

PAPER DETAILS

TITLE: Matematik Egitiminde Bilgi İşlemsel Düşünme- Kuramdan Uygulamaya

AUTHORS: Rümeysa Beyazhançer

PAGES: 214-235

ORIGINAL PDF URL: <https://dergipark.org.tr/tr/download/article-file/4111883>

Fen, Matematik, Giriřimcilik ve Teknoloji Eđitimi Dergisi

Journal of Science, Mathematics, Entrepreneurship and Technology Education

<https://dergipark.org.tr/tr/pub/fmgtd>

© ISSN: 2667-5323

Matematik Eđitiminde Bilgi İřlemsel Düşünme- Kuramdan Uygulamaya

Rümeysa Beyazhançer¹

¹Dr, Bursa Uludağ Üniversitesi, rumeysabeyazhancer@uludag.edu.tr, ORCID ID: 0000-0001-5061-8835

ÖZ

Bu çalışma, bilgi işlemsel düşünme (BİD) kavramının matematik eğitimi ile entegrasyonunu ve bu entegrasyonun öğretim süreçleri üzerindeki etkilerini incelemektedir. Çalışmanın amacı, BİD'nin matematik eğitiminde nasıl bir rol oynadığını ve bu sürecin öğretmenler ve öğrenciler üzerindeki etkilerini değerlendirmektir. Çalışma kapsamında, BİD'nin matematiksel kavramların anlaşılmasına, problem çözme becerilerinin gelişimine ve teknoloji destekli öğrenme ortamlarının kullanımına olan etkileri incelenmiştir. Araştırmanın bulguları, BİD'nin matematik eğitime entegrasyonunun öğrencilere problem çözme, soyutlama ve algoritma geliştirme gibi önemli becerileri kazandırdığını göstermektedir. Ayrıca, BİD tabanlı etkinlikler, öğrencilerin matematiksel kavramları daha iyi anlamalarını ve bu kavramları günlük hayatla ilişkilendirmelerini sağlamaktadır. Öğretmenler ise BİD tabanlı pedagojik uygulamaları kullanarak teknoloji ve bilgisayar destekli eğitim araçlarını daha etkili bir şekilde kullanabilmişlerdir. Bu süreçte öğretmenlerin karşılaştığı zorluklar ve bu zorlukların nasıl aşılabileceği konusunda önemli bilgiler elde edilmiştir. Sonuç olarak, BİD'nin matematik eğitime entegrasyonu, öğrencilerin analitik ve yaratıcı düşünme becerilerini geliştirmekte ve öğretmenlerin pedagojik uygulamalarını zenginleştirmektedir. Bu çalışma, BİD'nin eğitimdeki önemini vurgulamakta ve öğretim stratejilerinin geliştirilmesine yönelik değerli öneriler sunmaktadır.

MAKALE TÜRÜ

Derleme

MAKALE

BİLGİLERİ

Gönderilme Tarihi:

31.07.2024

Kabul Edilme Tarihi:

31.10.2024

ANAHTAR

KELİMELER:

Bilgi işlemsel düşünme, Matematik eğitimi, Algoritmik düşünme

Computational Thinking in Math Education-From Theory to Practice

ABSTRACT

This study examines the integration of computational thinking (CT) into mathematics education and its impact on teaching processes. The aim of the study is to evaluate the role of CT in mathematics education and its effects on both teachers and students. The study investigates the impact of CT on the understanding of mathematical concepts, the development of problem-solving skills, and the use of technology-supported learning environments. The findings of the research show that the integration of CT into mathematics education equips students with essential skills such as problem-solving, abstraction, and algorithm development. Additionally, CT-based activities help students better understand mathematical concepts and relate these concepts to daily life. Teachers have been able to use technology and computer-supported educational tools more effectively by applying CT-based pedagogical practices. This process has provided significant insights into the challenges faced by teachers and how these challenges can be overcome. In conclusion, the integration of CT into mathematics education enhances students' analytical and creative thinking skills and enriches teachers' pedagogical practices. This study emphasizes the importance of CT in education and offers valuable suggestions for the development of teaching strategies.

ARTICLE TYPE

Review

ARTICLE

INFORMATION

Received:

31.07.2024

Accepted:

31.10.2024

KEYWORDS:

Computational thinking, Mathematics education, Algorithmic thinking

Summary

Introduction, Purpose, and Significance

This study examines the integration of Computational Thinking (CT) into mathematics education, analyzing its effects on students and educators. CT is increasingly recognized as essential for fostering analytical skills, problem-solving, and a deeper understanding of mathematical concepts through technology-supported learning environments. The goal of this research is to highlight CT's role in developing foundational skills such as problem decomposition, abstraction, and algorithmic thinking, which are critical for academic and real-world applications. The frequent use of CT-based activities in classrooms is expected to improve students' ability to connect mathematical concepts to everyday life, enhancing both learning outcomes and instructional quality.

Methods

A traditional literature review methodology was employed to structure, analyze, and synthesize existing research on CT in mathematics education. This approach provides a comprehensive view of CT's theoretical and practical applications within the field, allowing the study to identify trends, gaps, and contradictions in the current literature. The review includes various qualitative and quantitative studies, focusing on both primary and secondary educational settings.

Findings

The research identifies several positive impacts of CT integration in mathematics education:

1. **Skill Development:** CT-based activities enhance students' problem-solving abilities, abstraction skills, and algorithmic thinking. These competencies enable students to understand complex mathematical concepts more effectively.
2. **Teacher Benefits:** Educators gain valuable insights into using technology-supported teaching tools and overcoming challenges in CT integration, which enriches their pedagogical practices.
3. **Interdisciplinary Relevance:** CT also supports the broader goals of STEM education, equipping students with the ability to recognize patterns, analyze situations, and approach problems systematically.
4. **Challenges and Recommendations:** Teachers encounter various obstacles in adopting CT methods; this study outlines strategies to address these challenges and improve CT's implementation across different educational levels.

Discussion and Conclusion

The study concludes that CT integration in mathematics education contributes to students' analytical, creative, and problem-solving skills. The findings suggest that CT-based curricula should be prioritized to cultivate students' resilience, adaptability, and critical thinking—qualities that are increasingly essential in today's technological world. Recommendations include targeted teacher training programs and continued research on CT's interdisciplinary applications, especially in STEM fields, to further refine its effectiveness.

Giriş

Yirmi birinci yüzyılın hızla değişen ve teknolojik olarak ilerleyen dünyasında, problem çözme yeteneklerini geliştirmek ve daha etkin kararlar almak için yeni yaklaşımlara ihtiyaç duyulmaktadır. Yirmi birinci yüzyıl öğrenimi, inovasyon ve yaratıcılık, esneklik, eleştirel düşünme gibi beceri setleri

etrafında řekillendirilmiř yeteneklerle donatılmayı gerektirmektedir. Bu bađlamda, öđrencilerin ekonomi ve iř dñnyası, teknoloji, çevre ve küreselleřmenin etkileri gibi alanlarda hızlı ve artan deđiřiklikler tarafından ortaya çıkarılan giderek karmařıklařan zorluklarla bařa çıkabilmek için bu özel yeteneklere gereksinim duymaktadır. Bařka bir deyiřle, bu çađın öđrencisini, esas olarak hızlı bađlantılar, robotik, büyük veri, yapay zekâ gibi teknolojik geliřmelerle tanımlanan Dördüncü Sanayi Devrimi'nin (4IR) karmařık zorluklarına hazırlamak söz konusudur.

"Computational Thinking" Türkçe ifade ile "Biliřimsel Düşünme" (Yıldız, Çiftçi & Karal, 2017), 21. yüzyılın en önemli becerilerinden biri olarak öne çıkmaktadır. Computational Thinking, Türkçeye "Bilgi İşlemsel Düşünme", "Hesaplamalı Düşünme", Biliřimsel Düşünme" olarak farklı řekillerde tercüme edilmiřtir. Bu çevirilerden "Bilgi İşlemsel Düşünme (BİD)" bilgiyi işleyen bir düşünme řekli de olduđundan ve matematik eđitiminde bilgi işleme ve bilgi oluřturma önem arz ettiđinden bu řekilde ifade edilmesi uygun görölmektedir (Kabakçı, vd., 2023). BİD Eđitimi terimi, öğrenimi kolaylařtırmada bilgisayar ve bilgi işlemsel düşünmenin rolünü vurgulamaktadır; bu bakıř açısı, Seymour Papert'in mirasıdır (Papert, 1993). "BİD" terimi, bilimin 19. yüzyılda nicel analize atıfta bulunmasıyla ortaya çıkmıřtır ve daha sonra aritmetik öđretiminde akıl yürütmenin vurgulanmasıyla ilişkilendirilmiřtir (Childs, 2015). Terimin modern olarak bilgisayarlar ve eđitimle ilişkilendirilmesi ise Papert'e dayanmaktadır. Papert'tan (1980, 1993, 1996, 2006) sonra Wings'in (2006) bu alanda yaptıđı çalıřmalarla BİD bir problem çözme biçimi olarak eđitimde kullanılmaya devam etmiřtir.

BİD, matematik eđitimi konusunda önemli bir düşünce, bilgisayarlařmanın ve teknoloji kullanımının pedagojik ilişkiyi kolaylařtırma derecesidir. Bu durum, birçok açıdan, öđretmenlere BİD kavramı ve çeřitli pedagojik bađlantılar, sosyal ve kültürel olarak, uygulanabilirliđi hakkında daha fazla bilgi edinmelerine yardımcı olmakla ilgilidir (Ye vd.,2023).

Bu çalıřmada, Bilgi İşlemsel Düşünme'nin ne olduđu, nasıl ortaya çıktıđı ve matematik eđitiminde nasıl bir rol oynadıđı ele alınmaktadır. Ayrıca çalıřma, eđitmcilerin ve arařtırmacıların katkılarını içermekte ve özellikle ilköđretim ve ortaöđretim seviyesinde eđitime odaklanarak Bilgi İşlemsel Düşünme Eđitimi (BİDE) teorilerinin matematik eđitimi bađlamında arařtırılmasını ve pedagojilerinin anlařılmasını derinleřtirmeyi amaçlamaktadır.

Yöntem

Arařtırma literatürde bilgi işlemsel düşünme becerisi ile ilgili var olan yaklařımları yapılandırma, literatürü analiz etme ve sentezleme bu bađlamda alana kuramsal anlamda katkı sađlaması amacıyla yapılmıřtır. Bu amaçla çalıřmada tarama modelinde geleneksel derleme yaklařımı kullanılmıřtır Geleneksel derleme belirli bir konuda yayınlanmış iki veya daha fazla çalıřma üzerinde inceleme yapılarak bulgu, sonuç ve deđerlendirmelerini sentezleyen, belirli bir yöntem izlenmeksizin, farklı yollarla ve farklı kaynaklardan elde edilen bilgilerin derlendiđi çalıřmalardır (Karaçam, 2013). Geleneksel derleme türü çalıřmalar, önceki arařtırmaların özetini sunarak kavramlar arasındaki bađlantıları, çeliřkileri ve boşlukları ortaya koyar, ilgili literatüre bütüncül bir bakıř sađlar. Bu tür çalıřmalar, konuyla ilgili genel eğilimlerin ve çıkarımların belirlenmesi açısından önem tařır (Yılmaz, 2021).

1.BİD'nin Tarihçesi ve Ortaya Çıkıřı

Bilgisayar destekli eđitimin ilk ve ortaöđretim öđrencilerine tanıtılması daha yavař ve düzensiz olmuřtur. Bu alandaki önemli bir çalıřma, John Kemeny ve Thomas Kurtz tarafından 1964 yılında Dartmouth College'da BASIC programlama dilinin icadı ile ortaya çıkmıřtır. Beginner's All-purpose Symbolic Instruction Code (BASIC) vizyonu, herkesin ya da en azından her Dartmouth lisans öđrencisinin kodlamayı öğrenmesi olmuřtur. BASIC, Dartmouth Zaman Paylařım Sistemi DTSS'ye bađlı terminallerde çalıřırken, üniversite DTSS'yi Dartmouth lisans öđrencilerine ve birkaç üniversite ile liseye erişilebilir kılmıřtır. BASIC, 1977'de Apple II ve diđer ev bilgisayarlarının tanıtımıyla patlama

yaşamıştır ve hala programlama öğrenmeye odaklanan hobi grupları ve okul bilgisayar kulüpleri arasında son derece popüler olmaya devam etmektedir.

Ortaöğretimde BİDE'nin kullanılması BİDE'nin doğrudan öncüsü- Seymour Papert'in çalışmasından ve 1967'de Papert, Cynthia Solomon ve Wallace Feurzeig tarafından Logo bilgisayar dilinin oluşturulmasıyla ortaya çıkmıştır (Solomon, 1986). Bu fikrin tam ifadesi ve "bilişimsel düşünme" olarak nitelenmesi, Papert'in Mindstorms adlı kitabında (Papert, 1980) ilk kez ortaya çıkmıştır, ancak Papert bu fikri "prosedürel düşünme" olarak da adlandırarak şöyle yazmıştır:

"Bu kitapta, prosedürel düşünmenin güçlü bir entelektüel araç olduğunu açıkça savundum ve hatta bir strateji olarak kendini bir bilgisayara benzetmeyi ya da bunu yapmayı önerdim... Bilgisayarın varlığının kültürel asimilasyonu, bilgisayar okuryazarlığının ortaya çıkmasına neden olacaktır. Bu ifade genellikle programlama bilmenin veya bilgisayarın çeşitli kullanımları hakkında bilgi sahibi olma anlamına geldiği şeklinde algılanır. Ancak gerçek bilgisayar okuryazarlığı, sadece bilgisayarları ve bilişimsel fikirleri nasıl kullanacağını bilmek değildir. Onu kullanmanın ne zaman uygun olduğunu bilmektir." (Papert, 1980, s. 155)

Bu süreçte bilgisayarın okullardaki kabulü zayıf olmuştur ve hesaplamanın sadece bilgisayar programlama kariyerini hedefleyen öğrenciler için değil, herkes için güçlü bir öğrenme çerçevesi olabileceği fikrine pek uyulmamıştır. Okul bilişim hareketinin birkaç liderinin 2003 yılında gözlemlediği gibi: 'literatür, etkinin potansiyeline işaret ederken, gerçeklik iç açıcudur: birinci dereceden bir yaklaşımla, son 25 yılda bilgi teknolojisinin ilköğretim ve ortaöğretime etkisi esasen sıfırdır' (Norris, vd., 2003). Bilgisayarların ilköğretim ve ortaöğretim eğitiminde benimsenmesi, toplumda bilgi teknolojisinin etkisinin büyük ölçüde artması ve bilgisayarın günlük yaşamda sürekli bir varlık olarak ortaya çıkması ile olmuştur. Değişimin önemli bir tetikleyicisi, Jeanette Wing'in (2006) etkili çalışması "Bilgi İşlemsel Düşünme"dir. Wing, terimi Papert'in geleneği içinde, BİD'nin sadece programlama olmadığını ve herkes için temel bir beceri olması gerektiğini vurgulayarak tekrar tanıtmıştır. Wing'in çalışması, bilgi teknolojisinin muazzam bir etkisinin başlamasıyla eşzamanlı olarak, BİD ve ilk ve ortaöğretim eğitiminde bilgisayar kullanımına yönelik uzun süreli gözlemcileri şaşırtan bir ilgi patlamasına yol açmıştır. Wing (2016) gözlemlerini şu şekilde paylaşmıştır:

"Bana ortaöğretimde bilgisayar bilimi öğretilecek mi diye sorulduğunda söylediğim şey; 'Hayır, benim ömrümde olmaz' dı. 2009 yılındaydı ve Ulusal Akademiler tarafından düzenlenen bilgi işlemsel düşünme üzerine bir çalıştayın katılımcılarına hitap ediyordum. Mutlulukla söyleyebilirim ki yanılmışım."

Bu artan ilgiye rağmen, BİD'nin tam olarak ne olduğu konusunda geniş bir belirsizlik sürmekte ve temel ilkelerini açıklamak için mücadele devam etmektedir. Wing'in yukarıda bahsettiği 2009 Ulusal Akademiler çalıştay raporu, bunun arkasındaki motivasyonu; "Farklı çabalar, ilköğretim ve ortaöğretim öğrencilerine en temel ve esas hesaplama kavramlarını tanıtmak için yapılmıştır ve üniversite müfredatları, öğrencilere artan yeni ve gelişmiş hesaplama kavramları ve teknolojileri için ömür boyu öğrenme temeli sağlamaya çalışmıştır. Ancak bu spektrumun her iki ucunda da çoğu çaba temel kavramlara odaklanmamıştır" şeklinde ifade etmiştir.

Ortaöğretim müfredatına hesaplamayı dahil etmek için yaygın bir yaklaşım, genellikle haber bültenleri, belgeler, web sayfaları, multimedya sunumları veya bütçeler oluşturmak için araçları kullanmayı içeren bilgisayar okuryazarlığına vurgu yapmaktır. İkinci yaygın bir yaklaşım, öğrencilere Java veya C++ gibi belirli programlama dillerinde programlama öğreterek bilgisayar programlamaya vurgu yapmaktır. Üçüncü yaygın bir yaklaşım, oyunlar, robotlar ve simülasyonlar gibi programlama uygulamalarına odaklanır. Birçok bilgisayar bilimcisine göre, bu üç ana yaklaşım kullanışlı ve tartışmasız önemli olmalarına rağmen BİD'i öğrenmekle karıştırılmamalıdır" (National Research Council, 2010). Burada dikkati çeken husus, BİD'i programlamadan ve bilgisayar araçlarının kullanımından ayırt etme endişesi, BİD hareketinin başlangıcında Papert'in Mindstorms kitabında ifade ettiği endişeyle aynıdır, bu da 30 yıl önceki zamana dayanır.

2. Bilgi İşlemsel Düşünme'nin Tanımı

BİD, problemleri çözme, sistemleri tasarlama ve insan davranışını anlama konularında bilgisayar biliminin temel prensiplerini kullanma yeteneğidir (Wing, 2006). Bu düşünce şekli,

karmařık problemleri daha basit alt problemlere blme, bu problemlere adım adım zm yolları geliřtirme ve bu srete algoritmik dřnme becerilerini kullanma zerine kuruludur (Wing, 2006; Denning, 2017; Weintrop vd., 2016).

"BİD nedir?" sorusuna cevap bulmak iin nce bilgi iřlemenin dođasına bakmak gerekir. Klasik olarak, bilgi iřleme, girdi verildiđinde ıktı reten bir algoritmik sretir. Peter Denning (2005), bilgi iřlemeyi bir elemandan diđerine geiřin bir temsil tarafından kontrol edildiđi bir sre olarak tanımlar. Bu nedenle, bilgi iřleme dřncesini, bir problemi bir bilgi sreci olarak temsil eden ve algoritmik bir zm arayan bir yaklařım olarak tanımlar. Birok diđer tanım gibi, bu tanımın deneysel olarak yardımcı olacak kadar geniř olduđu sylenebilir. Bilgi iřlemenin znn temsil ve modelleri aramak olduđu grř muhtemelen makul grnmektedir. İkiisi birbirine sıkıca bađlı ancak incelikli farklı kavramlardır. Modeller, bir bakıma temsil ederler. Bir model, modeldeki varlıkların nasıl temsil edildiđi, temsil etme řeklinin bir sonucudur ve sadece řeyleri temsil etme řeklini deđil, varlıkların temsil ediliřlerine dayanan dinamikleri yakalar. Bir model soyut olabilirken, bir temsil muhtemelen somuttur. Bir model, veriyi daha iyi anlařılır veya daha "kolay" maniple edilebilir hale getirmek iin veriyi bir temsilinden diđerine dnřtrmeye izin verir. Modeller tamamen yapay, matematiksel olarak dnřtrlmř veya mevcut veri kalıplarını tanıyarak algoritmik olarak oluřturulabilir (Denning,2005).

Yukarıdaki deđerlendirmelerin sonucunda, BİD; *somut veya soyut, temsil edilen veya modelin i iřleyiř mekanizmasını temsil etmek veya modelin yrtlecek bir bilgi sreci olarak temsil edilecek řekilde modeller ve temsiller oluřturarak problemleri zme, sistemleri otomatikleřtirme veya veriyi dnřtrme dřncesi* olarak ifade edilebilir.

Bu durumda BİD zorunlu olarak:

- Modellerin veya temsillerin esas olanını yakalaması iin **mantıklı**,
- Adım adım iřlemsel sreleri tanımlamak veya geliřtirmek iin **algoritmik**,
- Modellerin yeteneklerini anlamak, onları maksimum verimlilikle nasıl kullanılacaklarını đrenmek ve hesaplamanın orijinal problem alanındaki etkilerini keřfetmek iin **bilimsel**,
- Algoritmaların dođruluđunu gstermek, yazılım sisteminin iřlevselliđini kesin bir řekilde belirtmek, hesaplama srecinde yapılanların kalitesini lmek ve modellerin ve temsillerin karmařıklıđı ile etkili ve verimli alternatifleri keřfetmek iin **matematiksel**,
- Ama, varsayımlar ve bakıř aıları ile modelleme, prototipleme yoluyla modelleri ve temsilleri deđerlendirmek ve ayarlamak, sonularını incelemek iin **analitik**,
- Bilinen kısıtlamalara ve pratik endiřelere karřı modelleri ve temsilleri tasarlamak ve hesaplama srecini planlamak, yrtmek, ynetmek ve deđerlendirmek iin **mhendislik odaklı**,
- Dřnlemez olanı modellemek iin **yaratıcı**

olmalıdır.

Bu yaklařım, ilkesel olarak nceki tanımlarla uyumludur. Ancak bu tanımı farklı kılan, her paradigmayı bilgi iřlemeye uyguladıđımızda, her birinin ilgili dřnme yaklařımlarıyla olan bađlantısının ne kadar uygun olduđudur. Bir diđer deđiřle, hesaplama yani bilgi iřleme eřdidir. Bu nedenle, tm olası dřnme eřitlerini, bunların kombinasyonlarını ve trevlerini kapsayacak řekilde bilgi iřleme dřncesini tanımlamak zordur. Yukarıdaki tanım, belirli bir bilgisayar alanında ve bu alana odaklı zelliklere daha uygundur. rneđin, yazılım tasarımı "tasarım dřncesi" gerektirir, tasarımcının eliřkili olasılıkları tolere etmesine, byk resmi grmesine, belirsizlikle bařa ıkmasına, sađlıklı kararlar almasına ve birka tasarım dilinde iletiřim kurmasına izin veren bir dřnce trdr. Sonu olarak, nemli olan, bir dřnme srecini etiketleme deđil, eleřtirel dřnme yeteneđimizin kendisidir (Li vd., 2020).

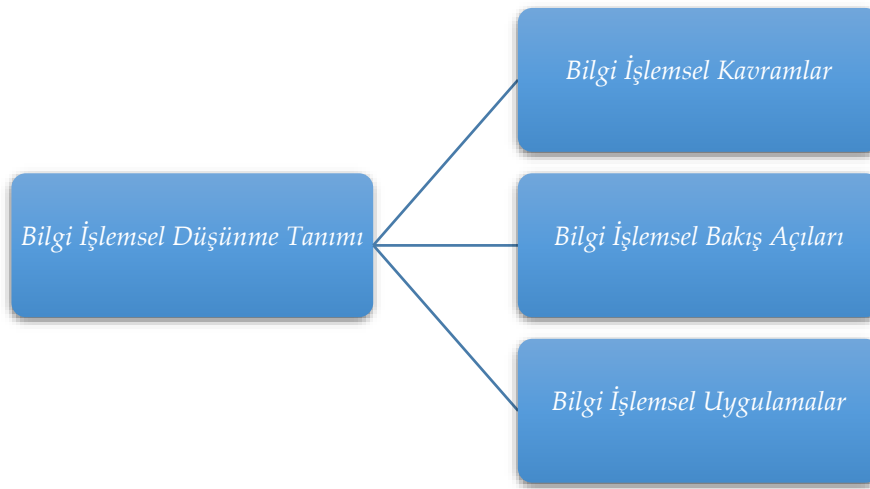
Bilgisayar bilimi bir disiplin olarak, hl zerine konulabilecek iyi kurulmuř hesaplama modellerine sahiptir. Ancak belki de diđer disiplinlerden izilen perspektifler, bilgisayarı gerekten zorluklarla ve bařarılarla dolu kılan řeydir. BİD, sadece hesaplamanın dođasından kaynaklanmaz, aynı zamanda insanların eleřtirel dřnmesinden kaynaklanır. Uygulamalar, dřnme řeklini etkiler. rneđin, STEM (Science, Technology, Engineering, Mathematics) eđitimi, đrencilere STEM

problemlerini çözdürerek değil, bilimsel, mühendislik veya matematik düşünceyi savunarak teşvik edilmelidir. STEM'in içinde hesaplama vardır. Bilgisayar öğrenmeden STEM öğrenmek temelde yetersizdir. Öte yandan, STEM problemlerini çözerken bilgisayar öğrenmek, düşünme becerisini her durumda geliştirecektir.

Bresnan ve Resnick (2012) tarafından geliştirilen BİD tanımı, üç ana boyutu içermektedir: bilgi işlemsel kavramlar, bilgi işlemsel uygulamalar ve bilgi işlemsel perspektifler. Bilgi işlemsel kavramlar, tasarımcıların programlama yaparken kullandıkları kavramları ifade ederken, bilgi işlemsel uygulamalar, tasarımcıların programlama sürecinde geliştirdikleri uygulamalara işaret etmektedir. Bilgi işlemsel perspektifler ise tasarımcıların çevrelerindeki dünya ve kendileri hakkındaki bakış açılarını kapsar. Araştırmacıların BİD'nin bileşenlerini açıkladığı model Şekil 1'de özetlenmiştir.

Şekil 1

BİD'in Tanımındaki Bileşenler (Bressnan&Resnick,2012)



Bu tanımın gelişiminde gözlem ve röportajlar önemli bir rol oynamıştır. Katılım ve proje portföylerinin haftalardan birkaç yıla kadar uzanan bir zaman dilimini kapsamayı, yaratıcıların uzun vadeli gelişimini anlamamıza yardımcı olmuştur. Aynı zamanda, atölye çalışmaları, yaratıcıların eylem halindeki uygulamalarını anlamada kritik bir öneme sahiptir. Bu boyutlar, BİD'in kapsamlı bir şekilde anlaşılmasına ve eğitim süreçlerine entegrasyonuna olanak sağlamaktadır.

Weintrop ve arkadaşları (2016) çalışmalarında, bilgisayar bilimi araştırmalarının kolektif bilgisinin ve derslerinin (bilgi işleme) tüm konuların tartışma ve geliştirilmesine dahil edilmesini önerirler. Ayrıca, öğretilecek konulara doğal olarak entegre edilebilecek bazı hesaplamalı etkinlikler önerirler. Bu şekilde, öğrenciler programlama öğrenirken hesaplama becerilerini geliştirebilir ve ilerledikçe daha başarılı olabilirler. Bununla birlikte, eğer hesaplama düşüncesinin ana hedefi süreçlerin soyutlanmasıysa, Jean Piaget'in Bilişsel Gelişim Evreleri teorisi, bu düşünme becerisinin etkili bir şekilde ancak ergenlik döneminde öğretilebileceğini öne sürer. Bu durumda, daha erken yaşlarda bir "hesaplama kültürü" oluşturmak gerekebilir. Bu kültür, BİD'in (çeşitli biçimlerde) öğretilen derslere doğal bir şekilde entegre edildiği bir eğitim anlayışını temsil eder.

Hesaplama kültürünü teşvik etmek mümkündür ve çeşitli yollarla gerçekleştirilebilir. Öğrencilere internette bilgi arama becerilerini kazandırırken, bu aramaları etkili bir şekilde nasıl yapabilecekleri üzerinde durulabilir. Öğrenciler Excel programını öğrenirken, bu yazılımı optimizasyon problemlerini çözmede nasıl kullanabilecekleri öğretilebilir. Denklem köklerini bulma yöntemleri öğretildiğinde, daha etkili arama algoritmaları (örneğin, ikili arama) ve diğer kök bulma yöntemleri hakkında bilgi verilebilir. Ayrıca, öğrenciler lise düzeyinde iken modelleme deneyimi kazanmaları amacıyla makul ölçekte veri tabanları tasarlamayı öğrenebilirler. Bu şekilde, öğrenciler

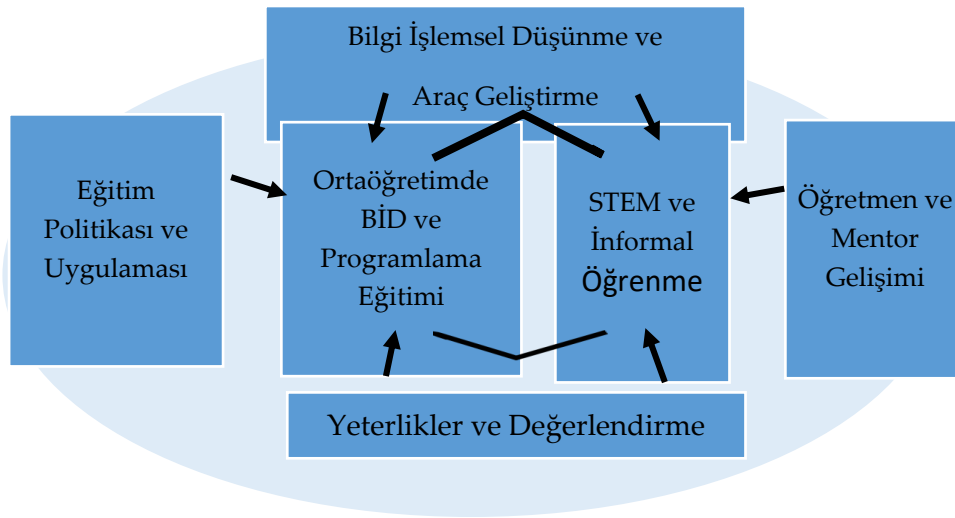
üzerinde inşa edilebilecek ve teşvik edilmesi gereken temel hesaplama yeteneklerine sahip olabilirler (Grover & Pea, 2013).

Bilgisayar bilimi ve mühendisliğin temel taşı olan bu düşünce tarzı, aslında sadece bilgisayar bilimcileri için değil, her alanda kullanılabilir bir yaklaşım sunar. Öğrencilerden mühendislere, iş insanlarından eğitimcilere kadar herkesin karşılaştığı problemleri çözmede bir rehber olarak düşünülebilir. BİD, soyutlama, desen tanıma, algoritmik düşünme ve verimli veri temsili gibi kavramları içerir. Bu beceriler, bir problemi anlama, analiz etme ve ona sistemli bir şekilde yaklaşma yeteneğini geliştirir (Wing, 2006; Denning, 2017; Weintrop vd., 2016).

Şekil 2’de Bilgi İşlemsel Düşünmenin kavramsal çerçevesi verilmektedir.

Şekil 2

Bilgi işlemsel düşünme eğitiminin kavramsal çerçevesi (Kong, S. C. & Abelson, H. 2019)



BİD, Şekil 2’de görüldüğü gibi altı temel bileşene sahiptir. BİD ve Araç Geliştirme ana bileşen olmakla birlikte, Ortaöğretimde BİD ve Programlama Eğitimi ve STEM ve informal öğrenme alt ana bileşenlerdir. Ortaöğretimdeki BİD ve Programlama Eğitimi’ne destek veren bileşen eğitim politikası ve uygulamasıdır. STEM ve informal öğrenmeye destek veren alt bileşen öğretmen ve mentor gelişimidir. Sonuç olarak, alt ana iki bileşen olan Ortaöğretimde BİD ve Programlama Eğitimi ve STEM ve informal öğrenmeyi analiz etmenin alt bileşeni olan Yeterlikler ve Değerlendirme ile BİD Eğitimi’nin çıktıları incelenmektedir (Kong & Abelson, 2019).

3. BİD'nin Eğitimde Uygulama Alanları

BİD araştırmalarının birçok alanı kapsadığı ve BİD’in birçok alandan akademisyenlerin ilgisini çektiği açıktır. Çalışmaların çoğunun BİD’i sosyal bilimlerde (%34,9), ardından bilgisayar bilimlerinde (%31,4) ve mühendislikte (%12,1) harmanladığı bulunmuştur (Tekdal, 2021). Son yirmi yılda, BİD becerisi hakkında önemli bir literatür gelişmiştir. Burada bahsedilen BİD 'problemleri ve çözümlerini formüle etme sürecindeki düşünce süreçleri, böylece çözümlerin bir bilgi işlem aracı tarafından etkili bir şekilde gerçekleştirilecek bir biçimde temsil edilmesi' olarak tanımlanmaktadır (Wing, 2011, s. 20). Terimi ilk kez kullanan Wing (2006, 2008), BİD’in günlük hayatta gereken temel bir beceri ve matematiksel düşünme, mühendislik ve bilimsel uygulamalarla yakın bağlantıları olan bir analitik düşünme türü olduğunu vurgulamıştır. BİD üzerine yapılan araştırmalar, özellikle ilköğretim ve ortaöğretim eğitimi bağlamlarına entegrasyonu açısından artan bir ilgi görmektedir (Feldhausen vd., 2018; Jocius vd., 2021; Sırakaya vd., 2020; Swaid, 2015). Bu tür entegrasyonların BİD ile açık bir bağlantısı

olmasına rağmen, BİD'in matematik eğitime entegrasyonuna dair henüz çok az şey bilinmektedir (Barr & Stephenson, 2011; Sneider vd., 2014; Weintrop vd., 2016).

BİD önemi giderek daha belirgin hale gelmiştir. Bu alanlarından biri, BİD'in resmi eğitime entegrasyonunu desteklemek ve motive etmek için BİD ilhamlı öğrenme araçları ve kaynaklarının geliştirilmesi olmuştur. Zhang ve Nouri (2019), Jiang ve Wong (2022), Critten ve arkadaşları (2022) ve Hooshyar ve arkadaşları (2021), öğrencileri BİD uygulamalarına aktif olarak dahil etmeyi amaçlayan kodlama ortamları, robotik kitle ve ilgi çekici eğitim uygulamaları gibi araçlar yaratmaya odaklanmışlardır. Bir diğer önemli çaba, BİD'i STEM (bilim, teknoloji, mühendislik ve matematik) konularına entegre eden müfredatların geliştirilmesine yönlendirilmesidir. Örneğin, Lee ve Malyn-Smith (2020), Huang ve Qiao (2022) ve Yuen ve arkadaşları (2023), STEM öğrenme aktivitelerine teknolojiyi sorunsuz bir şekilde entegre eden eğitim modelleri tasarlamışlardır. Bu müfredat yeniliklerinin temel amacı, öğrencilere STEM ile ilgili eğitim faaliyetlerinde bulunurken BİD yeteneklerini güçlendirme fırsatı sunmaktır. Yukarıdaki çalışmalar, BİD uygulamalarının çoğunlukla STEM konuları için kullanıldığını göstermiştir. Bu durum, BİD'nin matematiksel, algoritmik ve eleştirel düşünme gibi STEM alanlarında gerekli olan çeşitli becerilerle ilişkili olması nedeniyle faydalıdır. Benzer şekilde, Abdul Hanid ve arkadaşları (2022), BİD ilkelerinin geometri konularına entegre edilmesiyle, öğrencilerin geometrik kavramları daha derin bir şekilde anlayabileceklerini ve problem çözme becerilerini geliştirebileceklerini belirtmişlerdir. BİD, matematikte öğrencilerin durumları analiz etmelerine, desenleri tanımlamalarına ve matematiksel görevleri yerine getirmek için algoritmalar oluşturmalarına yardımcı olmuştur. (Gadanidis vd., 2018; Rodríguez-Martínez vd., 2020).

BİD'in kavram ve yeterlikleri ilk ve ortaöğretime entegre etmek için yaptıkları çalışmada (Barr & Stephenson, 2011), BİD'in Şekil 3'teki temel bileşenlerden oluştuğu fikrini öne sürmüşlerdir.

Şekil 3

Barr ve Stephenson'a göre ortaöğretimde BİD kavram ve yeterlikleri (2011)



Şekil 3'te yer alan bu bileşenleri Barr ve Stepehenson(2011) ortaöğretimde BİD'i diğer disiplinlere örneğin matematik fen ve sosyal bilimler gibi farklı disiplinlere uyarlanabileceğini ve bu disiplinlerin öğretiminde kullanılabileceğini göstermişlerdir. Ayrıca, araştırmacılar öğrencilerin BİD becerilerini ölçmek için değerlendirme araçlarının geliştirilmesine odaklanmışlardır. Algoritmik düşünme ve problem parçalama gibi alanlarda öğrencilerin yeterliklerini ölçmek için özel olarak tasarlanmış değerlendirme araçlarının oluşturulmasına yol açmıştır (Hadad vd., 2020; Kang & Lee ,2020; Stewart vd., 2021). Bu değerlendirmeler, öğrencilerin BİD yeteneklerini geliştirmeye yönelik derslerin etkinliğini belirlemek için önemlidir.

4. Bilgi İřlemsel Düşünmenin Matematik Eđitimindeki Yeri

Matematik öğrenmek için BİD ve programlama araçlarının kullanımı, Papert'in (1980) Logo programlamasına kadar izlenebilir ve Weintrop ve arkadaşları (2016) tarafından daha da geliştirilmiştir. Bu çalışmalar, BİD uygulamaları ile bilim ve matematik uygulamaları yani **veri işleme, modelleme ve simülasyon, bilişimsel problem çözme ve sistem düşüncesi** arasındaki benzerliđi gösterir. Son yıllarda ortaöđretimde BİD tabanlı matematik öğretim yaklaşımlarının daha yaygın olarak benimsenmesiyle (Miller, 2019; Pei vd., 2018; Shumway vd., 2021), 'bilgi işlemsel olarak geliştirilmiş matematik eğitimi' kavramının kavramsal temelini ve uygulamasını geliřtirmek için birçok tasarım tabanlı araştırma yapılmıştır (Ng & Cui, 2021, s. 848).

İlgili çalışmalar ayrıca, öğrencilerin matematik öğrenmelerini kolaylařtırmak için programlama görevleri dahil olmak üzere öğretim tasarımını ilerletmiştir. Bu çalışmalar, Kotsopoulos ve arkadaşlarının (2017) BİD aktiviteleri için pedagojik çerçevesini, ortaöđretimde matematiksel problem çözmede bağlantısız, düşünce, yapım, yeniden karışım gibi tasarımları içerir ve kombinatorik (De Chenne & Lockwood, 2022), cebir (Bråting & Kilhamn, 2021), sayı teorisi ve matematiksel modelleme (Ng & Cui, 2021) ve geometri (Miller, 2019) alanlarını kapsar.

Bu çalışmalar, bilgisayar bilimi ve matematik kavramları arasında karşılıklı bir ilişki olduğunu göstermektedir. BİD'in uygulanması, matematiksel disiplin bilgisini derinleřtirirken, BİD tabanlı matematik öğretileri bireyin BİD becerilerini geliřtirmektedir (Pei vd., 2018). Ancak, programlamanın bazı matematik konularını öğrenmede etkili olduđu belirtilmiş olsa da her zaman kâğıt-kalem yöntemlerinden üstün olduđu veya matematiđin tüm alanlarının bilişimle entegrasyonu için uygun olduđu sonucuna varılamaz (Lockwood & De Chenne, 2020). Öğrencilerin, programlama bağlamında matematiksel problemleri çözerken bilişimsel ve matematiksel düşünme arasındaki potansiyel farklılıklar nedeniyle çeşitli zorluklar yaşadıkları bulunmuştur (Cui & Ng, 2021). Bu nedenle, BİD ve matematik arasındaki ilişki ve örtüşme konusunda fikir birliđi eksikliđi özellikle ortaöđretim eğitiminde BİD'in yaşı uygun ve disiplinlerarası şekilde geliřtirilmesine büyük vurgu yapıldıđı çalışmalar da bulunmaktadır. Önemli bir soru, bilişimin ilk ve ortaöđretimde matematik düşünce ve öğrenimini teşvik etmek için nasıl etkili bir şekilde kullanılabileceđidir (Hickmott vd., 2018).

Hu'nun 2011 yılında yapmış olduđu çalışmada BİD'in matematiksel düşünme ile çok yakından ilgili olduđu konu edilmiştir. Hu 'ya göre bilgisayar bilimi, birçok kişinin fark ettiđinden daha matematiksel bir alandır. Bilgisayar biliminin bazı yönleri, matematikte olduđu gibi desenleri tanıma ve manipüle etmeyle ilgilidir. Yazılım geliřtirme gibi diđer bazı alanlarda, çeşitli geliřtirme süreçlerinde yapılanların kalitesini ölçmek için belirli bir düzeyde doğruluk gerekebilir, ancak etkili ölçümler bulmak genellikle zordur. Sınıflar ve nesneler gibi programlama kavramları, kullananların çođu farkında olmasa da temelde matematiksel varlıklardır ve programlama, bilgisayar biliminin önemli bir biçimi olarak, iddialı bir şekilde matematiksel bir etkinlik olarak görülebilir. Doğasının matematiksel olmasına rağmen, bilgisayar bilimi sıklıkla birçok kişinin beklediđinden daha az matematiksel görünür. Her bilgisayar alanının altında yatan matematiđi bilmeksizin anlaşılabilir ve kullanabilir kendi metodolojileri vardır. Bilgisayar bilimi yapmak oldukça esnek olabilir. Örneđin, doğru bir döngü yazmak yalnızca cebirsel terimlerle düşünmeyi gerektirir. Ancak pratikte aynı şeyi yapmak için bir dizi ifadeyi yığabilirsiniz. İnsanlar genellikle işlevsel modüllerin algoritmik olarak doğru olup olmadıđını göstermek için doğruluk kanıtları yapmak yerine birim testlerini çalıştırır.

Günümüzde, uygulama programlama arayüzleri, kütüphane çerçeveleri ve işletme düzeyinde bütünleşmiş geliřtirme ortamları, insanların düşük düzeyde bir formal eğitimle yazılım geliřtirmesine olanak tanımaktadır. Ancak soyut terimlerde düşünme yeteneđinin eksik olması, insanların açık, zarif tasarımlar ve programlar üretme konusunda zorluk yaşamasının nedeni olabilir. Bu nedenle, matematiksel düşünme yeteneđini artırmak, bilgisayar biliminde yapılanların kalitesini artırmak için mantıklı bir yol olarak görünmektedir (Hu,2011). Bu noktada matematiksel düşünmenin önemi daha iyi anlaşılmalıdır ve řu düşünme yeteneklerini içerebilir:

*Örnekleme ve özelleřtirme (genel olarak belirtilen şeylerin örneklerini bulmak için),

*Tamamlama, silme ve düzeltme (sonuçların yapılmasına, güvenceye veya çelişkilere izin vermek için),

*Karşılaştırma, sıralama ve düzenleme (varsayımları ve hipotezleri daha iyi anlamak için),

*Değiştirme, değişken, gözden geçirme ve değiştirme (soruları, varsayımları, hipotezleri, kısıtlamaları veya çözüm yollarını)

*Genelleştirme ve tahmin etme, açıklama, haklı çıkarma, doğrulama, ikna etme ve çürütme (sonuçlar, çıkarımlar, yeniden formülasyonlar, örnekler, vb.).

Matematiksel bir düşünme süreci, bir görevi parçalara ayırmak, varsayımlar yapmak, benzer görevleri tanımlamak, uygun bilgi ve becerileri uygulamak, desenleri veya bağlantıları aramak, alternatifleri düşünürken bir strateji seçmek ve düşünmeye örnekler, veriler veya görsel yardımlarla yardımcı olmaktır. İddia edilebilir ki, bir kişi, matematiksel düşünme yeteneği ve belki de farklı bir yönlendirme ile bir matematiksel düşünme sürecini takip etme yeteneği ile bilgisayar bilimi yapar. Özellikle bir sistem modellemek veya tasarlamak için düşünmek, sistemi alt sistemlere ayırmak, tasarım seçeneklerini kavramlaştırmak ve simüle etmek ve birleştirici-ayrıştırıcı düşünce döngülerini uygulamak örnek olarak verilebilir. Bu basamaklara sahip bilgisayar bilimi ve BİD, matematik yapmaya çok benzerdir (Hu,2011). Tablo 1’de “Matematiksel Uygulama Standartları ile BİD’in Problem Çözme Üzerinde Karşılaştırılması” verilmiştir.

Tablo 1

Matematiksel Uygulama Standartları ile BİD’in Problem Çözme Üzerinde Karşılaştırılması (Hu,2011)

Matematiksel Uygulama Standartları	Bilgi İşlemsel Düşünme
Problemleri anlama ve çözümünde ısrar etme	Ayrıştırma
Soyut ve nicel olarak akıl yürütme	Soyutlama, algoritmik düşünme
Matematik ile modelleme	Algoritmik düşünme
Yapıyı arama ve kullanma	Örüntü tanıma ve soyutlama
Tekrar eden akıl yürütmelerde düzeni arama ve ifade etme	Örüntü tanıma ve algoritmik Düşünme

BİD çeşitli düşünme biçimlerinin karışımı olduğu konusunda bir tartışma yaratmıştır. Bir bakıma, çeşitli düşünme paradigmaları birbirleriyle "hiyerarşik" olarak ilişkilendirilebilir. Hiyerarşide daha düşük olan düşünme paradigmaları, özgünlükleri nedeniyle kullanışlıdır. Örneğin, algoritmik düşünme uygulayarak adım adım sorunları çözersiniz, bu da muhtemelen analitik düşünme veya matematiksel düşünme biçimi olabilir. Eleştirel düşünme bağlamında, tüm düşünme paradigmalarının, derinlemesine alan bilgisinden bağımsız ortak özellikleri olabilir. Bu nedenle, "Kaç ayda 28 gün vardır?" gibi basit bir soru, kişinin mantıklı, matematiksel ve belki de bilgi işlemsel düşünme yeteneklerini test etmek için kullanılabilir. Ancak açıkça, anlamaya çalıştığımız şey, "yüksek düzey" bilgi işlemsel düşünme yeteneğidir (Hu,2011).

Anlamli düşünmeyle meşgul olduğunda düşünme paradigmalarını ayırt etmek oldukça felsefi olabilir. Bu düşünceyi "analitik" veya "matematiksel" olarak etiketlemek ve bu durumda "BİD' i" gereksiz kılmak, bunun yapılamayacağını iddia etmek zordur. Tersine, önce açıklanan felsefi bir girişim, bilgi işlemeyi derinden etkileyebilir, ancak hiç "bilgi işlemsel" görünmeyebilir. Argümanın yapıldığı gibi, muhtemelen temelde aynı düşünme süreci uygulanabilir ve farklı zihinsel ürünler üretilebilir. Örneğin, matematikçiler matematiksel yapıları zenginleştirmek, daha öngörülebilir hale getirmek veya daha tamamlanmış hale getirmek için soyutlamalar veya temsiller ararlar. Bilgisayar bilimcileri ise sıklıkla ampirik nedenlerle soyutlamalar veya temsiller eklerler. Ancak tüm bunları, aynı analitik düşünme becerisini soyutlamalar veya temsiller oluşturmak ve önyargılı varsayımları terk ederken yeni fikirleri ve yaklaşımları keşfetmek için kullanabilirler. Başka bir deyişle, tamamen farklı bir öğrenme deneyiminden aynı düşünme becerisi edinebilir. Bu nedenle, belki de günümüz

bilgisayarlarının dođduđu zamandan önce bile var olan “bilgi işleme” ürünlerine odaklanırken, bugün “bilgi işleme” düşüncesinin önemini vurgulamak felsefi bir zorluk olabilir (Hu,2011).

BİD, bir muhakeme ve problem çözme yaklaşımıdır. Aslında tüm bireyler bilgi işlemsel düşünürlerdir; çünkü insan beyni doğal olarak örüntüyü tanır, algoritma oluşturur ve çözümdeki hataları giderir. BİD, matematik eğitimindeki yeri konusunda yapılan arařtırmalar, bu becerinin özellikle ilköđretim ve ortaöđretimin matematik eğitiminde önemli bir yer tuttuđunu göstermektedir. BİD, günümüz eğitiminde, öğrencilerin sorun çözme becerilerini geliřtirmede kilit bir rol oynamaktadır ve bu bağlamda matematikle yakından ilişkilidir. Çeřitli arařtırmalar, BİD'in matematiksel düşünme ve öğrenme ile etkileřimini incelemiş ve bu süreçte kullanılan araçlar, öğretim yaklaşımları ve öğrenci öğrenme sonuçlarını ele almıştır. Aslında öğrenciler, matematik, fen, sosyal bilgiler ve dil çalışmaları dahil olmak üzere çeřitli konularda BİD geliřtirebilirler (Barr & Stephenson, 2011). Fakat BİD ve matematik özellikle birbirleriyle yakından ilişkilidir. NRC, K-12 Bilim Eğitimi Çerçevesi'nin bilimsel ve mühendislik boyutları için matematik ve BİD'in temel uygulamalar olması gerektiđini önermiştir (NRC, 2013). Programlama, matematik öğreniminde pedagojik olarak faydalıdır (Eisenberg, 2002; Repenning, Webb, & Ioannidou, 2010; Wilensky, 1995; Wilensky, Brady, & Horn, 2014). Syslo ve Kwiatkowska (2014), okul matematiđinde geleneksel konuların programlamanın uygulanması yoluyla genişletilmesi ve zenginleřtirilmesi gerektiđini savunmuşlardır. Lye ve Koh (2014) tarafından yapılan literatür taraması, öğrencilerin programlama yardımıyla matematiđi iyi öğrendiklerini ortaya koymuştur (Kahn, Sendova, Sacristán, & Noss, 2011). Yadav ve arkadaşları (2014) tarafından yapılan çalışmada, öğretmenlerin BİD'in mümkün olan en fazla derse uygulanmasını tercih ettikleri bulunmuştur. BİD'in problem çözmei içermesi nedeniyle matematikte tercih edilen bir yaklaşımdır. Arařtırmalar, BİD'nin matematik öğrenimine entegre edilmesinin uygulama geliřtirme yoluyla başarıyla gerçekleştirilebileceđini belgelemektedir. Morelli ve arkadaşları (2010) tarafından yapılan çalışmada, katılımcı öğretmenler ve öğrenciler, matematik gibi geleneksel konulara BİD'i yerleřtirmeyi teşvik eden bir öğrenme uygulaması geliřtirdiler. Ke'nin (2014) çalışmasında, bir grup lise öğrencisi, Scratch kullanarak matematik oyunları tasarlamıştır ve bu, matematik öğrenim sonuçlarını başarılı bir şekilde artırmıştır. Foerster'in (2016) çalışması, Scratch ve geometri öğrenimini birleřtirerek programlamayı matematik müfredatına entegre eden bir lisede benzer sonuçları belgelemiştir Hsu ve Hu (2017) tarafından yapılan bir çalışmada, 6. sınıf öğrencileri, öğretmenler tarafından günlük yaşam matematik problemlerini çözmek için Scratch'ta bir blok program yazmak üzere BİD uygulamaya yönlendirilmişlerdir. Bu, öğrencilerin matematik öğrenme etkinliđini artırmada başarılı olmuştur. Kanıtlar, BİD'in uygulama geliřtirme yoluyla anlamlı matematik öğrenimine entegre edilebileceđini göstermektedir.

Ye ve arkadaşlarının (2023) yaptıkları arařtırmada, matematik eğitiminde BİD'in nasıl ele alınabileceđi ve bu kapsamda hangi yönlerin vurgulanabileceđi üzerine odaklanılmıştır. Arařtırma, iki düşünce tarzının nasıl entegre edileceđine dair öneriler sunmaktadır. Bu süreçler, gerçek dünya problemlerinin matematiksel veya bilgi işlemsel modellere dönüřtürülmesi, bu modeller içinde akıl yürütme ve sonuçların tekrar gerçek dünya bağlamına çevrilmesi gibi aşamalardan oluşmaktadır.

4.1.BİD'in Matematik Eğitiminde Kabul Gören Bileřenleri

BİD, problem çözme ve veri analizi gibi bilgisayar bilimi kavramlarını temsil eden genel bir düşünme becerisi ve sürecidir. Bu becerileri geliřtirmek için kullanılan temel kavramlar şunlardır:

- 1) **Problemi Parçalara Ayırma (Problem Decomposition):** Büyük ve karmařık problemleri daha küçük ve daha yönetilebilir parçalara ayırmak ve bu parçaları ayrı ayrı ele almak.
- 2) **Örüntü Tanıma (Pattern Recognition):** Verilerde veya problemlerde belirli bir yapı veya düzeni tanıyarak bu örüntüleri kullanarak çözüm geliřtirmek.
- 3) **Soyutlama (Abstraction):** Problemi basitleřtirerek gereksiz ayrıntıları atlayarak esas noktaya odaklanmak.

- 4) **Algoritmik Düşünme (Algorithmic Thinking):** Problemi parçalara ayırdıktan, desenleri tanımladıktan ve gereksiz detayları filtreledikten sonra, gerçek problemleri çözmek için adım adım bir (matematiksel) model oluşturmak.

Şekil 5

Bilgi İşlemsel Düşünmenin Basamakları- (Wing,2006; Orrill,2022)



Şekil 5, bir süreci otomatize etmek için çözümleri bir dizi adım olarak ifade etmeyi içerir (Wing,2006). BİD ile hazırlanmış bir matematik ders planı, öğrencilere matematiksel kavramları anlamaları ve uygulamaları için bilgisayar biliminin temel prensiplerini kullanma fırsatı verir. Matematiksel düşünme ile BİD'nin birleşimi, öğrencilerin problemleri çözme, analitik düşünme ve yaratıcı çözüm geliştirme becerilerini artırabilir.

5. Matematik Öğretiminde Bilgi İşlemsel Düşünme Uygulamaları ve Matematiksel Becerileri ile Entegrasyonu

Matematik derslerinde BİD becerileri çeşitli yollarla entegre edilebilir ve uygulanabilir. Kodlama aktiviteleri, simülasyon yazılımları ve veri analizi araçları gibi dijital araçlar, öğrencilerin BİD becerilerini geliştirmelerine yardımcı olur. Örneğin, GeoGebra gibi yazılımlar kullanılarak geometrik şekillerin ve fonksiyonların görselleştirilmesi sağlanabilir, bu da öğrencilerin soyut matematiksel kavramları anlamalarına yardımcı olur (Brennan & Resnick, 2012).

Öğretmenler, BİD'i matematik öğretimine dahil etmek için çeşitli yöntemler ve stratejiler kullanabilirler. Proje tabanlı öğrenme, oyun tabanlı öğrenme ve işbirlikçi öğrenme gibi yaklaşımlar, öğrencilerin aktif katılımını sağlar ve BİD becerilerini geliştirir. Ayrıca, öğretmenler öğrencilerin algoritma oluşturma, hata ayıklama ve optimize etme gibi süreçlerde deneyim kazanmalarını sağlayan etkinlikler düzenleyebilirler (Barr & Stephenson, 2011). Barr ve Stephenson BİD'in kavram ve yeterlik ve ortaöğretime entegre etmek için yaptıkları çalışmada (2011), BİD'i ortaöğretimde farklı disiplinlerde entegre etme konusunda çalışmışlar ve aşağıda yer alan Tablo2'yi matematik eğitimi için önermişlerdir.

Tablo 2

BİD Kavram ve Yetkinliklerinin Matematik Eğitime Entegrasyonu (Barr& Stephenson,2011)

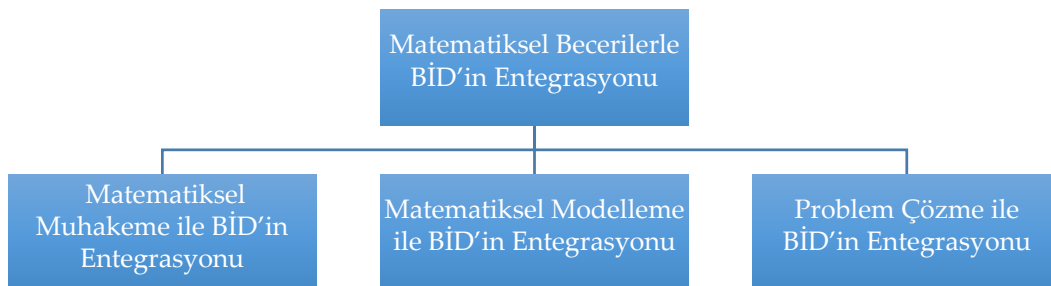
BİD Kavramı ve Yetkinliği	Matematik Eğitimi
Veri Toplama	Bir problem alanı için bir veri kaynağı bulunuz, örneğin; madeni para çevirme veya zar atma.
Veri analizi	Çevirme sayısını, zar atma sonuçlarını sayınız.
Veri temsili	Histogram, pasta grafiği, çubuk grafiği gibi grafikleri kullanarak veri temsil ediniz; listeler, grafikler vb. veri içermek için veri kümelerini kullanınız.
Problem parçalama	Bir matematiksel ifadede işlemleri uygulayınız.

Soyutlama	Cebirde deđiřkenleri kullanınız; bir kelime problemindeki temel gerekleri belirleyiniz, cebirdeki fonksiyonları inceleyiniz.
Algoritmalar ve prosedürler	Uzun bölme, arpanlara ayırma yapınız; eldeli toplama veya ıkarma işlemleri yapınız.
Otomasyon	Geometer's Sketchpad, Star Logo, Python kod paracıkları gibi araçları kullanınız.
Parallelleřtirme	Dođrusal sistemleri özünüz; matris arpımı yapınız.
Simülasyon	Kartezyen düzlemde bir fonksiyonun grafiđini iziniz ve deđiřkenlerin deđerlerini tabloyla gösteriniz.

BİD tabanlı matematik öğretim yaklaşımları ortaöğretimde daha yaygın hale geldike (Miller, 2019; Pei vd., 2018; Shumway vd., 2021), “hesaplamalı olarak geliştirilmiş matematik eğitimi” (Ng & Cui, 2021, s. 848) kavramını ve uygulamasını geliřtirmek için bir dizi tasarım tabanlı araştırma yapılmıřtır. İlgili alışmalar, öğrencilerin matematik öğrenimini kolaylařtırmak için programlama görevlerini içeren öğretim tasarımını da ilerletmiřtir. Bu alışmalar, Kotsopoulos ve diđerlerinin (2017) BİD aktiviteleri için pedagojik erevesini, kombinatorik (De Chenne & Lockwood, 2022), cebir (Bråting & Kilhamn, 2021), sayı teorisi ve matematiksel modelleme (Ng & Cui, 2021) ve geometri (Miller, 2019) alanlarında ortaöğretimde matematiksel problem özmede kullanılmaktadır. Bu alışmalar, BİD uygulamalarının matematiksel disiplin bilgilerini derinleřtirdiđini ve BİD tabanlı matematik öğrenimi bađlamının BİD becerilerini geliřtirdiđini göstermektedir (Pei vd., 2018). Lockwood ve De Chenne (2020) tarafından öne sürüldüğü gibi, programlamanın belirli matematik konularını öğrenmede etkili olduđu görölse de kâğıt-kalem yöntemine üstün olacağı veya matematiđin tüm alanlarının genel olarak biliřimle entegrasyona uygun olacağı sonucuna varılamaz. Ayrıca, öğrencilerin, programlama bađamlarında matematiksel problemleri özerken, iki farklı düşünme türü (hesaplamalı ve matematiksel) arasındaki potansiyel farklar nedeniyle eřitli zorluklar yařadıkları bulunmuřtur (Cui & Ng, 2021). Matematik Eğitimi’nin üç önemli ana becerilerinden olan muhakeme, modelleme ve problem özme ile BİD’in entegrasyonunun incelendiđi bu bölümde gelecek alışmalara yön vermesi aısından farklı bakıř aılarına literatür desteđiyle yer verilmiřtir.

řekil 6

BİD muhakeme, modelleme ve problem özme becerileri entegrasyonu



5.1. Matematiksel Muhakeme ile BİD’in Entegrasyonu

BİD, öğrencilerin matematiksel muhakeme yeteneklerini geliřtirmede önemli bir rol oynar. BİD, öğrencilerin karmařık problemleri küçük paralara ayırmalarını, desenleri tanımlarını, algoritmalar

oluşturmalarını ve soyutlamalar yapmalarını teşvik eder. Bu süreçler, matematiksel muhakemenin temel unsurlarıdır ve öğrencilerin analitik düşünme becerilerini geliştirmelerine yardımcı olur (Wing, 2006). Öğrenciler, BİD'i kullanarak matematiksel problemleri analiz etme, hipotez oluşturma ve çözümler üretme süreçlerine daha sistematik ve yapılandırılmış bir yaklaşım getirirler. Örneğin, bir problemin çeşitli çözümlerini algoritmik düşünme yoluyla test edebilir ve en etkili çözüm yöntemini belirleyebilirler. Bu da onların problem çözme becerilerini geliştirir ve matematiksel düşünme yeteneklerini derinleştirir (Grover & Pea, 2013). Bilgi işlemsel düşünme ayrıca öğrencilere soyutlama ve genel durumlara uygulanabilir çözümler üretme yeteneği kazandırır. Bu beceri, bir problemin belirli detaylarından ziyade temel prensiplerine odaklanmayı sağlar. Soyutlama sayesinde, öğrenciler benzer problemleri daha hızlı ve etkili bir şekilde çözebilirler (Barr & Stephenson, 2011).

Öğrencilerin BİD'i matematiksel muhakeme ile birleştirmesi, onların sadece problem çözme yeteneklerini değil, aynı zamanda yaratıcı düşünme kapasitelerini de artırır. Örneğin, öğrencilere gerçek dünya problemleri sunulduğunda, BİD tekniklerini kullanarak bu problemlere yenilikçi ve etkili çözümler geliştirebilirler. Bu süreçte, öğrenciler hem bireysel hem de grup çalışmaları aracılığıyla iş birliği yapma ve iletişim kurma becerilerini de geliştirme fırsatı bulurlar (Shute, Sun, & Asbell-Clarke, 2017). Ayrıca, bilgi işlemsel düşünme, öğrencilere matematiksel kavramların daha derinlemesine anlaşılmasını sağlar. Matematiksel modellerin ve algoritmaların nasıl çalıştığını anlamak, öğrencilerin matematiksel teorileri daha somut ve anlaşılır hale getirmelerine yardımcı olur. Bu, özellikle soyut matematiksel kavramların öğretiminde büyük bir avantaj sağlar (Lye & Koh, 2014).

Sonuç olarak, BİD ve matematiksel muhakemenin birleşimi, öğrencilerin analitik ve yaratıcı düşünme yeteneklerini geliştirir, problem çözme becerilerini artırır ve matematiksel kavramları daha iyi anlamalarını sağlar. Bu nedenle, eğitimde BİD'nin önemi vurgulanmalı ve bu becerilerin geliştirilmesine yönelik öğretim stratejileri benimsenmelidir.

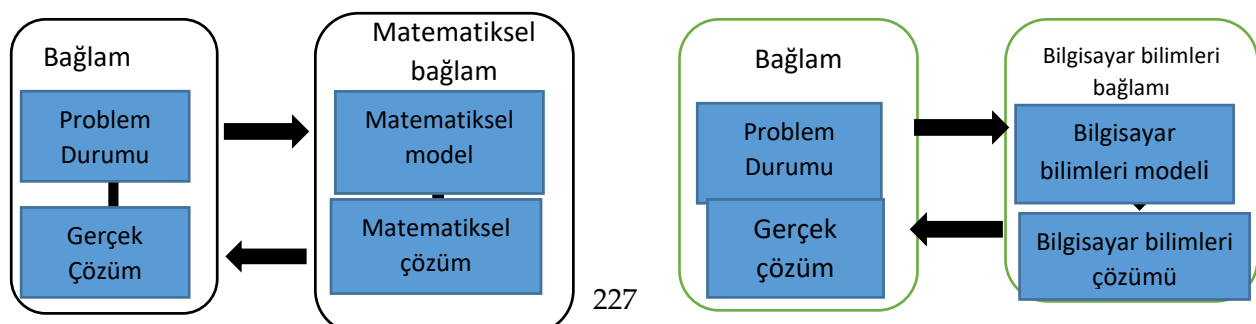
5.2. Matematiksel Modelleme ile BİD'in Entegrasyonu

Matematiksel modelleme, süreci ve okullardaki öğretim müfredatındaki yeri, son yıllarda birçok araştırmacı tarafından geniş bir şekilde tartışılmıştır. Gerçek süreç konusunda evrensel bir fikir birliği olmasa da modelleme sürecinin farklı aşamaları üzerinde genel bir anlaşma vardır ve süreci gösteren birçok farklı diyagram ve versiyon bulunmaktadır (Abrams, 2001; Berry & Houston, 1995; Biccadd & Wessels, 2011; Bloom, 2011; Borromeo Ferri, 2006; Hıdıroğlu, 2012; Lesh & Doerr, 2003; Mason, 1988; Mousoulides vd., 2006; Müller & Wittmann, 1984; Schoenfeld, 1985; Siller & Greefrad, 2010; Voskoglou, 2006). Matematiksel modelleme sürecinin önemli aşamalarını bilgi işlemsel düşünmenin önemli yönleri ile temelde ilişkilendirilebilir. Bilgi işlemsel düşünmeyle ilişkili becerilerin matematiksel modelleme için gerekli olan yetkinliklerle doğrudan uygulanabilir ve ilgili olduğu düşünülebilir (Ang & Tan, 2022).

Kallia ve arkadaşlarının (2021) yaptıkları çalışmada ise, BİD 'in matematik eğitimiyle ilişkilendirilmesi ve bu bağlamda nasıl uygulanabileceği incelenmiştir. Araştırmada, öğrencilere BİD gelişimini desteklemek için kullanılan yazılım araçları ve bu araçların matematik öğreniminde nasıl kullanıldığı ele alınmıştır.

Şekil 7

Matematiksel model ve Bilgisayar Bilimi Arasındaki Bağlantı (Kallia vd.,2011)

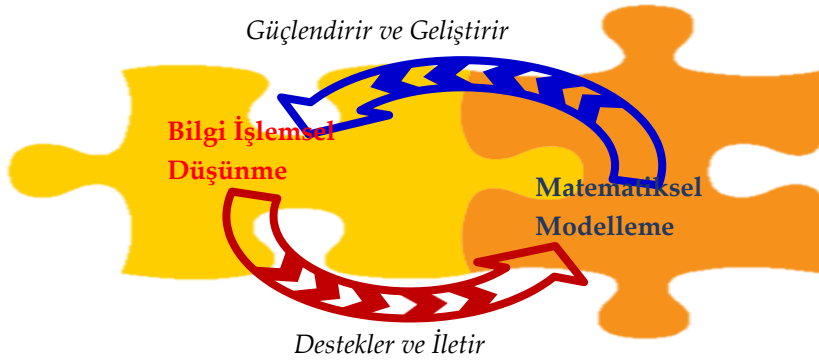


řekil 7, gerek dnya problemlerinin hem matematiksel hem de bilgisayar bilimleri bađlamında nasıl modellenebileceđini ve zmlenebileceđini gsteren bir akıř řemasını iermektedir. řema iki paralel sreci karřılařtırır: bir yanda matematiksel modelleme, diđer yanda bilgisayar bilimlerinde modelleme. İlk stunda "bađlam" altında "Problem Durumu" ile bařlanır ve bu "Gerek zm" ile bitirilir. Bu stundaki akıřın ikinci adımı "Matematik Modeli" olup, nc adım "Matematiksel zm" dr. Sađ taraftaki ikinci stun, aynı sreci bilgisayar bilimleri iin gsterir ve "Bilgisayar Bilimi İeriđi" altında sırasıyla "Bilgisayar Bilimi Modeli" ve "Bilgisayar Bilimi zm" n ierir. Her iki stunda da ilk adım, gerek dnyadan bir problemin seilmesi ve bu problemin ilgili disiplinin modelleri kullanılarak zlmesi srecini ifade eder. İinci adım, bu modelin zmnn gerek dnya bađlamına uygulanmasıdır.

Herhangi bir modelleme srecinde, gerek dnya probleminden bařlanır ve hedefimiz gerek dnya zmne ulařmaktır. Ancak, bu dođrudan yol genellikle basit deđildir ve bazen imkansızdır. Bu nedenle, gerek dnya problemini matematiksel bir probleme dnřtrmek fikri ortaya ıkar. Gerek dnyadan matematiksel dnyaya geiř sreci genellikle bir miktar soyutlamayı ierir. Problemin gereksiz kısımları ıkarılır, sadece nemli faktrler veya deđiřkenler bırakılır. Soyutlamaya rađmen, problem hala ok bykse, bir sonraki modelleme adımı bazı varsayımlar yaparak veya problemi daha ynetilebilir paralara ayırarak basitleřtirmek olacaktır. Bu sre, byk bir problemin daha kk paralara ayrılmasını sađlar. Bu, matematiksel modellemede nemli ve yaygın bir stratejidir (Ang & Tan, 2022).

řekil 8

Bilgi İřlemsel Dřnme ve Matematiksel Modelleme Arasındaki Etkileřim (Ang & Tan, 2022)



řekil 8'de, bilgi iřlemsel dřnme ve matematiksel modelleme arasındaki etkileřim gsterilmektedir. Modelleme srecindeki bir sonraki ařama, varsayımlar temelinde bir model oluřturmaya alıřmaktır. Bu ařamada, dikkate alınması veya modele dahil edilmesi gereken faktrler veya deđiřkenler belirlenmiř olacaktır. Bu modelleme srecinin yaygın bir uygulaması, deđiřkenleri ve davranıřlarını incelemek, mevcut verileri analiz etmek, eđilimleri veya rntleri belirlemek ve bazı mevcut, bilinen model veya zm yntemlerinin kullanılabilir olup olmadıđını kontrol etmektir. Bu, byk lde rnt tanıma yeteneđini gerektirir (Ang & Tan, 2022).

Sonuç olarak, matematiksel modelleme yeterliklerinin geliřimi bilgi iřlemsel dřnme ile desteklenebilir. Aynı zamanda bilgi iřlemsel dřnmenin farklı ynleri matematiksel modelleme srecinde grnmektedir.

5.3.Problem zme ile BİD'in Entegrasyonu

Matematiksel problem zme srelerinde bilgi iřlemsel dřnme nemli bir rol oynar. BİD, đrencilerin karmařık matematiksel problemleri zme srecinde kullanabilecekleri stratejiler sunar. rneđin, bir problemle karřılařtıklarında, đrenciler problemi daha kk paralara ayırabilir, desenleri tanıyabilir ve algoritmalar geliřtirerek zm retebilirler (Grover & Pea, 2013).

Bilgi işlemsel düşünme, öğrencilerin problem çözme sürecinde sistematik bir yaklaşım benimsemelerine olanak tanır. Wing (2006) tarafından vurgulandığı gibi, bilgi işlemsel düşünme, analitik düşünme becerilerini geliştirir ve öğrencilerin karmaşık problemleri daha kolay anlamalarına ve çözmelerine yardımcı olur. Örneğin, öğrenciler bir denklemi çözerken, adım adım ilerleyerek her bir adımı dikkatle analiz edebilir ve farklı çözüm yöntemlerini değerlendirebilirler.

Öğrencilerin karmaşık matematiksel problemleri çözme sürecinde bilgi işlemsel düşünme stratejilerini kullanmaları, onların analitik düşünme ve problem çözme becerilerini geliştirir. Brennan ve Resnick (2012), bilgi işlemsel düşünmenin öğrencilerin matematiksel problem çözme süreçlerinde kritik bir rol oynadığını ve bu süreçlerin öğrencilerin matematiksel düşünme yeteneklerini derinleştirdiğini belirtmektedir. Örneğin, bir öğrenci bir denklemi çözmek için çeşitli algoritmalar kullanabilir ve en uygun çözüm yöntemini belirleyebilir. Bu süreç, öğrencilerin matematiksel düşünme yeteneklerini ve problem çözme becerilerini derinleştirir. Ayrıca, bilgi işlemsel düşünme, öğrencilerin soyutlama becerilerini geliştirmelerine yardımcı olur. Soyutlama, öğrencilerin karmaşık problemleri daha basit ve yönetilebilir parçalara ayırarak çözmelerine olanak tanır. Bu, özellikle karmaşık matematiksel kavramların öğrenilmesinde büyük bir avantaj sağlar (Wing, 2006).

Bilgi işlemsel düşünme stratejilerinin kullanımı, öğrencilerin yalnızca bireysel problem çözme becerilerini değil, aynı zamanda iş birliği yapma ve takım çalışması becerilerini de geliştirmelerine yardımcı olur. Shute, Sun ve Asbell-Clarke (2017), bilgi işlemsel düşünmenin öğrencilerin grup çalışmaları sırasında daha etkili ve yaratıcı çözümler üretmelerine olanak tanıdığını belirtmektedir. Bu da öğrencilerin iletişim ve iş birliği becerilerini güçlendirir.

Öğrencilerin karmaşık matematiksel problemleri çözme sürecinde bilgi işlemsel düşünme stratejilerini kullanmaları, onların analitik düşünme ve problem çözme becerilerini geliştirir. Örneğin, bir öğrenci bir denklemi çözmek için çeşitli algoritmalar kullanabilir ve en uygun çözüm yöntemini belirleyebilir. Bu süreç, öğrencilerin matematiksel düşünme yeteneklerini ve problem çözme becerilerini derinleştirir (Brennan & Resnick, 2012).

Bu nedenle, BİD ve matematik arasındaki ilişkiler ve örtüşen bileşenler konusunda, özellikle ortaöğretim eğitimi bağlamında, yaşa uygun ve disiplinler arası yollarla BİD gelişimine büyük önem verilen yerlerde (Hong Kong Müfredat Geliştirme Konseyi, 2020; Uluslararası Eğitim Teknolojileri Derneği, 2016; Ulusal Araştırma Konseyi (NRC), 2013), bir fikir birliği eksikliği devam etmektedir. Daha fazla araştırılması gereken önemli bir soru, bilişimin ortaöğretimde matematiksel düşünme ve öğrenimini nasıl etkili bir şekilde destekleyebileceği ve bunun tersinin nasıl mümkün olabileceğidir (Hickmott vd., 2018).

Aşağıda literatürde BİD ve matematik eğitime entegrasyonu bağlamında yer alan çalışmaların (Brennan & Resnick, 2012; Grover & Pea, 2013; Weintrop vd., 2016) önerilerindeki sorulardan önemli bulunan birkaçına cevaplar yer almaktadır:

BİD, İlköğretim ve Ortaöğretim Matematik Eğitiminde Hangi Eğitim Bağlamlarında Entegre Edilmiştir?

BİD ilköğretim ve ortaöğretim matematik eğitiminde çeşitli eğitim bağlamlarında entegre edilmiştir. Özellikle problem çözme ve algoritmik düşünme becerilerini geliştirmek amacıyla, matematik derslerinde kullanılmaktadır. Örneğin, ilkokul seviyesinde basit algoritmalar ve desen tanıma faaliyetleri, ortaokul ve lise seviyesinde ise daha karmaşık veri analizi ve modelleme etkinlikleri yer almaktadır (Weintrop vd., 2016). Bu tür entegrasyonlar, öğrencilerin hem matematiksel hem de bilişsel yeteneklerini geliştirmeye yönelik yapılandırılmış etkinliklerle desteklenmektedir (Grover & Pea, 2013).

BİD Tabanlı Matematik Etkinliklerinde Hangi Araçlar ve Öğretim Yaklaşımları Kullanılmıştır?

BİD tabanlı matematik etkinliklerinde çeşitli araçlar ve öğretim yaklaşımları kullanılmaktadır. Kodlama platformları (Scratch, Python gibi), simülasyon yazılımları, robotik kitler ve veri analizi araçları (Excel, R) yaygın olarak kullanılan araçlar arasındadır (Wing, 2006). Öğretim yaklaşımları

arasında proje tabanlı öğrenme, oyun tabanlı öğrenme, sorgulamaya dayalı öğrenme ve işbirlikçi öğrenme yöntemleri öne çıkmaktadır (Brennan & Resnick, 2012). Bu yaklaşımlar, öğrencilerin aktif katılımını ve kendi öğrenme süreçlerini yönetmelerini sağlar. Ayrıca, öğrencilerin algoritma oluşturma, hata ayıklama ve optimize etme gibi süreçlerde deneyim kazanmalarına olanak tanır (Barr & Stephenson, 2011).

Hangi Teorik Yapılar veya Kavramsal Çerçeveler Kullanılmış ve Öğrencilerin BİD Tabanlı Matematik Etkinliklerindeki Öğrenme Sonuçları Nasıl Karakterize Edilmiştir?

BİD tabanlı matematik etkinliklerinde, çeşitli teorik yapılar ve kavramsal çerçeveler kullanılmıştır. Papert'in yapılandırmacı öğrenme teorisi, öğrencilerin kendi bilgi yapılarını inşa etmelerini teşvik ederken, Vygotsky'nin sosyal etkileşim teorisi, işbirlikçi öğrenme ve akran yardımlaşmasını desteklemektedir (Papert, 1980; Vygotsky, 1978). Ayrıca, Bloom'un taksonomisi ve Webb'in bilgi derinliği çerçevesi, öğrencilerin öğrenme çıktılarının değerlendirilmesinde kullanılmaktadır (Krathwohl, 2002; Webb, 1997). Öğrencilerin BİD tabanlı matematik etkinliklerindeki öğrenme sonuçları genellikle şu şekilde karakterize edilmiştir:

Problem çözme becerilerinde artış (Grover & Pea, 2013)

Algoritmik ve mantıksal düşünme yeteneklerinin gelişimi (Wing, 2006)

Matematiksel kavramlara daha derinlemesine bir anlayış (Weintrop vd., 2016)

Bilgi teknolojilerini kullanma becerilerinin artması (Brennan & Resnick, 2012)

Sonuç ve Öneriler

Bu çalışma, bilgi işlemsel düşünme (BİD) kavramının matematik eğitimi ile entegrasyonunu ve bu entegrasyonun öğretim süreçleri üzerindeki etkilerini incelemektedir.

BİD'nin matematik eğitimiyle entegrasyonu, öğrencilere problem çözme, soyutlama, algoritma geliştirme gibi becerileri kazandırmada etkili olduğu görülmüştür. BİD tabanlı etkinliklerin, öğrencilerin matematiksel kavramları daha iyi anlamalarını ve bu kavramları günlük hayatla ilişkilendirmelerini sağladığı başka bir ifade ile soyut matematiksel kavramları daha somut ve anlaşılır hale getirmelerine yardımcı olmaktadır.

BİD'nin pedagojik uygulamaları, öğretmenlerin teknoloji ve bilgisayar destekli eğitim araçlarını daha etkili kullanmalarını teşvik ettikleri belirlenmiştir. Ayrıca eğitimcilerin BİD tabanlı etkinlikleri uygularken karşılaştıkları zorluklar ve bu zorlukların nasıl aşılabileceği konusunda önemli bilgiler elde edilmiştir. Teknoloji destekli öğrenme ortamları, öğrencilerin yaratıcı problem çözme ve eleştirel düşünme becerilerini geliştirmede önemli bir rol oynadığı vurgulanmıştır.

Mevcut incelemeler, BİD ve ortaöğretim arasındaki çok yönlü bağlantıyı araştırmışken (Barcelos vd., 2018; Grover & Pea, 2013; Hickmott vd., 2018; Hsu vd., 2018; Tang vd., 2020), BİD ve K-12 matematik düşünme ve öğrenme arasındaki bağlantıyı veya etkileşimi tartışan incelemeler az sayıdadır (Shumway vd., 2021). BİD'nin ortaöğretim matematik sınıflarında nasıl uygulandığını geniş bir şekilde ele alan Hickmott ve diğerlerinin (2018) yaptıkları araştırmada BİD, analiz kategorisinde esas olarak tanımlayıcı olarak ele alınmış ve matematik eğitimi alanına açık veya örtük bağlantılar, etkinlik yaklaşımları ve öğrenci öğrenimi üzerindeki etkilerin kanıtlarını (çalışmaların nicel yöntemleri ile) içermektedir. Hickmott ve diğerleri (2016), BİD ile matematik kavramlarının öğrenimini açıkça ilişkilendiren çalışmaların eksikliğine vurgu yaparak yeni çalışmalara ihtiyaç olduğunu belirtmişlerdir. Bu durum göstermektedir ki; bu alanda yapılacak çalışmalara oldukça ihtiyaç vardır.

Bu çalışmanın bulguları, mevcut literatürdeki benzer çalışmalarla tutarlıdır. Örneğin, Wing (2006) ve Denning (2017) gibi araştırmacılar, BİD'in sadece programlama ile sınırlı kalmaması gerektiğini, aynı zamanda çeşitli disiplinlerde uygulanabilir bir düşünme biçimi olduğunu vurgulamaktadır. Bell ve Vahrenhold (2018) tarafından yapılan araştırmada, unplugged BİD yaklaşımlarının öğrencilerin problem çözme becerilerini geliştirdiğini göstermiştir. Öğretmen adaylarının BİD etkinliklerine katılımının, onların pedagojik yeteneklerini geliştirdiği ve sınıf içi uygulamalarında daha başarılı oldukları literatürde de desteklenmektedir (Grover & Pea, 2013). Bu

çalışma, BİD'in matematik eğitimi üzerindeki olumlu etkilerini ortaya koyarak bu görüşleri desteklemektedir.

Gelecek araştırmalarda, BİD etkinliklerinin farklı disiplinlerde, örneğin, fen bilimleri, mühendislik alanında nasıl uygulanabileceği ve bu uygulamaların öğrenci başarısı üzerindeki etkileri incelenebilir.

Kaynakça

- Abrams, J. P. (2001). Mathematical modeling: teaching the open-ended application of mathematics. *The Teaching Mathematical Modeling and the of Representation*. 2001 Yearbook, NCTM, (Eds. Cuoco, A.A. and Curcio, F.R.).
- Ang, K., & Tan, C. (2022). Mathematical modelling and computational thinking: Their intersections in STEM education. *Journal of STEM Education Research*, 5(3), 78-93.
- Barcelos, T. S., Rodrigues, R. A., & Carvalho, L. M. (2018). Computational thinking in K-12: An analysis of empirical literature. *Proceedings of the IEEE Frontiers in Education Conference (FIE)*, 1-9.
- Barr, V., & Stephenson, C. (2011). Bringing computational thinking to K-12: what is involved and what is the role of the computer science education community? *ACM Inroads*, 2(1), 48-54.
- Bell, T., & Vahrenhold, J. (2018). *CS Unplugged—How is it used, and does it work?* In *Adventures between lower bounds and higher altitudes* (pp. 497-521). Springer.
- Berry, J., & Houston, K. (1995). *Mathematical modeling*. London: Edward Arnold.
- Biccard, P., & Wessels, D. C. J. (2011). Documenting the development of modelling competencies of grade 7 mathematics students. *International Perspectives on the Teaching and Learning of Mathematical Modelling*. 1(5), 375-383.
- Blum, W. (2011). Can modelling be taught and learnt? Some answers from empirical research. In G. Kaiser, W. Blum, R. Borromeo Ferri, & G. Stillman (Eds.), *Trends in teaching and learning of mathematical modelling* (pp. 15–30). Dordrecht: Springer.
- Borromeo Ferri, R. (2006). Theoretical and Empirical Differentiations of Phases in the Modelling Process. In Kaiser, G., Sriraman B. & Blomhøj, M. (Eds.) *Zentralblatt für Didaktik der Mathematik*. 38(2), 86-95.
- Bråting, K., & Kilhamn, C. (2021). Programming in school mathematics: A historical epistemological perspective on the integration of programming in Swedish school mathematics. *Journal of Curriculum Studies*, 53(5), 694–710. <https://doi.org/10.1080/00220272.2021.1896132>
- Brennan, K., & Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. In *Proceedings of the 2012 annual meeting of the American Educational Research Association, Vancouver, Canada* (Vol. 1, p. 25).
- Critten, V., Hagon, H., & Messer, D. (2022). Can pre-school children learn programming and coding through guided play activities? A case study in computational thinking. *Early Childhood Education Journal*, 50(6), 969–981. <https://doi.org/10.1007/s10643-021-01233-z>
- Cui, L., & Ng, O. L. (2021). Computational thinking in mathematics education: Investigating the impact of programming on mathematical problem-solving. *Journal of Mathematical Education*, 52(1), 848.
- De Chenne, H., & Lockwood, E. (2022). Exploring students' use of computational thinking to solve combinatorial problems with Python. *Journal of Mathematical Behavior*, 66, 100944. <https://doi.org/10.1016/j.jmathb.2022.100944>
- Denning, P. J. (2005). Beyond Calculation: The Next Fifty Years of Computing. *Communications of the ACM*, 48(3), 29-32.
- Denning, P. J. (2017). Remaining trouble spots with computational thinking. *Communications of the ACM*, 60(6), 33-39.
- Eisenberg, M. (2002). Output devices, computation, and the future of mathematical crafts. *International Journal of Computers for Mathematical Learning*, 7(1), 1–44. <https://doi.org/10.1023/A:1013347104484>

- Feldhausen, R., Weese, J. L., Bean, N. H., & Bell, R. S. (2018). Collaborative learning in computer science and engineering: A multi-year study of long-term impacts. *Journal of Computing in Higher Education*, 30(1), 57–82. <https://doi.org/10.1007/s12528-018-9163-8>
- Foerster, P. (2016). Introducing computational thinking in high school mathematics: Challenges and strategies. *Mathematics Teacher*, 109(8), 611–615. <https://doi.org/10.5951/mathteacher.109.8.0611>
- Gadanidis, G., Namukasa, I., & Cendros, R. (2018). Computational thinking in mathematics teacher education. *International Journal of Information and Learning Technology*, 34(2), 133-139. <https://doi.org/10.1108/IJILT-09-2016-0048>
- Gal-Ezer, J., & Stephenson, C. (2009). Computer science teacher preparation is critical. *ACM Inroads*, 1(1), 61-66.
- Grover, S., & Pea, R. (2013). Computational thinking in K–12: A review of the state of the field. *Educational Researcher*, 42(1), 38-43. <https://doi.org/10.3390/educsci13040422>
- Hadad, R., Tang, X., Yin, Y., Lin, Q., & Zhai, X. (2020). Developing assessment tools for computational thinking in mathematics education. *International Journal of STEM Education*, 7(13), 1-15. <https://doi.org/10.1186/s40594-020-00225-6>.
- Hanid, M. F. A., Mohamad Said, M. N. H., Yahaya, N., & Abdullah, Z. (2022). Enhancing students' understanding of geometric concepts through computational thinking: A case study in secondary education. *International Journal of STEM Education*, 9(1), 110-120. <https://doi.org/10.1186/s40594-022-00323-8>
- Hıdırođlu, . N. (2012). *Teknoloji destekli ortamda matematiksel modelleme problemlerinin zm srelerinin analiz edilmesi: Yaklařım ve dřnme sreleri zerine bir aıklama* [Yayımlanmamıř yksek lisans tezi]. Dokuz Eyll niversitesi, İzmir.
- Hickmott, D., Prieto-Rodriguez, E., & Holmes, K. (2018). A scoping review of studies on computational thinking in K-12 mathematics classrooms. *Digital Experiences in Mathematics Education*, 4(1), 48-69.
- Hong Kong Curriculum Development Council. (2020). Mathematics education key learning area curriculum guide (Primary 1 - Secondary 6). Hong Kong: Education Bureau.
- Hooshyar, D., Yousefi, E., Lim, H., & Yang, Y. (2021). Development and evaluation of an adaptive educational system for improving students' computational thinking skills. *IEEE Transactions on Learning Technologies*, 14(2), 230-242. <https://doi.org/10.1109/TLT.2021.3056002>
- Hsu, T. C., & Hu, C. (2017). Applying computational thinking to mathematics education: A practical guide for teachers. *Computers & Education*, 115, 1–14. <https://doi.org/10.1016/j.compedu.2017.06.013>
- Hsu, T. C., Chang, S. C., & Hung, Y. T. (2018). How to learn and how to teach computational thinking: Suggestions based on a review of literature. *Computers & Education*, 126, 296-310.
- Hu, C. (2011). Computational thinking: what it might mean and what we might do about it. In *Proceedings of the 16th annual joint conference on Innovation and Technology in computer science education* (pp. 223-227).
- Huang, H., & Qiao, F. (2022). Exploring the integration of computational thinking in STEM education: A review of tools and practices. *Journal of STEM Education*, 23(4), 415-430. <https://doi.org/10.1007/s10956-022-09945-3>
- International Society for Technology in Education. (2016). *ISTE standards for students*. Arlington, VA: Author.
- Jiang, S., & Wong, L. H. (2022). Facilitating computational thinking through learning by teaching and game design. *Journal of Educational Computing Research*, 60(2), 355-380.
- Jocius, R., et al. (2021). Jocius, R., Goode, J., & Zhang, S. (2021). Building a virtual community of practice: Teacher learning for computational thinking infusion. *TechTrends*, 65(5), 718-727. <https://doi.org/10.1007/s11528-021-00611-8>
- Kahn, K., Sendova, E., Sacristn, A. I., & Noss, R. (2011). Developing mathematical thinking through programming activities: A case study. *International Journal of Mathematical Education in Science and Technology*, 42(4), 479–495. <https://doi.org/10.1080/0020739X.2010.550949>

- Kallia, M., van Borkulo, S. P., Drijvers, P., Barendsen, E., & Tolboom, J. (2021). Characterising computational thinking in mathematics education: a literature-informed Delphi study. *Research in mathematics education*, 23(2), 159-187.
- Kang, Y., & Lee, H. (2020). Computational thinking assessment in K-12 mathematics: Developing and validating a rubric. *Journal of Educational Technology & Society*, 23(4), 405-417.
- Karaçam, Z. (2013). Sistematik derleme metodolojisi: Sistematik derleme hazırlamak için bir rehber. *Dokuz Eylül Üniversitesi Hemşirelik Yüksekokulu Elektronik Dergisi*, 6(1), 26-33
- Ke, F. (2014). Designing and integrating purposeful learning in game play: A systematic review. *Educational Technology Research and Development*, 62(1), 57-82.
- Kong, S. C., & Abelson, H. (2019). *Computational thinking education* (p. 382). Springer Nature.
- Kotsopoulos, D., Lee, J., & Weber, K. (2017). Developing pedagogical frameworks for computational thinking in mathematics education. *Computational Thinking Journal*, 10(2), 123-137.
- Krathwohl, D. R. (2002). A revision of Bloom's taxonomy: An overview. *Theory into practice*, 41(4), 212-218.
- Lee, I., & Malyn-Smith, J. (2020). Integrating computational thinking and science in the elementary classroom. *Journal of Research on Technology in Education*, 52(1), 1-12.
- Lee, I., & Malyn-Smith, J. (2020). Integrating computational thinking and science in the elementary classroom. *Journal of Research on Technology in Education*, 52(1), 1-12.
- Lesh, R. A., & Doerr, H. (2003). Foundations of Model and Modelling Perspectives On Mathematic Teaching And Learning. In R. A. Lesh, and H. Doerr (Eds.), *Beyond Constructivism: Models and Modelling Perspectives on Mathematics Teaching, Learning and Problem Solving* (pp. 3-33). Mahwah, NJ: Lawrence Erlbaum.
- Li, Y., Schoenfeld, A. H., diSessa, A. A., Graesser, A. C., Benson, L. C., English, L. D., & Duschl, R. A. (2020). Computational thinking is more about thinking than computing. *Journal for STEM Education Research*, 3, 1-18.
- Lockwood, E., & De Chenne, H. (2020). Enriching students' combinatorial reasoning through the use of loops and conditional statements in Python. *Journal of Educational Computing Research*, 58(4), 763-784. <https://doi.org/10.1177/0735633120918173>
- Lye, S. Y., & Koh, J. H. L. (2014). Review on teaching and learning of computational thinking through programming: What is next for K-12?. *Computers in Human Behavior*, 41, 51-61.
- Mason, J. (1988). Modelling: What Do We Really Want Pupils to Learn? In D. Pimm (Ed.), *Mathematics, Teachers and Children*. (pp. 201-215). London: Hodder & Stoughton.
- Miller, D. (2019). Integrating computational thinking in secondary mathematics education: Challenges and opportunities. *Journal of STEM Education Research*, 5(1), 45-60. <https://doi.org/10.1007/s41979-019-0005-2>
- Morelli, R., Uche, C., Lake, P., & Baldwin, L. (2010). Analyzing the effectiveness of robotics to teach computational thinking. *Proceedings of the 41st ACM Technical Symposium on Computer Science Education*, 144-148. <https://doi.org/10.1145/1734263.1734314>
- Mousoulides, M., Pittalis, M., & Christou, C. (2006). Improving Mathematical Knowledge Through Modeling in Elementary Schools. In J. Novotna, H. Moraova, M. Kratka and N. Stehlikova (Eds.). *Proceedings 30th Conference of the International Group for the Psychology of Mathematics Education*, 4, 201-208.
- Müller, G., & Wittmann, E. (1984). *Der Mathematikunterricht in der Primarstufe*. Braunschweig: Vieweg.
- National Research Council. (2010). Report on computational thinking and K-12 education. National Academies Press.
- National Research Council. (2013). *Next generation science standards: For states, by states*. Washington, DC: The National Academies Press.
- Ng, O. L., & Cui, L. (2021). The integration of computational thinking and mathematical reasoning in secondary education: A case study. *Journal of Educational Computing Research*, 59(1), 45-67. <https://doi.org/10.1177/0735633120938871>

- Norris, C., Sullivan, T., Poirot, J., & Soloway, E. (2003). No access, no use, no impact: Snapshot surveys of educational technology in K-12. *Journal of Research on Technology in Education*, 36(1), 15-27.
- Papert, S. (1980). *Mindstorms: Children, Computers, and Powerful Ideas*. Basic Books, Inc.
- Papert, S. (1993). *The children's machine: Rethinking school in the age of the computer*. New York, NY, USA: Basic Books, Inc
- Papert, S. (2006). Keynote leBİDure. *Keynote at ICMI 17 Conference in Hanoi, Vietnam*. Retrieved from <http://dailypapert.com/wp-content/uploads/2012/05/Seymour-Vietnam-Talk-2006.pdf> Accessed 10 Feb 2024.
- Pei, C., Weintrop, D., & Wilkerson, M. H. (2018). Examining the role of computational thinking in mathematical problem-solving. *Educational Researcher*, 47(5), 329-338.
- Pei, F., Smith, J., & Jones, T. (2018). Computational thinking in geometry: Case studies in secondary education. *Computers & Education*, 127, 127-142. <https://doi.org/10.1016/j.compedu.2018.08.017>
- Repenning, A., Webb, D., & Ioannidou, A. (2010). Scalable game design and the development of a checklist for getting computational thinking into public schools. *Proceedings of the 41st ACM Technical Symposium on Computer Science Education*, 265-269.
- Rodríguez-Martínez, A., González-Calero, J. A., & Pérez-Pérez, C. (2020). Enhancing students' computational thinking skills: A computational experiment in a secondary school mathematics classroom. *Education and Information Technologies*, 25(2), 1455-1472.
- Schoenfeld, A. H. (1985). *Mathematical Problem Solving*. Academic Press Inc.
- Shumway, J. F., Berland, M., & Wilkerson, M. (2021). Computational thinking in K-12: In-service teacher perceptions and practices. *Journal of Research on Technology in Education*, 53(1), 63-79.
- Shute, V. J., Sun, C., & Asbell-Clarke, J. (2017). Demystifying computational thinking. *Educational Research Review*, 22, 142-158.
- Sirakaya, M., & Vural, M. (2020). The impact of computational thinking-based activities on students' problem-solving skills and attitudes towards programming. *Journal of Education and Learning*, 9(2), 115-125. <https://doi.org/10.5539/jel.v9n2p115>
- Siller, H. S., & Greefrath, G. (2010). Mathematical Modelling In Class Regarding To Technology. *CERME 6 – Proceedings of the sixth Congress of the European Society for Research in Mathematics Education*. 108-117.
- Sneider, C., Stephenson, C., Schafer, B., Flick, L., & Wolf, M. (2014). Computational thinking in high school science classrooms: Exploring the role of computers in science inquiry. *Journal of Science Education and Technology*, 23(1), 37-44. <https://doi.org/10.1007/s10956-013-9441-z>
- Stewart, M., et al. (2021). Exploring the role of computational thinking in mathematical modeling activities for high school students. *Journal of Mathematical Behavior*, 61, 100804. <https://doi.org/10.1016/j.jmathb.2021.100804>
- Swaid, S. I. (2015). Bringing computational thinking to STEM education. *Procedia Computer Science*, 65, 693-698. <https://doi.org/10.1016/j.procs.2015.09.030>
- Syslo, M. M., & Kwiatkowska, A. B. (2014). Informatics education in Europe: Are we all in the same boat? *Proceedings of the ITiCSE Conference*, 3-8.
- Tang, X., Yin, Y., Lin, Q., Hadad, R., & Zhai, X. (2020). Assessing computational thinking: A systematic review of empirical studies. *Computers & Education*, 148, 103798.
- Tekdal, M. (2021). Investigating the integration of computational thinking in Turkish STEM education. *Journal of Educational Technology & Society*, 24(4), 100-111. <https://doi.org/10.1109/EDUCON45650.2021>
- Tucker, A., McCowan, D., Deek, F. P., Stephenson, C., & Jones, J. (2006). A model curriculum for K-12 computer science: *Final report of the ACM K-12 task force curriculum committee*. ACM.
- Voskoglou, M. G. (2006). The Use of Mathematical Modelling as a Tool for Learning Mathematics. *Quaderni di Ricerca in Didattica*. 16, 53-60.
- Vygotsky, L. S. (1978). *Mind in society: The development of higher psychological processes*. Harvard University Press.

- Webb, N. L. (1997). Research Monograph Number 6: Criteria for Alignment of Expectations and Assessments in Mathematics and Science Education. Council of Chief State School Officers.
- Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L., & Wilensky, U. (2016). Defining computational thinking for mathematics and science classrooms. *Journal of Science Education and Technology*, 25(1), 127-147.
- Wilensky, U. (1995). NetLogo: An environment for simulating complex systems. Center for Connected Learning and Computer-Based Modeling, Northwestern University.
- Wilensky, U., Brady, C., & Horn, M. (2014). Fostering computational literacy in science classrooms: An agent-based approach. *Communications of the ACM*, 57(8), 24–28. <https://doi.org/10.1145/2633031>
- Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33–35. <https://doi.org/10.1145/1118178.1118215>
- Wing, J. M. (2008). Computational thinking and thinking about computing. *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 366(1881), 3717–3725. <https://doi.org/10.1098/rsta.2008.0118>
- Wing, J. M. (2011). *Research notebook: Computational thinking—what and why*. The Link Magazine, Pittsburg, PA: Computer Science. Retrieved from
- Yadav, A., Hong, H., & Stephenson, C. (2014). Computational thinking for all: Pedagogical approaches to embedding 21st-century problem-solving in K-12 classrooms. *TechTrends*, 58(6), 20–27.
- Ye, H., Liang, B., Ng, O. L., & Chai, C. S. (2023). Integration of computational thinking in K-12 mathematics education: a systematic review on CT-based mathematics instruction and student learning. *International Journal of STEM Education*, 10(1), 3
- Yıldız, M., Çiftçi, E., & Karal, H. (2017). Bilişimsel düşünme ve programlama. *Eğitim teknolojileri okumaları* (1st ed., s. 75-86).
- Yılmaz, K. (2021). Sosyal bilimlerde ve eğitim bilimlerinde sistematik derleme, meta değerlendirme ve bibliyometrik analizler. *Manas Sosyal Araştırmalar Dergisi*, 10(2), 1457-1490.
- Yuen, J., Lee, J. S. Y., & Chan, K. (2023). Impact of chatbot-assisted language learning on academic performance and motivation. *Education and Information Technologies*, 28(11), 15223–15243. <https://doi.org/10.1007/s10639-022-10879-4>.
- Zhang, L., & Nouri, J. (2019). A systematic review of learning computational thinking through Scratch in K-9. *Computers & Education*, 141, 103607. <https://doi.org/10.1016/j.compedu.2019.103607>.