

Matematik Eđitiminde Algoritma Tasarımı ve Python Programlamaya Giriş

Editör: Prof. Dr. ŞENOL DOST

Arş. Gör. ÖZGÜN ŞEFİK

Dr. SELİN URHAN





Editör: Prof. Dr. Şenol DOST

Yazarlar: Arş. Gör. Özgün ŞEFİK - Dr. Selin URHAN

MATEMATİK EĞİTİMİNDE ALGORİTMA TASARIMI VE PYTHON PROGRAMLAMAYA GİRİŞ

ISBN 978-625-6357-21-1

Kitap içeriğinin tüm sorumluluğu yazarlarına aittir.

© 2023, PEGEM AKADEMİ

Bu kitabın basım, yayım ve satış hakları Pegem Akademi Yay. Eğt. Dan. Hizm. Tic. AŞ'ye aittir. Anılan kuruluşun izni alınmadan kitabın tümü ya da bölümleri, kapak tasarımı; mekanik, elektronik, fotokopi, manyetik kayıt ya da başka yöntemlerle çoğaltılamaz, basılamaz ve dağıtılamaz. Bu kitap, T.C. Kültür ve Turizm Bakanlığı bandrolü ile satılmaktadır. Okuyucularımızın bandrolü olmayan kitaplar hakkında yayınevimize bilgi vermesini ve bandrolsüz yayınları satın almamasını diliyoruz.

Pegem Akademi Yayıncılık, 1998 yılından bugüne uluslararası düzeyde düzenli faaliyet yürüten **uluslararası akademik bir yayınev**idir. Yayımladığı kitaplar; Yükseköğretim Kurulunca tanınan yükseköğretim kurumlarının kataloglarında yer almaktadır. Dünyadaki en büyük çevrimiçi kamu erişim kataloğu olan **WorldCat** ve ayrıca Türkiye'de kurulan **Turcademy.com** tarafından yayınları taranmaktadır, indekslenmektedir. Aynı alanda farklı yazarlara ait 1000'in üzerinde yayını bulunmaktadır. Pegem Akademi Yayınları ile ilgili detaylı bilgilere <http://pegem.net> adresinden ulaşılabilir.

I. Baskı: Ocak 2023, Ankara

Yayın-Proje: Şehriban Türüldür
Dizgi-Grafik Tasarım: Müge Kuyrukcu
Kapak Tasarımı: Pegem Akademi

Baskı: Ay-bay Kırtasiye İnşaat Gıda Pazarlama ve Ticaret Ltd. Şti.
Çetin Emek Bulvarı 1314. Cadde No: 37A-B Çankaya/ANKARA
Tel: (0312) 472 58 55

Yayıncı Sertifika No: 51818
Matbaa Sertifika No: 46661

İletişim

Macun Mah. 204. Cad. No: 141/A-33 Yenimahalle/ANKARA
Yayınevi: 0312 430 67 50
Dağıtım: 0312 434 54 24
Hazırlık Kursları: 0312 419 05 60
İnternet: www.pegem.net
E-ileti: pegem@pegem.net
WhatsApp Hattı: 0538 594 92 40

Prof. Dr. Şenol DOST

1995 yılında Hacettepe Üniversitesi Eğitim Fakültesi Matematik Öğretmenliği programından mezun olmuştur. Yüksek lisansını 1998 yılında, doktorasını ise 2004 yılında Hacettepe Üniversitesi, Fen Fakültesi, Matematik Bölümü-Topoloji Anabilim Dalı'nda tamamlamıştır. 1995 yılında Hacettepe Üniversitesi'nde Araştırma Görevlisi olarak çalışmaya başlamış ve halen öğretim üyesi olarak görevini sürdürmektedir. Başlıca çalışma alanları genel topoloji, matematiksel modelleme, ileri matematiksel kavramların öğretimi, kavramsal anlama, problem çözme ve matematiksel kanıt yapmadır. Matematik ve matematik eğitimi alanında dersler vermekte, ulusal ve uluslararası projelerde yürütücü ve araştırmacı olarak yer almaktadır. Dr. Dost, Deniz'in babasıdır.



ORCID No: 0000-0002-5762-8056

Arş. Gör. Özgün ŞEFİK

2013 yılında Hacettepe Üniversitesi Eğitim Fakültesi Matematik Eğitimi programından mezun olmuştur. 2017 yılında Hacettepe Üniversitesi Fen Bilimleri Enstitüsü Matematik Ana Bilim Dalı ve Eğitim Bilimleri Enstitüsü Matematik ve Fen Bilimleri Eğitimi Ana Bilim Dalı'nda iki yüksek lisans programını tamamlamıştır. Doktorasına 2017 yılında Hacettepe Üniversitesi Eğitim Bilimleri Enstitüsü'nde başlamış ve halen devam etmektedir. 2014 yılında Hacettepe Üniversitesi'nde Araştırma Görevlisi olarak çalışmaya başlamıştır ve görevini sürdürmektedir. Başlıca çalışma alanları ileri matematiksel kavramların öğretimi, kavramsal anlama, problem çözme, matematiksel kanıt yapma ve teknoloji destekli matematik öğretimidir. Matematik ve matematik eğitimi alanında dersler vermekte, ulusal ve uluslararası projelerde araştırmacı olarak yer almaktadır.



ORCID No: 0000-0001-8680-9465

Dr. Selin URHAN



2011 yılında Hacettepe Üniversitesi Eğitim Fakültesi Matematik Eğitimi programından mezun olmuştur. Yüksek lisansını 2013 yılında, doktorasını ise 2018 yılında Hacettepe Üniversitesi Eğitim Bilimleri Enstitüsü Matematik ve Fen Bilimleri Eğitimi Ana Bilim Dalı'nda tamamlamıştır. 2014 yılında Hacettepe Üniversitesi'nde Araştırma Görevlisi olarak çalışmaya başlamıştır ve Hacettepe Üniversitesi'nde Dr. Öğretim Üyesi olarak görevini sürdürmektedir. Başlıca çalışma alanları argümantasyon, matematik kanıt yapma, matematiksel modelleme, ileri matematiksel kavramların öğretimi, kavramsal anlama işbirlikli öğrenme ve teknoloji destekli matematik öğretimidir. Matematik ve matematik eğitimi alanında dersler vermekte, ulusal ve uluslararası projelerde eğitmen ve araştırmacı olarak yer almaktadır.

ORCID No: 0000-0002-1665-7643

ÖN SÖZ

Teknolojinin yaşamımızın merkezine yerleştiği 21. Yüzyılda, dijital gerçekliđi şekillendirmek ve somut nesnelere kurduğumuz etkileşimi arttırmak için dijital dünyanın okur-yazarı olmak kaçınılmazdır. Mantıksal akıl yürütme becerilerinin gelişiminde önemli rol oynayan kodlama becerisi ise bu süreçte önemli bir bileşen olarak karşımıza çıkmaktadır.

Problem çözme becerilerinin içeriđi ve gerektirdikleri teknoloji çağının koşullarına göre yeniden yapılandırılmaktadır. Bir tür problem çözme becerisi olarak nitelendirilen ve son yıllarda bireylerin sahip olması beklenen becerilerden biri olan bilgi işlemsel düşünme, algoritma tasarımı ve kodlama ile problem çözme sürecinin arakesitinde var olan bir akıl yürütme biçimidir. Programlama temelli problem çözme etkinlikleri kapsamında bilgi işlemsel düşünmeyi desteklemek, ilköğretimden lisans düzeyine kadar çeşitli seviyelerde öğrencilerin problemlere yaratıcı çözümler bulma performanslarını geliştirmek açısından önemlidir.

Hacettepe Üniversitesi Matematik Öğretmenliđi Programında yaklaşık 25 yıldan bu yana verdiğim bilgisayar programlama derslerinin ve matematik derslerinde bu dilleri kullanarak yürüttüğüm uygulamaların yansması olarak bu kitabın yazılma fikri ve içeriđi tarafımca oluşturulmuştur. Kitapta algoritma tasarımı ve Python ile programlamaya giriş düzeyinde temel kavramlar ve örnekler sunulmuştur. Python, kolay okunabilme ve erişim kolaylıđı gibi çeşitli avantajları sayesinde, algoritma tasarımından programlamaya geçişte kod yazımının dili olarak kullanılmıştır.

Öte yandan programlama sanatına yönelik akademik çalışmalar yaptığımız bu kitabın iki yazarı da süreç içinde programlama derslerini yürütmüşler ve biriken deneyimlerini kitaba aktarmışlardır. Kitapta anlatım ve içerik bütünlüğünün sağlanabilmesi için kitabı oluşturan tüm bölümler iki yazar tarafından ortaklaşa yazılmıştır.

Kitap beş bölümden oluşmaktadır. Matematik eğitiminde problem çözme ve bilgi işlemsel düşünme kavramlarının programlama bağlamında ilişkileri birinci bölümde ele alınmıştır. İkinci bölümde problem çözmede algoritma tasarımı süreci anlatılmıştır. Algoritma tasarımından programlama diline geçiş için gerekli olan Python'un temel araçları üçüncü bölümde ele alınmıştır. Dördüncü bölümde ise Python ile programlamanın nasıl yapıldığı sunulmuştur. Son olarak beşinci bölümde Python'da fonksiyon oluşturmaya yer verilmiştir.

Bu kitap içerdiği programlamaya dayalı problem çözme etkinlikleri ile öğrenme ortamlarının tasarımcısı olan öğretmenler için bir rehber niteliđi taşımakla

birlikte, Eğitim Fakültelerinin ilköğretim ve ortaöğretim matematik öğretmenliği programlarında yer alan “Algoritma ve Programlama” dersine yönelik ders kitabı olma özelliği taşımaktadır. Kitabımızın matematik öğretmeni ve öğretmen adaylarının üretme ve öğretme tecrübelerine yenilikler getirmesini, onlara farklı bakış açıları kazandırmasını, matematik ve matematik eğitimi alanında yeni kapılar açacak yaratıcı fikirler için ilham kaynağı olmasını dileriz.

Editör

Prof. Dr. Şenol Dost

BEYTEPE-2023

ORCID No: 0000-0002-5762-8056

İÇİNDEKİLER

Yazarlar Hakkında.....	iii
Ön Söz.....	v

1. BÖLÜM

PROBLEM ÇÖZME YAKLAŞIMI OLARAK PROGRAMLAMA

Bilgi İşlemsel Düşünme.....	1
Programlama Nesneleri.....	3
Programlama Stratejileri.....	5
Problem Çözme ve Programlama Süreçleri.....	7
Matematik Eğitiminde Problem Çözme ve Programlama.....	9

2. BÖLÜM

ALGORİTMA VE BİLGİSAYARLA PROBLEM ÇÖZME

Algoritma.....	17
Doğrusal Mantık Yapısı.....	25
Karar Mantık Yapısı.....	26
Döngü Yapısı.....	30

3. BÖLÜM

PYTHON PROGRAMLAMA DİLİNİN TEMEL ARAÇLARI

Programlamanın Genel Yapısı ve Tanımlamalar.....	38
Python'a Giriş.....	39
Komutların Çağrılması.....	44
Python'da Veri Türleri.....	46
String Veriler.....	46
Sayısal Veriler.....	49
Sayıların Özellikleri ve Python Operatörleri.....	53
Değişkenler ve Atamalar.....	58
Cebirsel İfadeler ve Denklemler.....	60
Matematiksel Fonksiyonlar.....	66
Listeler ve Demetler.....	70
Diziler ve Matrisler.....	75

4. BÖLÜM

PYTHON İLE PROGRAMLAMA

Giriş ve Çıkış Komutları.....	83
Karşılaştırma Deyimi.....	86
Döngüler.....	91
For Döngüsü.....	91
While Döngüsü.....	96

5. BÖLÜM

PYTHON FONKSİYONLARI

Python'da Fonksiyon Yazma.....	108
Kaynakça.....	125
Dizin.....	127

1. BÖLÜM

PROBLEM ÇÖZME YAKLAŞIMI OLARAK PROGRAMLAMA

Yeni doğanlara dijital yerli, yaşlılara dijital göçmen denilen, yetişkinlerin ise bu iki topluluk arasında kendine bir yer bulmaya çalıştığı, yazılımlara dayalı dijital araçlarla çevrili bir dünyada yaşıyoruz. İnsanın dünyayı kontrol etme ve bu dijital ekosistemde anlamlı bir varoluş sergileme mücadelesinin mutlu sonla bitmesi dijital dünyanın dilinden anlamasına ve algoritmik akışı bilinçli şekilde yönetmesine bağlıdır. Dijital ekosistemin yalnızca tüketicisi değil aynı zamanda üreticisi olmak ise kod okuryazarlığı olarak adlandırılan daha teknik bir beceri gerektirmektedir. Günlük hayatta dijital gerçekliği şekillendirmek ve somut nesnelere kurduğumuz etkileşimi arttırmak için, dijital dilden anlayabilmek ve dijital dünyanın okuryazarı olmak gerekmektedir.

Dijital dünyanın temel kavramlarından “kodlama”, programlama sürecinde programın fiziksel olarak yazıldığı aşama; “programlama” ise, kodlama aşamasının yanı sıra tasarım, algoritma geliştirme, test etme, hata ayıklama, belgeleme ve devam ettirme gibi aşamalar içeren bir süreç olarak ele alınabilir. Bir kişi bilgisayarların ve diğer makinelerin dilini okuyabiliyor ve yazabiliyor ise kod okuryazarı olarak adlandırılır. Kod okuryazarlarının bir problemin çözüm sürecinde yürüttükleri akıl yürütme eylemlerinin temelinde bilgi işlemsel düşünme yer almaktadır.

Programlama, karmaşık görevleri anlamayı ve problem çözmeyi içeren bilgi işlemsel düşünme yeteneğini ortaya çıkarmaya ve geliştirmeye yönelik temel bir yoldur. Diğer yandan, bilgi işlemsel düşünme direkt programlama görevi içermeyen farklı türde problemleri çözmeye sürecinde de ortaya çıkabilir. Bu bağlamda, bilgi işlemsel düşünme herkesin günlük hayatında ihtiyaç duyduğu okuma ve yazma gibi temel beceriler arasında düşünülmelidir.

1.1. Bilgi İşlemsel Düşünme

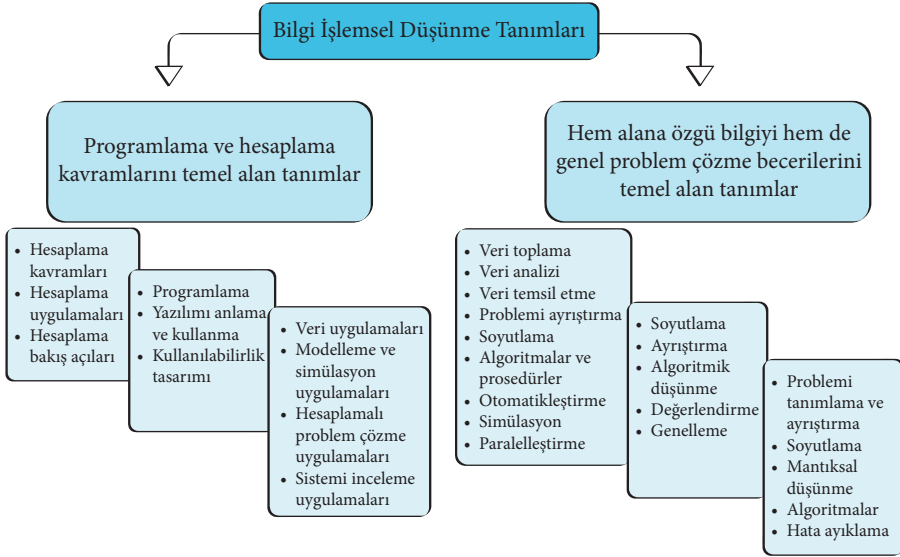
Papert (1980) tarafından ortaya atılan bilgi işlemsel düşünme kavramı ilk kez Jeanette Wing (2006) tarafından “Problem çözmeye bir yaklaşım, sistem tasarlama, programlamaya temel oluşturan kavramları kullanan insan davranışlarını anlama” şeklinde tanımlanmıştır. Papert bilgi işlemsel düşünmeyi direkt olarak

bilgisayar ile ilişkilendirmekte ve bilgisayarı “düşünmek için bir nesne” olarak görmektedir. Wing’in perspektifinden ise bilgi işlemsel düşünme, bilgisayarı kullanma ile sınırlı değildir; etkili ve verimli problem çözme için gerekli olan genel bir düşünme becerisidir.

Bilgi işlemsel düşünme, bilgisayarın düşünme tarzının kopyalandığı ve mekanik işlemlerin tekrarlandığı bir süreç değildir. İnsanın zekasını ve hayal gücünü kullandığı; bu sayede bilgisayarların düşünme tarzını ve kapasitesini geliştirdiği; problemleri çözmeye, iletişim ve etkileşim becerilerini düzenlemeye odaklı mantıksal bir düşünme sürecidir. Bilgi işlemsel düşünme sürecine ilişkin anlayışlar iki kategoride ele alınabilir:

- i. *Programlama ve hesaplama kavramlarını temel alan tanımlar*
- ii. *Hem alana özgü bilgiyi hem de genel problem çözme becerilerini temel alan tanımlar*

Bilgi işlemsel düşünme sürecine yönelik tanımlarda yapılan vurguların sınıflandırılmasına ilişkin diyagram Şekil 1’de (Tang ve diğ., 2020) yer almaktadır.



Şekil 1. Bilgi işlemsel düşünme kavramına ilişkin yapılan tanımların sınıflandırılması

Tanımlara bakıldığında bilgi işlemsel düşünmenin temel kavramlarının yanında;

- problemleri, bilgisayar ve diğer araçların kullanımı ile çözülebilecek şekilde formüle etme

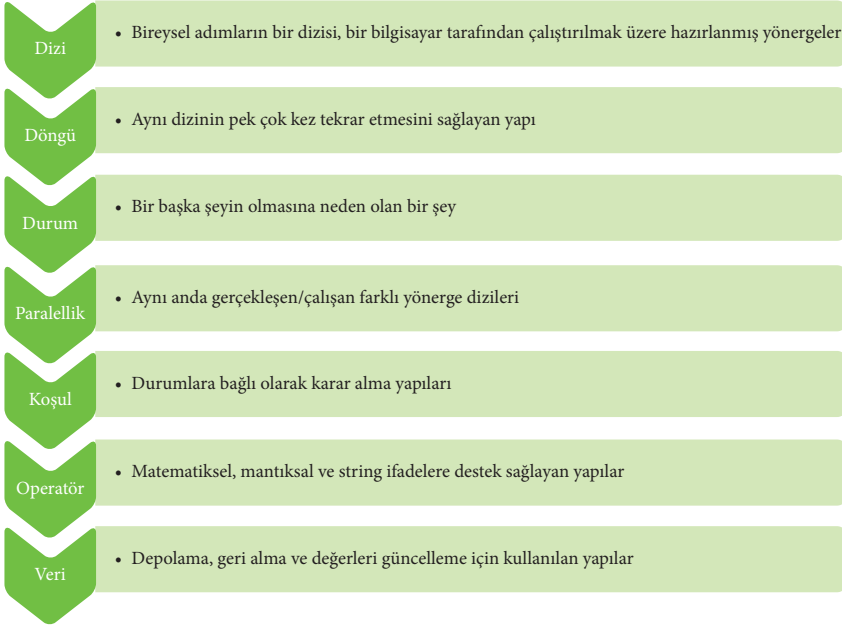
- verileri mantıksal biçimde organize ve analiz etme,
- verileri modeller ve simülasyonlar gibi soyutlamalar ile temsil etme,
- algoritmik düşünme yoluyla problemlerin çözümlerini sıralı adımlar halinde organize etme
- adımların ve kaynakların en etkili ve verimli kombinasyonunu yapmak için olası çözümleri tanımlama,
- çözüm yollarını analiz etme ve uygulama,
- problemin çözüm sürecini genelleştirme ve farklı problemlere transfer etme (Computer Science Teachers Association (CSTA) ve International Society for Technology in Education (ISTE), 2011)

gibi işlevsel özelliklere sahip olduğu görülmektedir. Bilgi işlemsel düşünmeyi, bir görevi yerine getirmek amacıyla bilgisayarla ve diğer bazı araçlarla yürütülen özel bir problem çözme türü olarak programlama nesnelere ve stratejileri bağlamında açıklamak faydalı olacaktır.

1.1.1. Programlama Nesnelere

Problem çözme geniş bir kümede ele alındığında matematiksel düşünme, bilgi işlemsel düşünme gibi süreçler bu kümenin altında yer alır. Her düşünme sürecinin kendine ait nesnelere ve bu nesnelere ile yürütülen problem çözme stratejileri vardır. Örneğin bir matematik problemini çözmek için nesnelere matematiksel kavramlar olan problem çözme stratejilerinin genelleştirilmesi, matematiksel düşünme ile problem çözme yaklaşımını ortaya çıkarır. Nesnelere bilgisayar kodları ile oluşturulan yapılar olan programlama stratejileri ile matematiksel bir problemi çözme süreçlerinin genelleştirilmesi ise bilgi işlemsel düşünme ile problem çözme yaklaşımını oluşturur.

Bir bilgisayar programı ile bir problemin çözümünde kullanılan programlama nesnelere aşağıdaki gibi tanımlanmıştır.



Şekil 2. Programlama Nesneleri (Brennan ve Resnick, 2012)

Dizi, programlamada bir görevi yerine getirmek için yürütülecek adımların ve yönergelerin amaca uygun sıralanmış halidir. Bir reçeteye benzer olarak, üretilmesi gereken bir davranış ya da eylemi ortaya çıkaracak programlama yönergelerinin mantık çerçevesinde sıralanması ile elde edilen anlamlı bir bütündür. Örneğin, klavyeden girilen bir n doğal sayısına kadar olan doğal sayılardan 2'ye tam bölünenleri bir listeye aktaran komut zinciri n doğal sayısına kadar olan çift sayıları veren bir diziyi temsil eder.

Döngüler, aynı dizinin pek çok kez gerçekleşmesini sağlayan bir yapıdır. Örneğin, klavyeden girilen bir n doğal sayısının asal olup olmadığı denetlenir. Bu durumda, sayının 1 ve kendisinden başka pozitif bölüneni olup olmadığı belirlenmesi gerekmektedir. Bunun için, n sayısının sırasıyla 2,3,4, ..., $n-1$ sayılarına tam bölünüp bölünmediği kontrol edilmelidir. Bu durum, dizinin oluşturulması sürecinde döngü kurulmasına ihtiyaç olduğu anlamına gelmektedir.

Bir şey bir başka şeyin oluşmasına neden oluyorsa bu şeye durum denir. Örneğin bir sayının 2'ye bölümünden kalan 0 ise ekrana bu sayının çift sayı olduğunun yazdırılması şeklinde oluşturulan bir komut zincirinde “sayının 2'ye bölümünden kalanın 0 olması” bir durumdur.

Paralellik farklı yönerge dizilerinin aynı anda çalışması anlamına gelmektedir. Örneğin terimlerinde üs alma ve faktöriyel işlemlerini aynı anda bulunduran

$\frac{x}{2!} - \frac{x^3}{4!} + \frac{x^5}{6!} - \dots$ serisinin klavyeden girilen bir $0 < x < 1$ gerçel sayısı ve $5 < N < 30$ doğal sayısı için N tane teriminin toplamı hesaplınsın. Bu amaçla öncelikle üs alma ve faktöriyel işlemlerine ilişkin ayrı ayrı komut dizileri oluşturulsun. Son aşamada sonlu toplamı hesaplamak için bu komut dizilerinin aynı anda çalıştırılması diziye paralellik kavramını dahil etmektedir.

Birtakım şartlara dayalı olarak karar alma yapılarına koşul denir. Döngüler için sunulan örnek durum dikkate alınırsa bir n sayısının asal sayı olup olmadığını belirlemek için önceden tanımlanan bir kontrol değişkenine bir başlangıç değeri verilsin. n sayısının 2, 3, 4, ..., $n - 1$ sayılarına bölümünden kalanların bulunduğu döngüde eğer kalan 0 ise kontrol değişkeninin değeri 1 arttırılsın. Burada “kalanın 0 olması” kontrol değişkeninin değerinin 1 artması için bir koşuldur. Sürecin devamında “kontrol değişkeninin başlangıç değerinde kalmış olması”, “ n sayısının asal olması” durumunun koşuludur.

Programcıların sayısal ve string manipülasyonlar yapmasını sağlayan matematiksel, mantıksal ve string ifadelere operatör denir. Çeşitli programlama dillerindeki

$$+ , - , * , / , ^$$

gibi yapılar operatörlere örnek olarak verilebilir.

Veri yapısı depolama, geri çağırma ve değerleri güncellemeyi sağlar. Genellikle veriler değişkenler ve listeler yoluyla saklanır. Daha sonraki bölümlerde detaylı değinileceği gibi değişkenler tek bir sayı ya da string saklayabilir; liste ise bir grup sayı ya da bir grup string saklayabilir. Örneğin 2 ile bir n doğal sayısı arasındaki asal sayıların bulunması amaçlanan bir görevde n sayısına kadar olan sayıların sırasıyla 1 ve kendisi haricindeki sayılara bölümünden kalanın 0 olup olmadığını belirlemek için bir döngü oluşturulsun. Bu döngüde başlangıç değeri 0 olan ve asal olup olmadığı denetlenen sayının 1 ya da kendisi haricinde bir sayıya bölümünden kalan 0 ise değeri bir artacak bir değişken tanımlandığını düşünelim. Burada kullanılan bu kontrol değişkeni veri kavramı için bir örnektir. Aynı algorithmada 2 ile n sayısı arasında denetlenen sayılardan asal olanları tespit ettikçe bir L listesinde depoladığımızı düşünelim. Burada kullanılan L listesi de veriye örnektir.

1.1.2. Programlama Stratejileri

Programlama yapmak bir kavramı tanımlamak, tasarım için bir plan yapmak ve sonrasında kodlarla tasarımı uygulamak şeklinde yürüyen ve ardışık ilerleyen bir süreç değildir. Küçük adımlarla yapılandırılması gereken ve çözüm yaklaşımı-

na bağlı olarak değişime uğrayabilecek olan bir modelin üretildiği ve bu modelin olası bazı değişimlere bağlı olarak uyarlanabilir nitelikte olması gereken bir modelleme sürecidir.

Bir tasarım sürecinde olaylar çok nadiren planlandığı gibi işler. Bir çeşit tasarımcı olan programcı için de durum böyledir. Önemli olan kritik anlarda kendiliğinden ortaya çıkan ya da tespit edilen problemlerle ilgili stratejiler geliştirmektir. Bir problemi bilgisayar programlama ile çözerken programlama nesnelерinin merkezde yer aldığı birtakım stratejiler tanımlanmıştır. Bu stratejiler programlama ile problem çözmeye sürecinde kullanılan ve bilgi işlemsel düşünme süreçlerinin çatısını oluşturan süreçlerdir. Şekil 3'te alanyazında tanımlanan, programlama yaparken kullanılan bazı stratejiler verilmiştir.



Şekil 3. Programlama yaparken kullanılan stratejiler (Ng ve Cui, 2021)

Programlama sürecinde programlama nesnelерini matematiksel nesneler ile birlikte amaca uygun ve doğru şekilde kullanarak kod dizileri oluşturmaya modelleme denir. Kodları amaca uygun bir mantık çerçevesinde bir görevi yerine getirmek üzere sıralı adımlar halinde yapılandırmak algoritmik düşünmeyi gerektirir. Algoritmik düşünme aynı zamanda algoritmaları anlamayı, uygulamayı ve değerlendirmeyi içerir. Soyutlama ve modülerleştirme programcının çözüm

üretmek istediği problem durumunu detaylarına ayrıştırması ve bu detayları ayrı ayrı incelemesidir. Bu durum, programcının modelini küçük bölümlerden oluşan anlamlı bir bütün olarak oluşturmasını sağlar. Bu strateji, programcının modelinin farklı bölümlerini daha kolay yapılandırmasını ve kontrol etmesini sağladığı gibi, programın başkaları tarafından daha kolay anlaşılmasına da fırsat verir. Bir programcının daha önceden üretilmiş modelleri geliştirerek ya da güncelleyerek başka bir model tasarlaması mümkündür. Programlamada yeniden karıştırma ve yeniden kullanma stratejisi olarak adlandırılan bu süreç, programcıların birbirlerinin modellerine ulaşmasına fırsat veren bir teknoloji ağı ile yürütülebilir. Test etme ve hata ayıklama, programcının önceden yapılandırılan modellerden transfer yaparak ya da bir uzmandan destek alarak kendi modelindeki eksik ya da hatalı adımları ortaya çıkardığı ve düzeltmeye çalıştığı stratejidir. Bu strateji kimi zaman deneme-yanılma yoluyla geliştirilen bir uygulama niteliğinde olabilir.

1.2. Problem Çözme ve Programlama Süreçleri

Programlamayı bir problem çözme olarak değerlendirmek eğitimde bilgi işlemsel düşünme alanında bu perspektifin ön plana çıkmasını sağlamıştır. Arzu edilen hedefe ulaşmak anlamına gelen problem çözümü, problemin anlaşılmasını ve aktarılmasını, bir planın geliştirilmesini, planlanan çözümün izlenmesini, uygulanmasını ve ilgili düşünme süreçlerini düzenlemeyi gerektirmektedir. Polya (1945) problem çözme sürecinde kullanılan adımları şöyle belirlemiştir:

- i. Bir problemi analiz yoluyla anlamak
- ii. Problemi çözmek için bir araç veya plan tasarlamak
- iii. Çözümü uygulamak
- iv. Çözümü değerlendirmek

Benzer şekilde, Mayer ve Wittrock (2006), problem çözme sürecini

- i. Anlama ve temsil etme
- ii. Planlama ve izleme,
- iii. Yürütme
- iv. Öz-düzenleme

adımları ile yapılandırmışlardır. Bu süreçte, problemi alt problemlere ayırmak, araç-amaç analizi yapmak, problemi geriye doğru işlemek ve örneklerle problem durumunu daha anlaşılır hale getirmek gibi stratejiler uygulanır.

Programlama ile problem çözme süreci, hesaplamalı soyutlamalar kullanılarak problemde verilenlerin temsil edilmesini ve problemler için yaratıcı çözümler geliştirilmesini sağlar. Bu bağlamda programlama, bilgisayar bilimi kavram ve