Programlama Eğitiminde Probleme Dayalı Öğretim Modelinin Klasik Model ile Karşılaştırılması Araştırması

Doç. Dr. Ahmet Kaya

ORCID ID: <https://orcid.org/0000-0002-6105-0787>

Ege Üniversitesi, Tire Kutsan Meslek Yüksekokulu, Bilgisayar Programcılığı Bölümü, İzmir - TÜRKİYE

Makale Geçmişi

Geliş:08.07.2018

Kabul: 21.09.2018

On-line Yayın:30.09.2019

Anahtar Kelimeler

Programlama Eğitimi

PDÖ

Klasik Öğrenme Modeli

Öğrenme Dinamikleri

Etkin Öğrenme

Araştırma Makalesi

DOI:

<http://dx.doi.org/10.29228/JASSS.28867>

Öz

Programlama araçlarının sürekli gelişmesi ve sektördeki değişimin çok büyük boyutlarda olması, meslek elemanları ve öğrenciler için kendi kendine öğrenme stratejisini zorunlu kılmaktadır. Öğrenilen bir yazılım aracının kısa sayılabilecek bir sürede güncelliğini kaybetmesi ve yeni araçların çok kısa bir zaman içinde öğrenilme zorunluluğu, öğretimde yeni dinamiklerin hayata geçirilmesini zorunlu kılmaktadır. Bu zorunluluk, özellikle programlama öğretiminde, kendi kendine öğrenme dinamikleri üzerine kurulu, Probleme Dayalı Öğrenme (PDÖ) modelini çok önemli bir destek modeli haline getirmektedir. Böylece klasik model ve PDÖ modeli ile karma bir öğretim modeli yaratmak ve bu modeli uygulamak mümkün olabilir. Bu amaca uygun olarak Ege Üniversitesi, Tire Kutsan Meslek Yüksekokulu Bilgisayar Programcılığı bölümünde öğrenim gören 100 öğrenci, 50'şer kişiden oluşan iki gruba ayrılarak, geleneksel öğretim modeli (kontrol grubu) ve PDÖ modeli (test grubu) ile C# programlama dilinde metot konusunu öğrenmeye tabi tutuldular. Metot konusunun hangi düzeylerde öğrenildiğini test etmek amacıyla öğrencilere 5'er adet soru soruldu ve 100 puan üzerinden değerlendirmelerle öğrencilerin konuya hâkimiyetleri test edildi. Sınav kâğıtları, konusunda uzman üç öğretim elemanının değerlendirdiği puanların ortalaması üzerinden yapıldı. Ortalamaların virgülden sonraki değerleri yuvarlandı ve notların tamsayı değerlerden oluşması sağlandı. İki farklı öğretim modeli ile öğretim gören öğrenci gruplarının aldığı puanlara ortalamalar arası fark testi uygulandı, test işlemi sonunda PDÖ modeli ile öğrenim yapılan grubun, daha başarılı olduğu sonucuna ulaşıldı

Atıf Bilgisi / Reference Information

Kaya, A. (2019). Programlama Eğitiminde Probleme Dayalı Öğretim Modelinin Klasik Model İle Karşılaştırılması Araştırması. *Jass Studies-The Journal of Academic Social Science Studies*, Number: 76, Autumn, p. 325-338.

A Research for Comparison of Problem Based Learning Model with Classical Model in Programming Training

Assoc. Dr. Ahmet Kaya

Ege University, Tire Kutsan Vocational School, Department of Computer Programming, Izmir – TURKEY

Article History

Submitted:08.07.2018

Accepted: 21.09.2018

Published Online:30.09.2019

Key Words

Programming Training
PBL
Classical Teaching Model
Dynamics of Training
Effective Learning

Research Article

DOI:

<http://dx.doi.org/10.29228/JASSS.28867>

Abstract

Continuous development of programming tools and the large size of the changes in IT training sector requires the self learning strategy. Learned a software tool which can be considered a tool to be come obsolete in a short time and new tools in this are a have to be learned in a very short time which makes education compulsory implementation of a new dynamic. This requirement especially in programming education, Problem Based Learning (PBL) board on the dynamics of self-learning (PBL) model makes it a very important alternative. In this way, it may be possible to create a hybrid model for education. In accordance with this objective, the students studying at Ege University Tire Kutsan Vocational High School, Computer Programming Department divided into two groups consisting each of 50. Traditional teaching model is named control group and PBL model is test subjected to learning method in the in C# programming language. In order to test whether the method learned at what level, students were asked for 5 questions on the subject methods and test results about mastery of subject were questioned. Evaluation is made based on the average of the scores by three instructors. The values of decimal points are rounded. The difference between the two mean scores was applied to the scores of the groups of students who were taught with two different teaching models. At the end of the test process the group that was teaching with PBL was found more successful than other educational method.

1. GİRİŞ

Eğitim, bireyin doğumundan ölümüne kadar süregelen bir olgudur. Bu bakımdan eğitimi, kelimelerle tanımlamak pek de kolay değildir. Bireyi yaşadığı toplum içinde var eden değerlerin tamamını bu kapsamda incelemek gerekmektedir. Eğitim; kişisel yetenek, duygu, düşünce, gelenekler, inançlar ve bütün değerleri olgunlaştıran bir temayüz işlemidir.

Eğitim; ailede başlayan, okul öncesi eğitimden üniversite eğitiminin sonuna kadar ve devamında hayatın son bulduğu güne kadar kesintisiz devam eden bir süreçtir. Hangi bilim alanında yapılsın yapılsın, eğitim; sürekli var olan, gün geçtikçe önemi artan, teknolojik yenilikler ve iletişim olanakları ile gelişen ve değişen olduğunu hissettiren bir gerekliliktir.

Eğitimde değişim, başlı başına yönetilmesi gereken bir problemdir. Değişime uyum, eğitim dinamizminin vazgeçilmez bir unsurudur (Kaya, 2015, s. 349).

Değişim, bir sistemde ya da süreçte meydana gelen değişkenliğin toplamıdır. Bununla birlikte; değişim yavaş gelişen ve zaman içerisinde bir birikimin toplamı biçiminde ortaya çıkan farklılıktır.

Fransızların, “*Bir şey ne kadar değişirse o kadar aynı kalır*” deyişi, değişimin gerekliliğini ve kaçınılmazlığını ortaya koyan bir olgudur (Taşkın, 2001).

Değişim, doğal ve olması gereken bir şey ise, değişimi araştırmalar bakımından konu haline getiren neden, değişimin hangi hızda gerçekleştiğidir. Bilgisayar alanında meydana gelen değişim adeta baş döndürücüdür. Bilgisayar eğitiminin özellikle programlama boyutu bu hızlı değişimden ziyadesiyle etkilenebilmektedir. Öyle ise, eğitimi çeşitlendiren ve sınıf içi etkinlikleri ilginç hale getiren modelleri hayata geçirmek gerekmektedir.

Değişimi yakalamak ve değişimin gereklerini yerine getirmek; bireyleri, rakiplerine üstün hale getiren, onlara rekabet yapma gücü veren ve yeni istihdam alanları sağlayan avantajları da beraberinde getirmektedir (Peppard, 1993, s.1).

Türkiye’de eğitim alanında yapılan değişiklikler, içerik veya model özellikli değil, şekilseldir. Oysa değişimin, eğitim modelleri temelinde yapılması gerekir. Şekilsel değişiklikler sistemi bozmakta, düzensizlik sinyalleri ve kafa karışıklığı, herkesi ve kesimi rahatsız etmektedir. Bu değişikliklerle, eğitimin niteliği artmamakta, tam tersine öğrencilerin algı düzeylerindeki düşüş, araştırma alanlarının konusu haline gelmektedir. İstihdam sağlayan çevreler; nitelikli eleman gereksiniminden yakınıırken, üniversite mezunları iş bulamamaktan şikâyet etmekte, akademisyenler yetiştirebilecek nitelikte öğrenci bulamamanın sıkıntılarını dile getirmektedir. Birbiri ile çelişen bu gerçekler, eğitimde bazı şeylerin yolunda gitmediğinin veya bazı eğitim modellerinin değişmesi ya da güncellenmesi gerektiğinin bir göstergesi gibidir.

Bütün öğreticilerde gelişen ortak kanı, öğrencilerin bilgi düzeylerinin ve donanımlarının sürekli azalmakta olduğudur. Öğrencilerin öğrenme zamanının önemli bir kısmını internet başında sosyal medya ortamlarında geçirdiği ve bunun önemli bir tehdit haline geldiğini herkesin görmesi gerekir. Geleneksel ya da klasik öğretim yöntemleri ile öğrencilere bir şeyler öğretmek her geçen gün daha zor bir hale gelmektedir. Öğretim elemanının aktif, öğrencinin pasif konumunda bulunduğu klasik öğretim modelinde, öğrenciyi öğrenmeye güdüleyen çok ta fazla yaptırım bulunmamaktadır. Bu durumda geleneksel ya da klasik öğretim modelini yeniden tasarlayarak gereksinimleri karşılar bir düzeye çıkarmak veya diğer bazı modellerle desteklemek gerekir.

Kaya (2015) tarafından yapılan, Ege Üniversitesinin bütün meslek yüksekokulları ölçeğinde yapılan bir araştırma ile öğretim elemanları ve öğrencilere uygulanan bir alan araştırmasından elde edilen sonuçlar dikkate değerdir: Öğrencilere sorulan, “bir öğrenci olarak birinci önceliğiniz olması gereken ders çalışmak, sizin açınızdan hangi sırada yapılan bir aktivitedir?” Sorusuna verilen yanıtlar şu şekilde bulunmuştur:

- %23,86 gezmek dolaşmak,
- %23,80 sosyal medya kullanmak,
- %20,30 müzik dinlemek,
- %9,64 kitap okumak,
- %8,12 diğer eğlenme faaliyetleri-ne katılmak,
- %5,08 ders çalışmak (Kaya, 2015, s.359).

Bu sonuçlar; vahim, bir o kadar da dürüstçe verilmiş yanıtlardır. Bu kötü durumun ortaya çıkmasına sebep olan çok sayıda ders dışı aktivite bulunmaktadır. Zira her geçen gün gelişmekte olan internet ve iletişim olanakları ders dışı öğrenme zamanını ortadan kaldıran ilginçlikler sunmaktadır. Öğretenin başrolde olduğu klasik öğretim modelinde, öğrenci ya dersi dinlemeye gelmemekte, derse gelmiş ise; akıllı telefonu ile meşgul olmakta ya da arkadaşları ile sohbetler yapmaktadır. Pasif öğrenme konumunda bulunan öğrenciyi, öğrenmeye mecbur bırakan bir yaklaşım bulunmadığından öğrenci, sınıfta öğrenme çabası içerisine girmemektedir. Ders dışı zamanının büyük bir kısmını ders çalışmadan harcayan öğrenciler için ders zamanını verimli hale getirmek gerekir. Bunun için öğrencileri öğrenme dinamiğinin merkezine çekmek ve öğrenme faaliyetini büyük oranda sınıf ortamında tamamlamak gerekir. Bunun için de klasik model dışında çağın ve olağan dışı değişimin gereksinimlerine uygun alternatif öğrenme modellerine ihtiyaç bulunmaktadır. Öğrenmede ve sınıf ortamında öğrenciyi pasif konumdan aktif konuma geçiren, değişim paradigmasına uygun, öğrenmeyi öğreten eğitim modelinin probleme dayalı öğretim (PDÖ) modeli olduğu değerlendirilmektedir.

Probleme Dayalı Öğrenme (PDÖ), “*yaparak ve yaşayarak öğrenme*” temeli üzerine inşa edilen ve ilk kez 1960’lı yıllarda Kanada’da Mc Master Üniversitesinde Howard Borrows tarafından tıp eğitimi amacı ile kullanılan bir modeldir (Barrows & Tamblyn, 1980).

Bir Çin atasözü; “*Bana söylediğini unuturum, gösterdiğini hatırlarım, yaptırdığını öğrenirim*” der (Boran&Aslaner, 2008).

PDÖ modeli, tıpkı Çin atasözünü destekler nitelikte model olup, öğrenenin kendisine yaptırıp öğretildiği bir modeldir. Ayrıca, PDÖ, öğrencilere öğrenme becerisi kazandırmayı ve öğrenme kapasite-lerini artırmayı amaçlayan bir yaklaşımdır (Yaman&Yalçın, 2010).

Bununla beraber PDÖ yaklaşımı, öğrencileri önceki deneyim ve bilgilerinden yararlanarak karşılaştıkları yeni durumlara anlam verdikleri ve özümlediklerini savunan yapılandırmacı öğrenme-nin, eğitimde kullanımına yönelik geliştirilen modellerden biridir (Ülger&İmer, 2013).

PDÖ, öğrenenlerin öğrenme sürecine katılımına ve öğrenmeye sağladığı farklı bakış açısı ile geleneksel eğitim programlarının değişimini de beraberinde sağlamaktadır.

Farklı bakış açıları ile PDÖ, karmaşık ve gerçek hayat problemlerinin araştırılması ve çözümünü etrafında organize edilmiş bireylerin güç, zihin ve de beceri yönünden aktif katılımlarını gerektiren, tecrübeye dayalı öğrenmeyi temsil eder (Demirel&Turan, 2010).

Bu açıklamaları da fazlasıyla destekleyen Boran ve Aslaner (2008)’in makale çalışmalarında vurguladıkları bir hikâye çok önemli ve dikkate değerdir:

“*Bir genç, babasıyla kırlarda dolaşırken kozasından çıkmaya çabalayan bir kelebek görür. Kelebek, kozanın lifleri arasından sıyrılmaya çabalamaktadır. Baba, hemen kelebeğin yardımına koşar ve dikkatli bir şekilde kozanın liflerini sıyırtırır. Lifleri açar ve kelebeğin pek zorlanmadan kozadan çıkmasını sağlar. Ancak kelebek kozadan kolaylıkla çıkmasına rağmen uçamaz. Biraz çırpınır ve yerinde kalır. Çünkü kelebek kendini liflerden kurtarma çabası sırasında aslında kaslarını geliştirmekte, kendini ayakta tutacak, güçlü kılacak, uçmaya hazırlayacak hareketleri öğrenmektedir. Baba, lifleri sıyırmakla kelebeğe iyilik yapmamıştır. Kelebeğin güçlenmesine engel olmuştur. Kelebek hiçbir zaman özgürlüğü tadamamış, gerçekten uçamamıştır*”.

Bu hikâyeden çıkarılması gereken sonuç şudur: Biri bir şey öğrenirken; uğraşmalı, didinmeli, yorulmalı belki zaman zaman üzülmeli ancak mutlaka kendi başına öğrenmelidir. Zira kendi kendine

öğrenilen bilgi, öğrenene aittir ve tam öğrenme de budur. Böylece, yukarıda hikâyesi verilen kelebek gibi gerçek hayatın zorluklarına uçabilmek mümkün hale gelecektir. İşte PDÖ, böylesine zor ve zahmetli bir süreci zorunlu kıldığından gerek öğretim elemanları gerek öğrenciler cephesinde pek de kabul görmemektedir. Çünkü uzun bir süreden beri kullanılan klasik model, gerek öğretici gerekse öğrenci cephesinde bir alışkanlık ve kolaylık haline gelmiştir. Öğrenci bakımından değişim, yeni zorluklarla karşılaşmak ve bir dizi külfete adapte olmak demektir. Bu durum, birçok kimse bakımından tercih edilir bir yol değildir. Çünkü değişime adapte olmak kolay değildir. Öğretim elemanları tarafında da benzer zahmetlerin bulunması nedeniyle değişime taraf olmak, kabul görmemektedir.

Önemli bir eğitim araştırmacısı Nancy Law (2011)'in değişim hakkında görüşleri dikkate değerdir. Law şöyle demektedir: Değişimi kabul ettirmek çok zordur. Değişim karşısında ciddi direnişlerle karşılaşılır, buna rağmen değişimin sistematik modelini hayata geçirmek gerekir. Bunun için; Öncü olmak, yenilik yapmak, uygulamak, kurumsallaştırmak ve yükseltmek gerekmektedir (Law, 2011).

Değişim, kaçınılmaz ve herkesin yüzleşmek durumunda olduğu bir gerçek olduğuna göre, değişimle yüzleşmek gerekir. Değişim karşısında ortaya çıkan amansız direnci kırmanın yollarından biri, güçlü bir iletişim ve ikna etme yeteneğidir (Kotter&Schlesinger, s.70-72).

Değişimi savunmak, sağlamak ve öncüsü olmak kolay bir iş değildir. Kişisel gelişim uzmanı, Dr. Gary McKay (2010), yazmış olduğu kitabın başlığını, "*Cesur ol ve değişime ayak uydur*" biçiminde belirlemiştir. Çağımızda artık sıklıkla ortaya çıkan değişime ayak uydurmak için cesur, özgüveni yüksek, donanımlı ve vizyon sahibi mezunlar yetiştirmek gerekmektedir.

Programlama eğitime bir yaklaşım olması açısından PDÖ, uzun yıllardır benimsenen nesnelci (objectivist) yaklaşıma bir alternatif olarak karşımıza çıkan yapısalcı (constructivist) bir perspektif ve yeni bir öğrenme paradigması oluşturmuştur (Alper&Deryakulu, 2008).

Programlama eğitimi, öğrencinin çok yönlü düşünmesini gerektiren zor, zahmetli bir o kadar da zevkli bir örüntü kurgulama sanatıdır. Bir bilim alanı olmasına rağmen programlama işinin adeta bir sanat alanı olduğunu kabul etmek gerekir. Çünkü bazı öğrenciler, hiç çalışmadan bu sanatı çok iyi icra edebilirken, bazı öğrenciler çok çalışmalarına rağmen, bu mesleğin iyi birer icracıları durumuna yükselmediklerini görmek mümkün olmuştur. Bu özelliği gereği, programlama eğitim sürecinin diğer eğitim süreçlerinden farklı olduğunu kabul etmek gerekmektedir.

Edens (2000)'e göre de PDÖ yöntemi, öğrencinin çok yönlü düşünmesine ve problem çözme yeteneğine yardımcı olmayı amaçlayan bir model olduğuna göre, programlama eğitimi ile PDÖ modeli arasında bir benzerlik bulunduğunu göz ardı etmemek gerekir.

Birçok araştırmacı, yapısalcı yaklaşımın daha iyi öğrenmeye ve daha iyi öğretim yazılımları oluşturmaya önderlik edeceğini beklemektedir. Bilgisayar destekli ortamların yapısalcı yaklaşımın ilkelini desteklemesi nedeniyle araştırmacı ve eğitimciler yapısalcı yaklaşımı desteklemeye başlamışlardır (Alper&Deryakulu, 2008).

1.1. PDÖ Yöntemi

PDÖ bir senaryo temelinde saptanan ders bölümünün nasıl öğrenileceği konusunda başlatılan çalışma sürecinde önceki bilgilerin kullanılması ve öğrencilerin gereksinim duyduğu konuların belirlenmesi, öğrenilmesi ve tartışılması temeline dayanan aktif bir eğitim yöntemidir. Bu yöntem bir eğitim yönlendiricisi ile 6-8 öğrenciden oluşan küçük gruplara uygulanmaktadır.

PDÖ oturumlarında eğitim, öğrenci merkezlidir. Eğitim yönlendiricisi bilgi aktarımında bulunmayan ancak senaryo hazırlayan ve öğrenmeyi kolaylaştıran bir danışman rolündedir. Çalışmanın temelini oluşturan problemler, gerçek yaşamda karşılaşılabilen olgu veya durumları içermektedir. Öğrenciler, problemi önceki bilgileri kullanarak çözümlemeye çalıştıklarında bilgi sınırlarına ulaşılmakta ve grup tartışmasında açıklanamayan sorular ortaya çıkmakta ve öğrenme hedefleri bu şekilde belirlenmektedir (Musall&ark, 2002).

Tıp eğitimi gibi karmaşık ve sürekli dönüt gerektiren öğretim süreçlerinde PDÖ modelini uygulamak çok daha farklı olabilmektedir. Bu tür süreçlerde PDÖ modelini sürekli iyileştiren yaklaşımlar kullanılabilmektedir. PDÖ modeline katkı, zaman zaman öğrencilerin eleştiri ve katkıları ile de zenginleşebilmektedir. Öğrencilerin eğitim modeline ilgisi, yaklaşımı, iş birliği arzusu ve öğrencinin algı düzeyi de katkı sunan diğer unsurlardır.

PDÖ oturumları aşağıdaki dinamiklerden oluşmaktadır:

1. Oturum öncesi hazırlık,
2. Başlangıç aşaması,
3. Senaryonun dağıtılması, okunması ve bilinmeyenlerin açıklanması,
4. Sorunların belirlenmesi,
5. Hipotezlerin beyin fırtınası (Brain storming) yöntemi ile listelenmesi,
6. Hipotezlerin açıklanması ve tartışılması,
7. Senaryonun daraltılması,
8. Öğrenme konularının saptanması,
9. Öğrenme süreci,
10. Geri bildirim.

1.1.1. PDÖ'de Senaryo

PDÖ'de senaryo, temel eğitim amacını gerçek yaşamla uyumlu hale getiren, sorunların belli bir sistematik içerisinde sunulduğu olgular biçiminde hazırlanır. PDÖ'de başarılı senaryo hazırlamak çok önemlidir. Senaryo sürecinde öğrencinin problemi keşfetme temeline dayanan bir örüntünün kurgulanması amaçlanır. Böylece, öğrenilecek konu ile öğrenci arasında, beceriyi aynı anda geliştiren, bilgiyi sentez ve analiz eden, kendi kendine sürekli öğrenme yetisi kazanan bir dinamik geliştirilmiş olur. Böylece öğrenci, öğrenmede pasif konumdan aktif konuma geçer. Senaryo geliştirme sürecinde bütün öğretim elemanları, fikir paylaşımı yaparlar, işbirlikçi çalışma ön plandadır. Bilim alanına ve öğretilecek konunun niteliğine göre, senaryo basılı, sözel, sahne ortamında ya da video çekimi biçiminde geliştirilebilir.

1.2. Programlama Eğitimi

Programlama eğitimi; gerek öğretim elemanları, gerek öğrenciler bakımından önemli güçlükler içeren bir eğitim sürecidir (Paula, Ambrosio ve Costa, 2010).

Programlamada PDÖ yaklaşımı, öğrencileri sosyalleştiren ve onları daha etkili öğrenmeye sevk eden ve fayda sağlayan, öğrencilerin birer aktif katılımcı oldukları bir yaklaşımdır (Kinnunen & Malmi, 2005).

Programlamada PDÖ yaklaşımı, öğrencinin öğrenmenin odağında bulunduğu bir yaklaşımdır. Bir ekip içerisinde öğrenme çabası, öğrenmeyi etkinleştiren, güçlendiren ve kalıcı kılan bir yöntemdir (Nuutila & ark., 2008).

PDÖ modelinin, bütün öğretim süreçleri için öğretim elemanı ve öğrenci bakımından zengin bir çevre ortamı sunduğu kabul edilmektedir. Bu nedenle değişen eğitim koşullarının eşanlı bir biçimde karşılandığı ve eğitim bileşeni olarak kabul edilen; dersin planlanması, uygulanması, yönetilmesi ve değerlendirilmesinin eşanlı yapıldığı bir yaklaşımdır (Grabinger, Dunlap ve Duffield, 1997).

Programlama eğitiminde; proje çalışmalarında gayret sarf eden, ödevlerini eksiksiz yapan, sürekli yazılım üretme gayreti içerisinde olan, öğretim elemanını bir danışman ve yardım alınan bir referans gibi kullanan öğrencilerin başarılı oldukları görülmektedir.

Türk yükseköğretim sisteminde programlama eğitimi; ön lisans düzeyinde bilgisayar programcılığı, lisans düzeyinde yazılım mühendisliği veya bilgisayar mühendisliği adı altında yapılmaktadır. Lisans eğitimi, mühendislik fakülteleri bünyesinde, ön lisans eğitimi meslek

yüksekokulları bünyesinde yapılmaktadır. Bununla beraber, işletme ve idari bilimler fakültesi bölümleri olarak, yönetim bilişim sistemleri veya işletme enformatiği, fen fakültelerinde bölüm olarak, bilgisayar bilimleri adı altında lisans eğitimleri verilmektedir. Gerek lisans, gerek ön lisans bütün bölümlerinin temel amacı, iyi bilgisayar programcıları yetiştirmektir. Bu amacı gerçekleştirmek adına müfredatın önemli bir kısmını da programlama dersleri oluşturmaktadır.

Lisans ve ön lisans düzeyinde geleneksel yöntemle yapılan eğitimlerde ciddi sorunlar bulunduğu, öğrencilerin başarılı birer yazılımcı olamadıkları ve bu türden eğitim sorunlarının her geçen gün daha da büyüdüğü sıklıkla dile getirilmektedir. Bu sorunlar, öğrencinin niteliğinden kaynaklı olabileceği gibi, programlama eğitiminin gereksinim duyduğu zor ve zahmetli mantık oluşturma dinamiğinin yoğunluğu ve teknolojik araçlarının çok hızlı değişimden de kaynaklanabilmektedir. Bununla beraber, bilgisayar eğitiminin gerek programlama veya yazılım, gerek donanım boyutunda meydana gelen gereksinimlerin bir süreklilik arz etmesi nedeniyle başarılı bir değişim yönetiminin hayata geçirilemediği değerlendirilmektedir.

Nitekim Microsoft yazılım firmasının sahibi Bill Gates'in, 1999 yılında kaleme aldığı ve piyasaya sürdüğü "The speed of thought" başlıklı kitabında, bilgisayar alanında en büyük riskin, bu alanda meydana gelecek değişiminin yönetilememesi olduğunu vurgulamıştır (Gates, 1999).

Sektörde meydana gelen bu değişim, yazılım araç ve materyallerinin dolayısıyla eğitim müfredatının çok kısa sayılabilecek sürelerde güncellenmesi gibi zorunlulukları ortaya çıkardığı görülmektedir. Ortalama iki yılda bir yenilenen işletim sistemi platformları, daha güncel yazılım araçlarının öğrenilmesini zorunlu hale getirmektedir. Bu kadar kısa sürede, zor ve zahmetli yazılım araçlarını öğrenmek ve öğretmek için çok çalışmak bir zorunluluk haline gelmiştir. Bu durumda bilişim eğitiminde "öğrenmeyi öğretmek" paradigmasının bir felsefe haline getirilmesi gerekmektedir. Kendi kendine öğrenme adı da verilen bu felsefe, bir bakıma PDÖ yöntemi ile özdeşleşen bir anlayışı ortaya koymaktadır. Bu bakımdan PDÖ modelinin programlama eğitiminde daha etkili olabileceği değerlendirilmektedir.

Nitekim Aydede ve Kesercioğlu (2012) tarafından yapılan bir araştırma, aktif öğrenme uygulamalarının kendi kendine öğrenme becerilerini olumlu etkileyen bir araç olduğunu ortaya koymuştur. Zira bu yeteneği kazanan bilişim meslek elemanları, iş yaşamlarında da sürekli öğrenme gereksinimlerini karşılayabilecek nitelikleri kazanabilmektedir.

Akpınar ve Altun (2014)'e göre, bilgi toplumu olarak kabul edilen günümüz toplumlarında, programlama eğitimi de bir zorunluluk haline geldiğine göre, PDÖ yönteminin her öğretim modelinin alternatifi veya yardımcı eğitim modeli haline geldiği değerlendirilmektedir.

Teknolojik değişim ve gerekliliklerin, başta telefonlar olmak üzere, bütün akıllı cihazların her yaşta bütün bireyleri, bilişim okuryazarı haline getirdiği bu dönemde, PDÖ yönteminin bilişim eğitiminde alternatif bir araç haline getirilmemesi bir eksiklik gibi değerlendirilmektedir. Çünkü toplumda kendi kendine öğrenme dinamiği, bir zorlama olmadan kendiliğinden gelişmektedir.

Dünya'da en çok sosyal medyanın ve internetin kullanıldığı bir ülke olmak, birçok bakımdan ciddi olumsuzluklar içerebilir. Ancak, neredeyse bağımlılık haline gelen bu kullanım düzeyinin eğitim tarafında bazı avantajları beraberinde getirdiğini görmek ve bu durumu avantaja dönüştürmek gerekmektedir.

İnternetin öğrenme açısından çok önemli avantajlar sunması, özellikle youtube video platformunun sayısız öğrenme aracını kişilerin kullanımına sunması göz ardı edilmemesi gereken bir faydadır. Esasında yaşlısı genci, bütün bireylerin birçok uygulamayı kendi kendine öğrenebilmesi, PDÖ eğitim modelini destekleyen bir unsur olarak değerlendirilebilir.

Öğrencilerin öğrenme ortamlarına gelirken zihinlerinde bulunan bilgileri çeşitli edinimlerle ve yaşadıkları çevreden edindikleri bilinmektedir. Öğrenme ortamına bu bilgilerle gelen öğrencilerin bu türden bilgi edinimleri bir nebze yanlışlık ta içerebilmektedir (Toros & Yeşiltaş, 2015, s.159).

Oysa bir bilgiye erişirken çeşitli kaynaklara başvurmak gerekir. Bu sayede bilgiyi farklı açılardan ele alabilmek, çeşitli yöntemlerle elde edilen bilgileri düzenlemek aslında düşüncüyü geliştiren ve düşüncenin tutarlılığını kontrol eden bir yaklaşımdır (Körükçü & Şengül, 2018, s.4). Bu yaklaşımı sunan aracın internet olduğunu kabul etmek gerekir.

Bilişim dolayısıyla programlama teknolojilerinin çok hızlı bir değişim içerisinde olması, bilişim eğitiminin ve meslek elemanlarının dinamik bir süreç içerisinde sürekli öğrenmelerini zorunlu kıldığından bu kişilerin bilgi depolayan değil, bilgi üreten bireyler olmaları zorunludur (Günhan&Başer, 2008).

Çünkü bilişimcilerin depolayacakları bilgiler kısa bir zaman sonra, güncelliğini yitirmiş bilgiler haline gelmektedir. Bu bakımdan bilişimcilerin yakından bildiği atak (çevik) programlama (agile programming) mantığına uygun olarak, durmadan, ara vermeden yeni yazılım modelleri öğrenmek ve kimsenin uygulamasına fırsat vermeden uygulamak, ekonomik kazanımlar elde etmek, temel hedeftir.

Görüldüğü üzere, bilişimcinin bilgi edinimi ile matematikçinin, fizikçinin, kimyacının ve tarihçinin bilgi edinim felsefesi tamamen farklıdır. Meselâ, tarihçi öğren ve biriktir mantığından hareket eder, fizikçi ya da kimyacı, öğrenir, öğrendiğini laboratuvar ortamında uygular, bilgiyi paylaşır. Oysa programcı öğrenir, uygular, uyguladığını satar ve yenileri öğrenmek üzere eskileri bir kenara bırakır. Programcı teknolojik gelişmelere paralel olarak sürekli ve kendi başına öğrenmek zorundadır. Programcı açısından başarının kaynağı budur. Bu felsefe de PDÖ yöntemine çok uygun bir gerekliliktir.

1.3. C# Programlama Dilinde Metot Tanımı

C# programlama dilinde yazılım geliştiren uygulamacıların bildiği üzere, metot, programın herhangi bir yerinde kullanılmak üzere bir görevi yerine getiren tasarlanmış altprogramlara verilen addır. Metotlar tek başlarına çalıştırılabilen yapılar değildir. Metotlar, çağrıldıkları program ortamında çalışabilen, parametre aktaran, görevini tamamladıktan sonra, tanımlı bulunduğu bölgeye çekilen, programlama düzeninin yapısal ve anlaşılır olmasını sağlayan yazılım parçacığı ve aracıdır (Algan, 2010).

Diğer yazılım fonksiyonlarında olduğu gibi metotların da mutlak anlamda uygulama yapılarak öğrenilmesi ve yazılım desenlerinin bir boyutu haline getirilmesi gerekir. Bunun için de öğrencinin sıklıkla ve çok sayıda uygulama yapmasına gereksinim vardır.

1.4. C# Metot Kullanımı

C# programlama dilinde metot kullanımı, yazılımı basitleştirmek, etkin hale getirmek ve yapısal özellik kazanmasını sağlamak amacıyla kullanılan alt program parçacıklarıdır. Metot geliştirmek ve kullanmak, nitelikli program kullanmanın ön koşullarındandır. Bu bakımdan bilgisayar mühendisliği ve programcılığı eğitimi alan öğrencilerin mutlaka öğrenmeleri gereken bir konudur.

```
Using System;
Class Metotlar;
{
int Topla(int a, int b)
{
return a+b;
}
Staticvoid Main()
{
Metotlar nesne=newMetotlar();
```



```
int a=nesne.Topla(2, 5);
Console.Write(a);
}
}
```

1.5. C#'ta Metot Kavramının Senaryosu

Uluslararası bir lojistik firması, yurtiçi ve yurtdışı ofislerinde kullanmak üzere bir otomasyon yazılımı satın almak istemektedir. Yazılımın muhtemel bazı güncelleme işlemlerine fırsat vermesi için güncel C# programlama araçları ile geliştirilmesi istenmektedir.

Firma, kurulu donanım birimlerini kullanacağından geliştirilecek yazılımın ekstra bir donanım (yeni bir bilgisayar veya hard disk, işlemci, ana bellek) maliyeti doğurmasını istememektedir.

Firma, geliştirilecek yazılımın uluslararası kod yazılım standartlarını uygun yapılmasını ve yazılım raporlama işleminin de aynı standartlara uygun olmasını talep etmektedir.

Firmanın kurulu donanım kapasitesi, çok üst düzeyde değildir. Bu bakımdan yazılımın bir ana bloğa bağlı birbirini çağıran küçük ancak hızlı alt programlar biçiminde geliştirilmesi gerekir. Bellek tasarrufu nedeniyle görevini tamamlayan alt programların değişken ve parametreleri işlevsiz bırakılacaktır. Farklı bloklardan program parçacıklarının çağırılması da mümkün olmalıdır.

Yazılımın değişik bloklarında aynı alt programların işlem yapması mümkün hale getirilmelidir.

Alt programların recursive (birbirini çağıran) özellikte geliştirilmesi gerekmektedir.

Bu maksatla firmanın yazılım talebine uygun bir şekilde bir yazılım bloğu oluşturunuz ve firmanın bütün işlemlerinin alt programlar vasıtasıyla geliştirildiği C# araçlarının ne(ler) olduğunu tespit ederek bütün davranışları ile birlikte öğreniniz. (Doğru ilerlemek maksadıyla sıklıkla öğretim elemanı danışmanlığına başvurunuz.)

2. YÖNTEM

2.1 Araştırma Deseni

Araştırma için Ege Üniversitesi Tire Kutsan Meslek Yüksekokulu Bilgisayar programcılığı bölümünde gündüz ve ikinci öğretim öğrencilerinden oluşan ve Görsel Programlama-1 adı altında C# dersini alan 105 öğrenci bir amfide toplandı. PDÖ modeli ilgili tanıtım yapıldı. PDÖ tanıtımında; öğretimin felsefesi, öğrenme sürecinin adımları, öğrenme materyallerinin nasıl kullanılması gerektiği, öğrenciden nasıl faydalanılması gerektiği, grup çalışmasının nasıl olacağı gibi tanıtımlar yapıldı. PDÖ grubu öğrencilere iki hafta süren bir kurs faaliyeti ile modeli özümsemeleri sağlandı.

Bundan sonra, sınıfta bulunan 105 öğrenciden devam durumu güçlü, 100 öğrenci seçildi. Tamamen rastgele olmak üzere bir veri tabanına aktarılan öğrenciler, C# programlama yazılımının Random fonksiyonu kullanılarak 50'şer kişilik iki gruba ayrıldı.

Birinci grup öğrencilere C# metotlar konusu geleneksel yöntemlerle öğretim elemanı tarafından öğretildi ve bu gruba kontrol grubu adı verildi. İkinci grup öğrenciler, metotlar konusunu PDÖ yaklaşımı çerçevesinde aynı öğretim elemanının yönlendirmesi ve danışmanlığında kendileri öğrendiler. PDÖ öğretim felsefesinin gereği olarak; 45 kişilik test grubu için 9 kişilik 5 grup oluşturuldu. Her grubun bir U masası etrafında toplandığı bir düzenek oluşturuldu. Birinci grup öğrenciler için öğretmen rolünü üstlenen öğretim elemanı, ikinci grup öğrenciler için sadece yönlendirme ve danışmanlık görevi üstlendi. Öğrenciler danışmanlık dışında öğrenmeyi destekleyici kitap, dergi, video araçlarını ve visual studio 2010 derleyicisinin yardım (help) araçlarını yoğun bir şekilde kullandılar.

Bu makalenin yazarı, Dokuz Eylül Üniversitesi Fen Fakültesi, İstatistik bölümünde çok sayıda senaryo oturumuna katılmış olduğundan iki öğretim modelini de uygulama şansına sahip oldu. Böylece, iki farklı modelin aynı öğretim elemanı tarafından uygulanması ile ortaya çıkması muhtemel öğrenciler arası öğretim farkı da ortadan kalkmış oldu.

2.2 Çalışma Grubu

Araştırmamızın çalışma grubu, tamamen rastgele olmak üzere bir veri tabanına aktarılan öğrencilerden oluşturuldu. Öğrenciler, 50'şer kişilik iki gruba ayrıldı. Birinci grup öğrencilere C# metotlar konusu geleneksel yöntemlerle laboratuvar ortamında bilgisayarlar başında öğretim elemanı tarafından öğretildi ve bu gruba kontrol grubu adı verildi. İkinci grup öğrenciler, metotlar konusunu PDÖ yaklaşımı çerçevesinde aynı öğretim elemanı danışmanlığında destekleyici kaynaklar ve senaryo dâhilinde kendileri öğrendiler. Bu grup ta test grubu olarak isimlendirildi.

2.3 Verilerin Toplanması

Verilerin toplanması şu şekilde gerçekleştirildi: C# programlama dili bakımından oldukça önemli bir işlevi bulunan metotlar konusu iki farklı gruba, iki farklı yöntemle öğretildi. Öğrencilerin metotları ne şekilde öğrendiklerini test etmek amacıyla, iki grup öğrenciye metotlar konusundan 5'er soruluk aynı sınav (Bkz. (2.4)) eşzamanlı uygulandı. Sınav kâğıtları öğretici öğretim elemanının bulunmadığı, meslekte uzman üç öğretim elemanı (1 Prof.Dr., 1 Dr. Öğr. Üyesi., 1 Öğr.Gör.Dr.) tarafından 100 üzerinden değerlendirildi.

Üç öğretim elemanının verdiği puanların aritmetik ortalaması alınarak, öğrencilerin metotlar konusunda aldıkları puanlar belirlendi, virgülden sonraki değerler yuvarlandı. Gruplar, 50'şer kişilik öğrenci gruplarından oluşturulmalarına rağmen; birinci gruptan 47, ikinci gruptan 45 öğrencinin sınava katılımı sağlanabildi. 8 öğrenci çeşitli mazeretlerle sınava katılmadı. Bu öğrenciler sonradan sınava alınmalarına rağmen yanlışlık oluşturma olasılıkları göz önünde bulundurularak değerlendirme dışında tutuldu.

2.4 Sınavda Sorulan Sorular

1. Metotlar ne işe yarar? Neden kullanılır? Programlama etkinliği bakımından sunduğu avantajlar nelerdir? Yazınız.
2. Ekrandan okutulan iki adet integer (tam sayı) sayıyı toplayan ve sonucu Main Blok içerisine çağıran ve de ekrana yazan Metot yazılımı nasıl yapılır? Gösteriniz.
3. Metotlar içerisinde tanımlı değişkenler nasıl bir çalışma davranışı sergiler? Metot'un çalışması sona erdiğinde değişkenlerin değeri ne olur? Bilgi veriniz.
4. Parametre ve değişken değerlerini kendiniz atayarak geçerli ve geçersiz iki metot yazılımı yapınız.
5. Metotlarla tür dönüşümü nasıl yapılır? Gösteriniz.

3. BULGULAR

3.1 Verileri Analizi

Birinci gruptan 47, ikinci gruptan 45 öğrencinin katılımı ile elde edilen 100 üzerinden notlar, iki ortalamaların farkı testine tabi tutuldu. Örnek genişlikleri 30'dan büyük olduğu için t testi yerine büyük örnek Z testi uygulandı. Esasında 30'dan büyük serbestlik dereceleri için t ve Z tablo değerleri birbirlerine çok yakın, örnek büyüklüğü artıkça eşit değerlerdir. Test için öngörülen yanlışma payı geleneksel olmak üzere % 5 ($\alpha=0.05$) olarak belirlendi.

Geleneksel öğretim yöntemi ile öğretim yapılan öğrencilerin C# programlama dilinin metotlar konusu sınavından almış oldukları puanlar şu şekilde elde edilmiştir.

Kontrol grubu öğrencilerinin 100 üzerinden aldıkları notlar:

34, 16, 75, 27, 18, 07, 21, 54, 41, 21, 10, 04, 79, 44, 82, 21, 11, 23, 55, 20, 32, 24, 80, 12, 16, 19, 55, 02, 11, 07, 02, 60, 54, 29, 20, 26, 50, 23, 19, 21, 32, 56, 34, 16, 71, 74, 13.

PDÖ öğretim yöntemi ile eğitilen öğrencilerin C# programlama dilinin metotlar konusu sınavından almış oldukları puanlar şu şekilde elde edilmiştir.

Test grubu öğrencilerinin 100 üze-rinden aldıkları notlar:

52, 29, 09, 75, 68, 67, 24, 90, 57, 66, 35, 38, 21, 63, 80, 82, 43, 55, 29, 95, 62, 08, 92, 21, 18, 27, 69, 66, 52, 43, 39, 89, 76, 25, 58, 48, 59, 66, 29, 26, 64, 65, 50, 61, 02.

Kurulan hipotez şu şekildedir:

A: Sıfır hipotezi: Kontrol grubu ile test grubu arasında fark yoktur.

B: Alternatif Hipotez: Kontrol grubu ile test grubu arasında fark vardır.

Belirtilen hipotezleri test edebilmek için varyansların homojenliği testi yapılmıştır. Test işlemi sonucunda varyanslar arası fark önemsiz bulunmuştur. Dolayısıyla varyanslar homojen olduğu sonucuna ulaşılmıştır. Test sonuçları Tablo 1’de verilmiştir. Varyanslar homojen olduğundan ortalamalar arası fark testi kullanılarak Tablo 2 sonuçları elde edilmiştir.

Tablo 2’de belirtildiği gibi, $3.77 > Z = -1.96$ olduğundan ortalamalar arası fark önemli bulunmuştur. Buna göre, PDÖ temelli öğrenmeden elde edilen sınav puanlarının klasik temelli

Tablo 1.Varyansların homojenliği testi

Başlıklar	N	Varyans	Hata Ort.	F Testi
Kontrol Grubu (Klasik)	47	530.84	0.0073	1.10
Test Grubu (PDÖ)	45	584.19	0.0075	

modele göre daha yüksek bulunduğu sonucuna ulaşılmış, dolayısıyla sıfır hipotezi ret edilerek, alternatif hipotez kabul edilmiştir. Böylece öğretim modelleri arasındaki farkın önemli bulunduğu görülmüştür. Ayrıntılı değerler, Tablo-2’de verilmiştir.

Tablo 2.Ortalamalar arası fark tablosu

Başlıklar	N	Ortalama	St. Sapma	Hata Ort.	Z Testi
Kontrol Grubu (Klasik)	47	32.36	23.04	0.0073	-3.77*
Test Grubu (PDÖ)	45	50.96	24.17	0.0075	

Görüldüğü üzere test grubu, yani PDÖ yaklaşımı ile C# programlama dilinde metotlar yaklaşımını öğrenen öğrencilerin sınavdan elde ettikleri başarı farklıdır. Kontrol grubu olarak isimlendirilen ve geleneksel öğretim araçları ile konuyu öğrenen öğrencilerin sınav ortalaması 32.36 olurken, test grubu olarak isimlendirilen ve PDÖ yaklaşımı ile konuyu öğrenen öğrencilerin sınav ortalamaları 50.96 düzeyinde gerçekleşmiştir. Bu sonuçlara göre test grubunun kontrol grubuna göre daha başarılı olduğu, ortalamalar arası fark testi ile doğrulanmış bulunmaktadır.

3.2. Araştırma Sürecinde Yapılan Tespitler

Gruplara ayrılan öğrencilerden kontrol grubu, geleneksel öğretim yöntemi ile konuyu öğrenmeye çalıştı. İki hafta devam eden ders öğrenme sürecinde derslere devam etmeyen öğrencilerin kontrol grubunda daha yüksek oranlarda gerçekleştiği görüldü. Kontrol grubunda öğrencileri öğrenmeye motive etmek için daha fazla çaba gösterildi. Sınav sürecinde sınava ayrılan sürenin büyük oranlarda kullanılması yönünde telkinler yapıldı.

PDÖ modeli ile konuyu öğrenmek durumunda olan test grubunda yardımlaşmanın daha fazla olduğu, konuyu öğrenmek için ekstra bir çabanın gösterildiği gözlemlendi. Test grubu, PDÖ oturumlarına daha yüksek oranlarda katılım sağladı. Öğrenme sürecinde yardımlaşmanın ve grup çalışmasının üst düzeylerde olması, çok başarılı olmayan öğrencilerin daha fazla motive olmasını sağladı.

Ders süreci sonucunda kontrol grubu, geleneksel yöntemle konuyu öğrendiği için ekstra bir dönüt sağlamadı. Ancak test grubundan, öğrenileni teyit etmek maksadıyla nispeten daha fazla dönüt sağlandı.

PDÖ modeli ile öğrenen grup, birlikte bir şeyler öğrenmenin bir kolaylık sağladığını, bunun da tıpkı eşle programlama uygulamalarında görülen yararlarla benzer kazanımlara sahip olunduğunu dile getirdiler. Bir de test grubunun her ders oturumunda sonra küçük bir sınava tabi tutulmasının ekstra motivasyon sağladığı değerlendirilmektedir.

4. TARTIŞMA VE SONUÇ

Programlama eğitim süreci, bütün öğretim süreçlerinden çok daha farklıdır. Öğrencinin kendi kendini yetiştirmesi ve çok çalışması esastır. Zira, programcılık alanlarından mezun elemanlar, iş piyasasında çok çalışmak, sürekli yenilenmek, bazı programlama araç ve modüllerini kendi başlarına öğrenmek ve bir karmaşık proje çalışmalarını tamamlama yeteneğine sahip olmak zorundadır.

Birçok üniversite öğrencisinin akademide edindiği bilgiyi iş yaşamında kullanmadığı yakınmaları, bilişim alanı için geçerli değildir. Meslek elemanları üniversite yaşamlarında öğrendikleri hemen bütün bilgileri iş yaşamında kullanırlar. Hatta bu yetmez, çok daha fazlasını iş yaşamında öğrenmek durumunda kalırlar. Bu bakımdan öğretim elemanlarının çok başarılı ve doğru yönlendirme yapmaları ve öğrencilere, öğrenmeyi öğretmeleri gerekmektedir.

Bilgisayar programcılığı eğitiminde de öğrencinin kendi kendine öğrenebilmesi çok önemlidir. Klasik öğrenme yaklaşımları, öğrenciden çok öğretim elemanının kendi kendine öğrettiği ve katkı yaptığı süreçlerdir. Bir dersi anlatırken, dinleyenlerin (öğrencilerin) öğrenme zorunlulukları yoktur. Bu süreçte öğrencileri öğrenmeye mecbur bırakan bir yaklaşım da söz konusu değildir. Öğrenciyi öğrenmeye mecbur bırakan tek modelin PDÖ modeli olduğu düşünülmektedir.

Dersi öğrenme sürecinde bir şeyler öğrenmek ve eğitim danışmanı olan öğretim elemanına anında dönütler sağlamak gerekir. Bu dönütleri sağlamak ve grubunda bulunan öğrencilere katkı sunmak için ekstra gayretli olmak gerekmektedir. Bir de öğrenci, bulunduğu grup içinde en çok katkırı sunan öğrenci olma çabası içerisindedir. Yani PDÖ modeli, aynı zamanda grubu oluşturan öğrenciler arasında rekabeti kızıştıran ve daha çok öğrenmeye sevk eden bir anlayışı hâkim kılmaktadır.

Ayrıca, her dersten sonra bilgi sınavasının yapılması, öğrenciyi derse devam etmek ve bir şeyler öğrenmek zorunda bırakan bir faktördür. Bu yönü ile dersi devam etmemek, gerek öğretim elemanı veya öğretmen, gerekse öğrenci için yapılmaması düşünülen bir eylemdir. Oysa klasik öğretim sürecinde öğrencilerin bazen de öğretim elemanlarının keyfi devamsızlıkların bir parçası olduğunu görmek mümkün olabilmektedir.

Bütün bunlara rağmen, bazı bilim alanlarında klasik öğrenme süreci, alternatifsiz ve vazgeçilmez olabilir. Bu türden alanlarda klasik öğretim sürecini kullanmak gerekir. Ancak bilişim gibi, teknolojik gerçeklemelerin sıklıkla değişime uğradığı alanlarda, kendisini bilgi bakımından sürekli yenilemek durumunda olan meslek elemanlarına bu yeteneğin kazandırılması adına PDÖ modelini, hiç olmazsa yardımcı bir eğitim modeli olarak kullanmak gerekmektedir. Bu sayede, klasik modeli, sorgulamak ve varsa eksikliklerini tespit etmek çok daha kolay olabilir.

Bu çalışma kapsamında bilişim mesleğinin karakterine uygun sonuçların elde edildiğini söylemek mümkündür. Öğretim elemanının başrolde olduğu klasik öğretim modelinde öğrenmeye tabi tutulan kontrol grubu, PDÖ modelini esas alan test grubunun çok gerisinde kalmıştır. Öğretim elemanı,

kontrol grubunda öğretmenlik, test grubunda danışmanlık yapmıştır. Danışmanlık yaptığı PDÖ grubu, çok daha başarılı olmuştur. Çünkü PDÖ grubu kendi kendine öğrenmiş ve öğrendiğini çok daha somut vurgularla kâğıda yansıtmıştır. Başarı farkı büyüktür. Kontrol grubu %32.36 başarı ortalamasına sahip iken, test grubu %51'e yaklaşan bir başarı ortalaması tutturmuştur. İstatistik test sonuçları da başarı farkının önemli olduğunu göstermiştir.

Teknolojik araçların tutsağı ve bağımlısı durumuna gelmiş üniversite gençliğini, yeniden öğrenme platformlarına çekmek kolay olmamaktadır. Bunun için rekabeti ve ders içi ilginçlikleri ön plana çıkarmak gerekmektedir. Bu anlamda PDÖ modelinin başarılı olabilecek bir model olarak ele alınıp incelenmesinde ve yaygınlaşmasında faydalı olabileceği düşünülmektedir. Özellikle yükseköğretim gençliğini sanal ortamlardan, gerçek öğrenme ortamlarına çeken çözüm yollarını bulmak gerekir. Aksi halde, sayıları 200'ü aşan üniversitelerimiz yakın gelecekte okutacak öğrenci bulamama riski ile karşı karşıya kalabilir.

Bütün bu tespitlere rağmen, klasik modelin tamamen kaldırılması ve yerine PDÖ modelinin getirilmesi tavsiye edilmemektedir. Bazı bilim alanlarında klasik model, alternatifsiz bile olabilir. Bu çalışma, bilişim meslek alanının programlama boyutuna yönelik yapılmış bir araştırmadır. Gözlemler, bu alanda kendi başına öğrenenlerin daha başarılı olduğu biçimindedir. PDÖ modeli de bu gözlemleri desteklemiştir. Ancak, PDÖ modeli klasik modele alternatif değil, klasik modeli zenginleştiren ve karma bir model olmasını sağlayan bir model olarak değerlendirilebilir. Yani PDÖ modeli, klasik modelle birlikte kullanılarak karma ve etkili bir eğitim modeli elde etmenin çabası içinde olmak gerekir.

KAYNAKÇA

- Akpınar, Y. ve Altun, A. (2014). Bilgitoplumu okullarında prog-ramlama eğitimi gereksinimi, *İlköğretim Online*, 13(1), dy:1-4, On-line.
- Algan, S. (2010). *Her yönüyle C# programlama*. Pusula Yayıncılık, İstanbul.
- Alper, A. ve Deryakulu, D. (2008). Web ortamı problemlere dayalı öğrenmede bilişsel esneklik düzeyinin öğrenci başarısı ve tutumları üzerindeki etkisi, *Eğitim ve Bilim* 2008, Cilt:33, Sayı:148.
- Aydede, M.N. ve Kesercioğlu, T. (2012). Aktif öğrenme uygulamalarının öğrencilerin kendi kendine öğrenme becerilerine etkisi, *Hacettepe Üniversitesi Eğitim Fakültesi Dergisi*, 43: 37-49.
- Boran A.İ. ve Aslaner, R. (2008). Bilim ve sanat merkezlerinde matematik öğretiminde problemlere dayalı öğrenme, *İnönü Üniversitesi Eğitim Fakültesi Dergisi*, Cilt:9, Sayı:15, Bahar 2008, 5: 15-32.
- Barrows, H.S. and Tamblyn, R. M. (1980). *Problem-based learning and approach to medical education*. New York: Springer Publishing Co.
- Demirel, M. ve Turan, B. A. (2010). Problemlere dayalı öğrenmenin başarıya, tutuma, biliş ötesi farkındalık ve içgüdü düzeyine etkisi, *Hacettepe Üniversitesi Eğitim Fakültesi Dergisi*, 38:55-66.
- Edens, K.M. (2000). Preparing problem solvers for the 21st. Century through problem-based learning, *College Teaching*, 48(2), 55-57.
- Gates, B. (1999). *The speed of thought*, Doğan Kitap, Güneşli, İstanbul.
- Grabinger, S., Dunlap, J.C. and Duffield, J.A. (1997). *Rich environments for active learning in action: Problem based learning*, *Research in Learning Technology*, ALT-J. Routledge Taylor and Francis Group.
- Günhan, B. ve Başer, N. (2008). Problemlere dayalı öğrenme yönteminin öğrencilerin matematiğe yönelik tutumlarına ve başarılarına etkisi, *Abant İzzet Baysal Üniversitesi Eğitim Fakültesi Dergisi*, Cilt:8, Sayı:1, Yıl:8, Haziran 2008.
- Gürkan, E. (2011). Problemlere dayalı öğrenmenin öğrenme ürünlerine problem çözme becerisine, öz yeterlilik algı düzeyine etkisi. *Hacettepe Üniversitesi Eğitim Fakültesi Dergisi*, 40: 221-232.
- İkiz, F.; Püskülcü, H. ve Eren, Ş. (2015). *İstatistiğe giriş*, Seçkin Yayıncılık, İstanbul.
- Kaya, A. (2015). A reserach on motivation deficiency for associate degree students of higher education,

- International Conference on Re-search in Education and Science (ICRES), April 23-26, Antalya, Turkey.
- Kinnunen, P. And Malmi, L. (2005). Problems in problem-based learning experiences, analysis and lessons learned on an introductory programming course, *Journal Informatics in Education-an International Journal*, page:193-214, issue no. 2., Central and Eastern European Online Library.
- Kotter, P. Schlesinger, J. ve Leonard, A. (1991). *Choosing strategies for change, Management of change*, A Harvard Business Review Paper-back, Boston.
- Körükçü, E., Şengül, S. (2018), Zenginleştirilmiş öğrenme ortamında 7. Sınıf Öğrencilerinin Matematiksel Zihin Alışkanlıklarından İletişim Kurma Becerisinin Gelişiminin İncelenmesi, *The Journal of Academic Social Science Studies (JASSS)*, Number 68, p.1-40, Summer I 2018.
- Law, N.Y.W. (2011). Keynote Spekar of ICITS 2011, Associate Dean of Education Faculty Director of Centre for Information Technology in Education. University of Hong Kong.
- McKay, G. (2010). *Cesur ol, değişime ayak uydur*, Kalipso Kişisel Gelişim, Kalipso Yayınları, İstanbul.
- Musal, B., Akalın, E., Kılıç, O., Esen, A. ve Alıcı, E. (2002). Dokuz Eylül Üniversitesi Tıp Fakültesi probleme dayalı öğretim programı, süreçleri ve eğitim yönlendiricilerinin rolü, *Tıp Eğitimi Dünyası*, 9 Ekim 2002.
- Nuutila, E., Törma, S., Kinnunen, P. and Malmi, L. (2008). Learning programming with PBL method- Experiences on PBL cases and tutoring, *Teaching of programming*, LNCS 4821, pp:47-67. Springer-Verlag Berlin Hiedelberg, 2008.
- Paulo, L, Ambrosio, Costa, F.M. (2010). *SIGCSE' 10 Proceedings of the 41 st. ACM technical Symposium on Computer Science Education*, Pages:495-499, Milwaukee, Wisconsin, USA.
- Peppard, J. (1993). *IT strategy for business*, Pitman Publishing, New York.
- Selçuk G.S., Karabey B. ve Çalışkan S. (2011). Probleme dayalı öğrenmenin matematik öğretmen adaylarının ölçme ve vektör konularındaki başarıları üzerine etkisi, *Mustafa Kemal Üniversitesi Sosyal Bilimler Enstitüsü Dergisi*, Yıl:2011, Cilt: 8, Sayı:15, s:313-322.
- Taşkın, E. (2001). *İşletme yönetiminde eğitim ve geliştirme*, Papatya yayıncılık, İstanbul.
- Toros, S, Yeşiltaş, E., (2015). Sosyal Bilgiler öğretimde bilgisayar destekli öğretiminin kavram yanlışlarını giderme üzerine etkisi, *The journal of Academic Social Science Studies*, (JASSS), No:39, S:157-172.
- Ülger, K. ve İmer, Z., (2013). Probleme dayalı öğrenme yaklaşımının öğrencilerin yaratıcı düşünme becerileri üzerine etkisi, *Hacettepe Üniversitesi Eğitim Fakültesi Dergisi*, 28(1), 382-392.
- Yaman, S. ve Yalçın, N. (2005). Fen bilgisi öğretiminde problem dayalı öğrenme yaklaşımının yaratıcı düşünme becerisine etkisi, *Hacettepe Üniversitesi Eğitim Fakültesi Dergisi*, 29:229-236.