagogy - A psychological overview. *SIGCSE Bulletin*, 28(3), 17-22.
- Yıldırım, A. ve Şimşek, H. (2006). Sosyal bilimlerde nitel araştırma yöntemleri. Ankara: Seçkin Yayıncılık.
- Zyda, M. (2009). Computer science in the conceptual age. *Communications of the ACM*, 52(12), 66-72.

Kaynak Gösterme

Çetin, İ. ve Top, E. (2014). Programlama eğitiminde görselleştirme ile ACE döngüsü. *Türk Bilgisayar ve Matematik Eğitimi Dergisi*, 5(3), 274-303.

Citation Information

Çetin, İ., & Top, E. (2014). ACE cycle in programming education by using visualization. *Turkish Journal of Computer and Mathematics Education*, 5(3), 274-303.