

PROGRAMLAMA KAVRAMLARI

Programlama: Bilgisayar programlarının yazılması, test edilmesi ve bakımının yapılması sürecine verilen isimdir.

Programlama Dili: Programcının bilgisayara hangi veri üzerinde işlem yapacağını, verinin nasıl depolanıp iletileceğini, hangi koşullarda hangi işlemlerin yapılacağını tam olarak anlatmasını sağlar. Bir programlama dili, insanların bilgisayara çeşitli işlemler yaptırmasına olanak sağlayan her türlü simge, karakter ve kurallar grubudur.
Örnek: Python, C#, C++, Java, Swift, GoLang

Yazılımcı: Bir programlama dili kullanarak, yazılım (program) üreten kişilerdir.

Algoritma: Belli bir problemi çözmek veya belirli bir amaca ulaşmak için tasarlanan ve izlenecek olan yoldur.

Neden Programlama Öğrenmeliyiz?

Bilgisayara bir işi yaptırırken yani programlarken düşünmeyi öğreniriz. Analiz yapabilme ve problem çözebilme yeteneğimiz artar. Bir sorunla karşılaştığımızda daha iyi düşünebiliriz ve o sorun için bulduğumuz çözümlerin sayısı artar. Olaylara daha ayrıntılı bakabiliriz. Bilgisayarda program yazarken algoritmalar geliştiririz.

Algoritma problemin adım adım çözülme sürecidir. Bu da bize olayları detaylı düşünebilme ve çözebilme yeteneği kazandırır.

ETİK DEĞERLER

Etik: Bireylerin ahlaklı ve erdemli bir hayat yaşayabilmesi için hangi davranışlarının doğru, hangilerinin yanlış olduğunu araştıran bir felsefe dalıdır.

Bilişim Etiği: Bireylerin bilişim teknolojilerini ve interneti kullanımı sırasında uymaları gereken kuralları tanımlayan ilkelere bilişim etiği denir.

Fikri Mülkiyet: Kişinin kendi zihni tarafından ürettiği her türlü ürün olarak tanımlanmaktadır.

Fikri ve kültürel eserlerden bazıları **Creative Commons (CC)** organizasyonuna dahildir. Creative Commons, telif hakları konusunda esneklik sağlamayı amaçlayan, eser sahibinin haklarını koruyarak, eserlerin paylaşımını kolaylaştırıcı modeller sunan, kar amacı gütmeyen bir organizasyondur. Bu organizasyona dahil olan eserler, kaynağı belirtmek ön şartıyla belirli kısıtlamalar göz önünde bulundurularak kullanılabilir.

Bilişim dünyasında yazılımları lisanslarına göre, özgür yazılımlar ve ticari yazılımlar olmak üzere ikiye ayırabiliriz. Özgür yazılım dünyasına ait GPL'ye (General Public Licence - Genel Kamu Lisansı) sahip yazılımlar ücretsiz kullanılabilirken, ticari faaliyet gösteren firmaların ürettiği yazılımların lisanslarıysa çoğunlukla yüksek bedeller karşılığında alınabilmektedir.

Lisanssız yazılım kullanmanın etik uygunsuzluk yanında hukuki yaptırımları ve teknik sakıncaları da vardır.

Uluslararası Bilgisayar Etik Enstitüsüne göre bilişim teknolojilerinin doğru bir şekilde kullanılabilmesi için aşağıda belirtilen 10 kurala uyulması gerekmektedir.

1. Bilişim teknolojilerini başkalarına zarar vermek için kullanmamalısınız.
2. Başkalarının bilişim teknolojisi aracılığı ile oluşturduğu çalışmalarını karıştırmamalısınız.
3. Başkasına ait olan verileri incelememelisiniz.
4. Bilişim teknolojilerini hırsızlık yapmak için kullanmamalısınız.
5. Bilişim teknolojilerini yalancı şahitlik yapmak için kullanmamalısınız.
6. Lisanssız ya da kırılmış/kopyalanmış yazılımları kullanmamalısınız.
7. Başkalarının bilişim teknolojilerini izinsiz kullanmamalısınız.
8. Başkalarının bilişim teknolojileri aracılığı ile elde ettiği çalışmalarını kendinize mal etmemelisiniz.
9. Yazdığınız programların ya da tasarladığınız sistemlerin sonuçlarını göz önünde bulundurmalısınız.
10. Bilişim teknolojilerini her zaman saygı kuralları çerçevesinde kullanmalı ve diğer insanlara saygı duymalısınız.

ALAN ADLARI

Adresler incelendiğinde,



Bu adresin Türkiye Cumhuriyeti'ne (.tr) ait bir devlet/hükümet (.gov) sitesi olduğu görülebilir.



Bu adresin de Türkiye Cumhuriyeti'nde (.tr) faaliyet gösteren bir vakıf ya da derneğe (.org) ait olduğu anlaşılabilir.

Alan adı uzantıları birçok internet sitesi için fikir verebilir:

- .com ve .net → Ticari firma ,
 - .gov → Resmi kurum,
 - .org → Dernek veya Vakıf ,
 - .edu → Üniversite,
 - .k12 → Eğitim kurumu (anaokulu, ilkokul, lise),...
- Ülke kodu:** İnternet adreslerinin son kısmı ülke kodudur.
Ülke kodu .tr olan siteler daha güvenilirdir.

İNTERNET ETİĞİ

İnternet kullanımı ile ilgili olarak dikkat edilmesi gereken etik ilkeler; kişilik hakları, özel yaşamın gizliliği ve veri güvenliği gibi başlıklar altında incelenebilir. İnternet ortamında uyulması gereken etik kurallar aşağıda verilmiştir.

- 1: İnternet'te karşılaştığımız ancak yüzünü görmediğimiz, sesini duymadığımız kişilere saygı kuralları çerçevesinde davranmalıyız.
- 2: İnternet'i kullanırken her kültüre ve inanca saygılı olmak, yanlış anlaşılacak davranışlardan kaçınmak gerektiği unutulmamalıdır.
- 3: Özellikle sosyal medya, sohbet ve forum alanlarındaki kişiler ile ağız dalaşı yapmaktan kaçınmalı, başka insanları rahatsız etmeden yazışmaya özen göstermeliyiz.
- 4: Sürekli olarak büyük harfler ile yazışmanın İnternet ortamında bağırarak anlamına geldiği unutulmamalıdır.
- 5: İnsanların özel hayatına karşı saygı göstererek kişilerin sınırlarının internet ortamında paylaşılmamasına dikkat edilmesi gerektiği unutulmamalıdır.

6: İnternet'te kaba ve küfürlü bir dil kullanımından kaçınarak gerçek hayatta karşımızdaki insanlara söyleyemeyeceğimiz ya da yazamayacağımız bir dil kullanmamalıyız.

7: İnternet'i başkalarına zarar vermek ya da yasa dışı amaçlar için kullanmamalı ve başkalarının da bu amaçla kullanılmasına izin vermemeliyiz.

8: İnternet ortamında insanların kişilik haklarına özen göstererek onların paylaştığı bilginin izinsiz kullanımından kaçınmamız gerektiği de unutulmamalıdır.

Siber (dijital) Zorbalık: İnternet ortamında başkalarından kaynaklanan kötü davranışlara, internet etiğine uymayan davranışlara denir. Siber zorbalığa maruz kalınması durumunda yapılması gerekenler:

1: Zorbalık yapan hesaplara cevap vermeyiniz, onlarla tartışmaya girmeyiniz. İlk yapmanız gereken, zorbalık yapan hesabı engellemektir.

2: Bu hesapları, bulunduğunuz sosyal medya platformundaki "Bildir/Şikayet Et" bağlantısını kullanarak şikayet ediniz.

3: Size yönelik etik dışı davranışlar artarak ve ağırlaşarak devam ederse bunların ekran görüntülerini ve mesajları kaydediniz. Bu kanıtlarla birlikte ailenizin ya da rehber öğretmeninizin gözetiminde hukuki yollara başvurunuz.

4: Siber zorbalığa maruz kalan başka kişiler de olabilir. Böyle durumlarda bu kişilere ne yapmaları gerektiği konusunda yardımcı olabilir, kötü kullanım bildirimini siz de yapabilirsiniz.

BİLGİ GÜVENLİĞİ

Bilgi Güvenliği: Kişisel ya da kurumsal düzeyde bizim için büyük önem teşkil eden her tür bilgiye izin alınmadan ya da yetki verilmeden erişilmesi, bilginin ifşa edilmesi, kullanımı, değiştirilmesi, yok edilmesi gibi tehditlere karşı alınan tüm tedbirlere bilgi güvenliği denir.

Siber Suç: Bilişim teknolojileri kullanılarak gerçekleştirilen her tür yasa dışı işlemdir.

Siber Zorbalık: Bilgi ve iletişim teknolojilerini kullanarak bir birey ya da gruba, özel ya da tüzel bir kişiliğe karşı yapılan teknik ya da ilişkisel tarzda zarar verme davranışlarının tümüdür.

Siber Savaş: Farklı bir ülkenin bilgi sistemlerine veya iletişim altyapılarına yapılan planlı ve koordineli saldırılardır.

SAYISAL DÜNYADA KİMLİK VE PAROLA YÖNETİMİ

Parola: Bir hizmete erişebilmek için gerekli olan, kullanıcıya özel karakter dizisidir.

Şifre: Sanal ortamdaki verilerin gizliliğini sağlamak için veriyi belirli bir algoritma kullanarak dönüştüren yapıdır.

Parola, bilgi güvenliğinin en önemli ögesidir. Parolanın da ele geçirilmesi durumunda oluşacak zarar, bir evin anahtarını ele geçiren hırsızın sebep olacağı zarardan çok daha fazla olabilir. Parolanın kötü niyetli kişiler tarafından ele geçmesi durumunda:

1: Elde edilen bilgiler yetkisiz kişiler ile paylaşılabilir ya da şantaj amacıyla kullanılabilir.

2: Parolası ele geçirilen sistem başka bir bilişim sistemine saldırı amacıyla kullanılabilir.

3: Parola sahibinin saygınlığının zarar görmesine yol açabilecek eylemlerde bulunulabilir.

4: Ele geçirilen parola ile ekonomik kayba uğrayabilecek işlemler yapılabilir.

5: Parola sahibinin yasal yaptırım ile karşı karşıya kalmasına yol açabilir.

Güçlü bir parolanın belirlenmesi için aşağıdaki kurallar uygulanmalıdır.

- Parola, büyük/küçük harfler ile noktalama işaretleri ve özel karakterler içermelidir.
- Parola, -aksi belirtilmedikçe- en az sekiz karakter uzunluğunda olmalıdır.
- Parola, başkaları tarafından tahmin edilebilecek ardışık harfler yada sayılar içermemelidir.
- Her parola için bir kullanım ömrü belirleyerek belirli aralıklar ile yeni parola oluşturulması gerekir.

KİŞİSEL BİLGİSAYARLARDA VE AĞ ORTAMINDA BİLGİ GÜVENLİĞİ

Teknolojinin hızlı ilerleyişi ile birlikte gelen güvenlik riskleri ve insanların bu konudaki yetersiz farkındalıkları bilgisayar ve İnternet kullanımı sırasında pek çok tehlikenin ortaya çıkmasına neden olmaktadır.

Bilişim sistemlerinin çalışmasını bozan veya sistem içinden bilgi çalmayı amaçlayan Virüs, Solucan, Truva Atı ya da Casus yazılım gibi kötü niyetlerle hazırlanmış yazılım veya kod parçaları zararlı programlar olarak adlandırılır. Bu zararlı programlar:

- İşletim sisteminin ya da diğer programların çalışmasına engel olabilir.
- Sistemdeki dosyaları silebilir, değiştirebilir ya da yeni dosyalar ekleyebilir.
- Bilişim sisteminde bulunan verilerin ele geçirilmesine neden olabilir.
- Güvenlik açıkları oluşturabilir.
- Başka bilişim sistemlerine saldırı amacıyla kullanılabilir.
- Bilişim sisteminin, sahibinin izni dışında kullanımına neden olabilir.
- Sistem kaynaklarının izinsiz kullanımına neden olabilir.

Virüsler: Bulaştıkları bilgisayar sisteminde çalışarak sisteme ya da programlara zarar vermek amacıyla oluşturur. Virüsler bilgisayara e-posta, bellekler, İnternet üzerinden bulaşabilir. Bilgisayarın yavaşlaması, programların çalışmaması, dosyaların silinmesi, bozulması ya da yeni dosyaların eklenmesi virüs belirtisi olabilir.

Solucanlar: Kendi kendine çoğalan ve çalışabilen, bulaşmak için ağ bağlantılarını kullanan kötü niyetli programlardır. Sistem için gerekli olan dosyaları bozarak bilgisayarı büyük ölçüde yavaşlatabilir ya da programların çökmesine yol açabilir.

Truva Atları: Kötü niyetli programların çalışması için kullanıcının izin vermesi ya da kendi isteği ile kurması gerektiği için bunlara Truva Atı denmektedir. Truva Atları saldırganların bilişim sistemi üzerinde tam yetki ile istediklerini yapmalarına izin verir.

Casus Yazılımlar: İnternet'ten indirilerek bilgisayara bulaşan ve gerçekte başka bir amaç ile kullanılsa bile arka planda kullanıcıya ait bilgileri de elde etmeye çalışan programlardır.

Zararlı Programlara Karşı Alınacak Tedbirler:

- Bilgisayara antivirüs ve İnternet güvenlik programları kurularak bu programların sürekli güncel tutulmaları sağlanmalıdır.
- Tanınmayan/güvenilmeyen e-postalar ve ekleri kesinlikle açılmamalıdır.
- Ekinde şüpheli bir dosya olan e-postalar açılmamalıdır. Örneğin resim.jpg.exe isimli dosya bir resim dosyası gibi görünse de uzantısı exe olduğu için uygulama dosyasıdır.
- Zararlı içerik barındıran ya da tanınmayan web sitelerinden uzak durulmalıdır.
- Lisanssız ya da kırılmış programlar kullanılmamalıdır.
- Güvenilmeyen İnternet kaynaklarından dosya indirilmemelidir.

Ardışık Sudoku

Her satırda, her sütunda ve kalın çizgilerle çevrili her 2x3'lük alanda 1'den 6'ya rakamlar bir kez yer alacak şekilde diyagramı doldurun. Tüm ardışık komşuların arasında bir siyah nokta vardır.

Örnek

•	•	•	•	•	6
•	•	•	•	•	1
•	•	•	•	•	•
•	•	•	•	•	•
•	•	•	•	•	1
•	•	•	•	•	•
4	•	•	•	•	•

5	1	3	2	4	6
6	4	2	1	5	3
2	3	6	5	1	4
1	5	4	3	6	2
3	6	1	4	2	5
4	2	5	6	3	1

•	•	•	•	•	•
•	•	•	•	•	•
•	•	•	•	•	•
•	•	•	•	•	•
•	•	•	•	•	•
•	•	•	•	•	•
3	•	•	•	•	•

•	•	•	•	•	•
•	•	•	•	•	•
•	•	•	•	•	•
•	•	•	•	•	•
•	•	•	•	•	•
•	•	•	•	•	•
9	•	•	•	•	•

•	•	•	•	•	•
•	•	•	•	•	•
•	•	•	•	•	•
•	•	•	•	•	•
•	•	•	•	•	•
•	•	•	•	•	•
•	•	•	•	•	•



[Ardışık Sudoku Nasıl Çözülür:](https://yureklis.wordpress.com/2015/12/24/ardisik-sudoku-nasil-cozulur/)

<https://yureklis.wordpress.com/2015/12/24/ardisik-sudoku-nasil-cozulur/>

PROBLEM ÇÖZME KAVRAMLARI VE YAKLAŞIMLAR

Programlama: Bilgisayar programlarının yazılması, test edilmesi ve bakımının yapılması sürecine verilen isimdir.

Bilgi işlemsel düşünme: Bilgisayar biliminin kavramlarından yararlanarak problem çözme, sistem tasarlama ve insan davranışlarını anlama olarak tanımlanabilir.

Sorunu daha biçimsel bir şekilde yeniden ifade etmek, bir problemi anlamak için mükemmel bir tekniktir. Birçok programcı, diğer programcıları bir sorunu tartışmak için arar; sadece diğer programcıların yanıtı olabileceğini düşünür fakat aynı zamanda problemi yüksek sesle ifade etmek genellikle yeni ve yararlı düşünceleri tetikler.

PROBLEM ÇÖZME TEKNİKLERİ

Her Zaman Bir Planınız Olsun: Belirsiz bir durumu yaşamak yerine her zaman bir planınız olmalıdır. Bu, en önemli kuraldır. Belki oluşturduğunuz çözüm planı ilk denemelerde sonuç vermeyecek ama her seferinde sizi çözüme biraz daha yaklaştıracak ipuçları elde etmenizi sağlayacaktır.

Problemi Tekrar İfade Edin: Önceki problemlerde de gördüğümüz üzere bazen problemi tekrar ifade etmek, göremediğimiz bir ayrıntıyı görmemizi ya da problemi daha kolay çözmek adına bir ipucu yakalamamızı sağlayabilir. Hatta bazen probleme ilişkin bir yanlış anlamının ortaya çıkmasına ya da hedefin daha iyi anlaşılmasına neden olur.

Problemi Küçük Parçalara Ayırın: Verilen problemi adımlara ya da bölümlere ayırmak, çözümü kolaylaştırır. Bir problemi iki bölüme ayırdığımız düşünüldüğünde, her bir parçanın çözümünün tümünü çözmeye göre yarı yarıya kolaylaştırdığını düşünebiliriz.

Önce Bildiklerinizden Yola Çıkın: Programlama yaparken öncelikle bildiklerimiz ile başlamalı ve sonra yeni çözümler arayışına girmeliyiz. Problemi küçük parçalara bölerek çözebildiğiniz parçadan başlayınız.

Problemi Basitleştirin: Çözmekte zorlandığınız bir problemle karşılaşırsanız problemin kapsamını daraltmayı deneyebilirsiniz. Bunun için koşulları azaltmayı ya da çözebileceğiniz biçime dönüştürmeyi, değişkenleri azaltmayı ya da problemin kapsama alanını küçültmeyi düşünebilirsiniz.

Benzerlikleri Arayın: Burada ele aldığımız benzerlik kavramı, çözülmesi istenen problemle önceden çözülen problem arasındaki olası örtüşme ya da yeni çözüme ilham verme olarak tanımlanabilir. Benzerlik, farklı biçimlerde karşımıza çıkabilir. Bazen problemler aynı, değişkenler ya da veriler farklıdır. Bazen problemin belirli bir bölümü başka bir problemle benzerlik gösterebilir.

Deneme Yapın: Bazen bir problemi çözenin en kolay yolu denemek ve sonuçlarını gözlemlemektir. Bu, tahmin etmekten çok farklıdır. Bir çözümü tahminen öngörmek ile kodu yazarak denemek ve sonuçlarını incelemek çok farklı sonuçlar verir. Böylece problemi çözebilmek için gereken ipuçlarını elde edebilirsiniz.

Asla Vazgeçmeyin: Asla vazgeçmemek, kişisel bir özelliktir. Kararlılık, güven ve istek kaybolduğu zaman acık düşünemezsiniz, işlemler olması gerektiğinden uzun sürer ve gittikçe zorlaşır. Hatta öfke ve kızgınlığa bile dönüşebilir.

PROBLEM ÇÖZME ADIMLARI

- 1. Problemi Tanımlama:** Problemi çözmeye başlamadan önce problemin acık, anlaşılır ve çok doğru bir şekilde tanımlanmış olması gerekir. Problemin ne olduğunu bilemezseniz onu çözemezsiniz.
- 2. Problemi Anlama:** Çözüme doğru yol almadan önce problemi çok iyi anladığınızdan emin olmanız gerekir. Problemin neler içerdiğini ve kapsamını doğru anlamalısınız.
- 3. Problemin Çözümü İçin Farklı Yol ve Yöntemler Belirleme:** Problemin çözümü için olabildiğince farklı yol ve yöntem belirlemeli ve bu listenin, tüm olasılıkları içerdiğinden emin olmalısınız.
- 4. Farklı Çözüm Yolları Listesi İçerisinden En İyi Çözümü Seçme:** Bu adımda her bir çözümün olumlu ve olumsuz yönlerini ortaya koymalısınız.
- 5. Seçilen Çözüm Yolu ile Problemi Çözmek İçin Gerekli Yönergeleri Oluşturma:** Bu adımda numaralandırılmış ve adım adım yönergeler oluşturmanız gerekir.
- 6. Çözümü Değerlendirme:** Çözümü test etmek ya da değerlendirmek, sonucun doğruluğunu kontrol etmek anlamına gelir. Sonucun doğru olması ve problemi olan bireyin beklentilerini karşılama düzeyi önemlidir. Sonuç yanlış çıkmış ya da bireyin beklentilerini karşılamamış ise problem çözme sürecine baştan başlamak gerekir.

Sayfa 47'deki Şekil 1.7: Problem çözme adımları görselini inceleyiniz.

PROBLEM TÜRLERİ

Problemlerin her zaman sıradan çözümleri olmaz. Kek yapmak ya da araba kullanmak gibi problemleri çözmek için bir dizi eylem gerekir. Adım adım yönergelere dayalı olan bu çözümlere “**algoritmik çözümler**” denir. En iyi yolu seçtikten sonra sonuca, ilgili adımları izleyerek ulaşılır. Bu adımlardan oluşan yapıya “**algoritma**” denir.

En lezzetli ekmeği seçmek ya da işleri büyütme için yatırım yapmak gibi problemlerin ise acık ve net ifade edilen yanıtları yoktur. Bu çözümler bilgi ve deneyim gerektirir, bir dizi deneme ve yanılma sürecinden oluşur. Doğrudan işlem adımları ile ulaşılamayan sonuçlara “**keşfe dayalı çözümler**” denir.

VERİ TÜRLERİ

Sayısal Veri: Sayısal veri, hesaplama işlemlerinde kullanılabilen tek veri türüdür. Pozitif ya da negatif tam sayılar ve reel sayılar kullanılabilir. Sayısal veriler; acılar, uzaklık, nüfus, ücret, yarıçap gibi hesaplama sürecinde gerekli değerler için tanımlanır. Banka hesap numarası ya da posta kodu gibi sayısal ama hesaplama için kullanılmayan veriler de vardır. Bu tür veriler sayısal olarak tanımlanmaz.

Veri Türü	Veri Seti	Örnek	Python'daki Veri Türü
Sayısal: Tam Sayı	Tüm sayılar	66578 -250	integer (int)
Sayısal: Reel Sayı	Tüm reel sayılar ve ondalıksayılar	-56,23 3,56	float

Karakter Veri (Alfanümerik): Karakter veri seti; tüm tek haneli sayılar ("0".. "9"), harfler ("a".. "z", "A".. "Z") ve özel karakterleri ("#", "&", "*", ..) kapsar. Bu veri setinden oluşturulan değer, tırnak içinde belirtilir. Büyük ve küçük harf duyarlıdır yani "a" ile "A" farklı algılanır. Karakterler sadece sayıdan oluşsa bile hesaplama işlemlerinde kullanılamaz.

Veri Türü	Veri Seti	Örnek	Python'daki Veri Türü
Karakter	Tüm rakamlar, harfler ve özel semboller	"A", "Y", "k", "i", "6", "0", "+", "%"	string
Karakter Dizisi	Birden fazla karakterden oluşan kombinasyon	"Bilgisayar", "532-5556633"	string

Mantıksal Veri: Mantıksal veri, veri setinde yalnızca iki kelime barındırır: doğru ve yanlış. Bu veri evet ya da hayır şeklindeki karar verme süreçlerinde kullanılır. Örneğin elde edilen değer, beklenen değer mi, evli mi, arabası var mı, öğrenci lise mezunu mu gibi sonucu kesin doğru ya da yanlış olan durumlarda mantıksal veri tanımlaması yapılır.

Veri Türü	Veri Seti	Örnek	Python'daki Veri Türü
Mantıksal	Doğru/Yanlış True/False	2<3 => True 6>8 => False	Boolean (bool)

Kitabınızın 51.sayfasındaki örnekleri inceleyiniz.

Bilgisayar veriyi saklar? Bilgisayar veriyi hafızada saklar. Her bir değişken için hafızada belirli bir alan ayrılır ve bu alan her seferinde tek bir değer saklayabilir. Kullanıcı, var olan değer yerine yeni bir değer atadığında eski değer silinir. Hafızada bu konumlar geçicidir. Programın çalışması bittiğinde ya da bilgisayar kapatıldığında bu veriler silinir. Verilerin daha sonra tekrar kullanılması gerekiyorsa sabit disk gibi kalıcı bir konuma kaydedilmeleri gerekir.

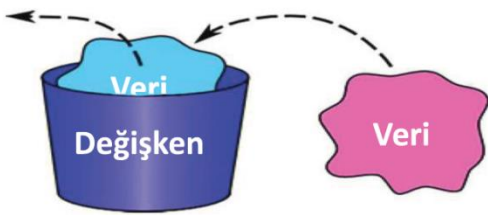
SABİT VE DEĞİŞKENLER

Sabit: Bilgisayarlar problemleri çözmek için süreç boyunca sabit ve değişken olarak adlandırılan verileri kullanır. "Sabit" olarak tanımlanan veriler problemin çözüm süreci boyunca asla değişmeyen değerlerdir. Sabit değerler sayısal, karakter ya da özel semboller olabilir. Bu durumda bu değere bilgisayarın hafızasında bir yer ayrılır ve bir isim verilir. Program çalıştığı sürece bu değer kendisine verilen isim ile çağrılır ve değeri asla değiştirilemez. Örneğin, pi değeri değişmeyen bir değer olacağı için sabit olarak tanımlanmalıdır.

Değişken: Sabitlerin tam tersi şekilde bir "değişken" tanımlandığında değeri, program çalıştığı sürece değişebilir. Değişkenlere taşıdığı değerleri ifade eden isimler verilir, bu şekilde belirleyici özellikleri de oluşur. Programcılar çözüm sürecinde ihtiyaç duyulan her bir değişkene ayrı bir isim vermelidir. Böylece bilgisayar bu ismi, ilgili değeri hafızada bulmak için kullanır. Değişken, farklı veri türlerinde olabilir ancak ismi, içerdiği değer ile tutarlı olmalıdır. Örneğin fiyat isimli bir değişkenin içerisinde 50 değeri atanmış olabilir, program çalıştığı süre içerisinde bu değer değişebilir ancak değişkenin ismi hiçbir zaman değişmez.

Değişken isimlendirilirken dikkat edilmesi gerekenler:

1. Değişkene içerdiği değer ile tutarlı isimler veriniz.
2. Değişkenlere isim verirken boşluk kullanmayınız.
3. Değişkenlere isim verirken bir karakter ile başlayınız.
4. Matematiksel semboller kullanılmaya dikkat ediniz.
5. Bazı platformlar desteklemediği için Türkçe karakter kullanımı tavsiye edilmez.
6. Programlama dillerinde kullanılan komut isimleri değişken olarak kullanılamaz. Çok bilinenleri; if, for, while, else, do, int, vb.
7. Değişken isimlendirmelerinde boşluk karakteri yerine alt çizgi (_) karakteri kullanılabilir ancak değişken isimlendirmede genellikle küçük harfle başlanır ve ikinci bir kelime yazılacaksa ilk kelimenin hemen ardından büyük harfle devam edilir. Örnek: tcKimlikNo
8. Özel karakterler değişken isimlerinde kullanılamaz (*,/, -,+, #,%,&,(,=,?,\$,[,,{ gibi...).



taksitSay = 9 (*int*)

adSoyad = "Ali Kara" (*string*)

değişken

değer

? Sizce hayali bir program içerisinde aşağıdaki değerleri tutacak olan değişkenleri nasıl isimlendirebiliriz?

Hesap Numarası, Okul Numarası, Kitap Sayısı, Doğum Tarihi, Anne Kızlık Soyadı, 1.Yazılı Notu

FONKSİYONLAR

Fonksiyonlar, belirli işlemleri yürüten ve sonuçları döndüren bir işlem kümesidir. Her programlama dili, içerisinde kendine özgü fonksiyonlar barındırır. Bu fonksiyonlar kütüphanesi, programlama dili bilgisayara göre değişiklik gösterir. Ayrıca pek çok programlama dili, programcıların kendi fonksiyonlarını yazmalarına da olanak verir.



Fonksiyona gönderilen verilere **“parametre”** denir. Fonksiyonlar parametreleri değiştirmez ama işlemlerde kullanır. Örneğin karekök fonksiyonunu ele alalım. $\text{Sqrt}(N)$, gönderilen N değeri için karekök değeri hesaplamaktadır. Sqrt fonksiyonun ismi, N işlem yapılacak veri yani parametredir.

Sayfa 54’de bulunan örnek fonksiyonları ve çıktılarını inceleyiniz.

```
>>> import math
>>> math.sqrt(81)
```

```
>>> isim = "Saime İnal Savi"
>>> len(isim)
```

```
>>> print("merhaba dünya")
```

```
>>> math.sqrt(abs(-144))
```

```
>>> abs(-40)
```

```
>>> math.cos(0)
```

```
>>> a = "5"
>>> type(a)
```

```
>>> a = "5"
>>> int(a)
```

Fonksiyonları iç içe de kullanabildiğimize dikkat edin.

OPERATÖRLER

Bilgisayara, verileri nasıl işleyeceğini belirtmek gerekir. Bu işlem için operatörler kullanılır. “Operatörler” verileri, ifade ve eşitlikler ile birleştirir. Bu yazım, aynı zamanda operatörler bilgisayara ne tür bir işlem (matematiksel, mantıksal vb.) olduğuna dair bilgi verir.

Operatörler; **matematiksel**, **mantıksal** ve **ilişkisel** operatörler olarak sınıflandırılabilir.

Matematiksel Operatörler

Operatör	Python’daki Sembolü	İşlem	Sonuç
Toplama	+	5.1+8.2	13.3
Çıkarma	-	8.5-5.0	3.5
Çarpma	*	4*3	12
Bölme	/	12/4	3
Modül Alma	%	12%3	0

Modül, bir sayının başka bir sayıya bölümünden kalandır.

$$\begin{array}{r|l} 3 & 5 \\ \hline 0 & 0 \\ \hline 3 \text{ kalan} & \end{array}$$

$$\begin{array}{r|l} 33 & 5 \\ \hline 30 & 6 \\ \hline 3 \text{ kalan} & \end{array}$$

$$\begin{array}{r|l} 48 & 5 \\ \hline 45 & 9 \\ \hline 3 \text{ kalan} & \end{array}$$

```
>>> 3%5
3
>>> 33%5
3
>>> 48%5
3
>>> 12%3
0
```

Mantıksal Operatörler

Operatör	Python'daki Sembolü	İşlem	Sonuç
Eşittir	==	2==1	False
Küçüktür	<	2>1	True
Büyüktür	>	2<1	False
Küçük ya da eşittir	<=	2<=1	False
Büyük ya da eşittir	>=	2>=1	True
Eşit değildir	!=	2!=1	True

Operatörlerle yapılan işlemlerin sonucunda ortaya mantıksal değer olarak Doğru (True) ya da Yanlış (False) çıkar.

İlişkisel Operatörler

Operatör	Python'daki Sembolü	İşlem	Sonuç
Değil	NOT	not 1==1	False
Ve	AND	1==1 and 2==2 1==1 and 1==2	True False
Veya	OR	1==1 or 1==2 3==1 or 1==2	True False

AND operatörünün kullanıldığı bir ifadenin sonucunun True olabilmesi için ifadedeki tüm önermelerin sonucunun True olması gerekir.

OR operatörünün kullanıldığı bir ifadenin sonucunun True olabilmesi için ifadedeki önermelerden **birinin** True olması yeterlidir.

İŞLEM ÖNCELİĞİ

Matematiksel, mantıksal ve ilişkisel operatörlerin bir hiyerarşisi yani öncelikleri vardır. İşlemler, bu sıralamaya göre yapılmaz ise sonuç, beklendiği gibi çıkmayabilir. En içteki ayraçtan en dıştakine doğru işlem yapılmalı, ayraç içerisinde ise işlem önceliklerine dikkat edilmelidir.

() hiyerarşiyi sıralar, ayrıca içerisindeki işlemler en içten en dışa doğru yapılmalıdır.

Matematiksel Operatörler

** Kuvvet (Üs)
% Mod
*, /
+, -

İlişkisel Operatörler

==, <, >, <=, >=, !=

Mantıksal Operatörler

NOT
AND
OR

İFADE ve EŞİTLİKLER

Şimdiye kadar yaptığımız işlemlerde hep ifadeleri kullandık. İfadeler bizlere sonuç döndürür ancak bu sonuçlar hafızada saklanmaz. İfadelerin sonuçlarını ileride program içerisinde yeniden kullanabilmek için **eşitliklerden** faydalanırız.

İFADELER	EŞİTLİKLER
A + B A ve B sayısal veridir. Sonuç sayısaldır ve hafızada korunmaz.	C = A + B A, B ve C sayısal veridir. Sonuç sayısaldır ve C değişkenine atanarak korunur.
A < B A ve B sayısal, karakter ya da dizi olabilir. Sonuç mantıksal değerdir ve hafızada korunmaz.	C = A < B A, B ve C sayısal, karakter ya da dizi olabilir. Sonuç mantıksal değerdir ve C değişkenine atanarak korunur.
A OR B A ve B mantıksal veridir. Sonuç mantıksaldır ve hafızada korunmaz.	C = A OR B A, B ve C mantıksal veridir. Sonuç mantıksaldır ve C değişkenine atanarak korunur.

ÖDEV

```
>>> a = 5
>>> b = 3
>>> not a>b
```

```
>>> a = 3
>>> b = 2
>>> a>b or b>4
```

```
>>> a = 3
>>> b = 2
>>> a>b and b>4
```

```
>>> a = 5
>>> b = 3
>>> a*b < 20
```

```
>>> a = 5
>>> b = 3
>>> not a>b or b==5
```

```
>>> x = 3
>>> y = 4
>>> x*y > x**2
```

```
>>> a = 5
>>> b = 3
>>> not a>b or b==5
```

```
>>> a = 4
>>> b = 3
>>> a%b == 1
```

```
>>> a = 4
>>> b = 2
>>> a**b == b**a
```

```
>>> a = 4
>>> b = 2
>>> a**b != b**a
```

```
>>> kısaKenar = 4
>>> uzunKenar = 6
>>> alan = kısaKenar*uzunKenar
>>> alan
```

```
>>> x = 20
>>> y = 15
>>> z = (x*y)%2
>>> z == 0
```

```
>>> x = 3
>>> y = 4
>>> z = x*y
>>> z >=x**y or x<4
```

```
>>> x = 3
>>> y = 4
>>> (x*y)%5
```

```
>>> a = 4
>>> b = 2
>>> a!=b or 2<1
```

```
>>> a = 3
>>> b = 4
>>> a!=b and 2<1
```

```
>>> a = 3
>>> b = 4
>>> a!=b and not b<a
```

```
>>> x = 3
>>> y = 4
>>> (x**y)%5
```

4. ALGORİTMALAR

Yukarıdaki çizelgeler geliştirildikten sonraki adım, yapılacak işlemleri bilgisayarın anladığı dilde yazabilmektir. Bu yönergeler “**algoritma**” olarak adlandırılır.

“**Sözde kod**” algoritmaya çok benzer bir dildir ve bazen algoritma yerine kullanılabilir. Algoritmayı oluşturmak, bilgisayarda problem çözme sürecinin en zor bölümüdür. Modüller etkileşim çizelgesinden ve süreç GSC çizelgesinden alınır. Algoritmadaki işlem sayısı, programcının problemi çözme yoluna bağlıdır.

5. AKIŞ ŞEMALARI

Problem çözme sürecimiz, bilgisayarın iletişim kurma yöntemi ile şekillenir. **Algoritma**, bilgisayara hangi işlemi hangi sırada yapması gerektiğini söyleyen yönergeler bütünüdür.

Akış şeması ise algoritmanın görsel gösterimidir. Programcı, oluşturulan algoritmadan grafiksel gösterimler oluşturur. Akış şeması, program geliştirmeye başlamadan önceki son adımdır. Akış şemasında hatalar rahatlıkla görülüp düzeltilebilir. Akış şemalarını oluşturmak için kullanılan evrensel semboller ve bu her bir simgenin anlamı vardır.

Akış şeması, bir problem çözümünün başlangıcından bitişine kadar olan süreci gösterir. Akış şeması içerisindeki her bir simge, algoritmadaki bir işlemi ifade eder. Genellikle işlemler tek yönlü olmasına rağmen karar kutularından iki farklı ok çıkar. Bir karar simgesinden çıkan ok, bazı işlemlerin tekrarlanmasını sağlayabilir; böylece bir “döngü” oluşur.

ALGORİTMA YÖNERGELERİ VE AKIŞ ŞEMASI SEMBOLLERİ

Simge	İşlev
	Başla/Bitir
	Giriş
	Atama/İşlem
	Denetim (Karar)
	Çıkış
	Döngü
	Akış Yönü
	Bağlaç
	Önceden Tanımlı İşlem/Fonksiyon

Şekil 1.10: Akış şeması sembolleri

Akış şemalarını oluştururken dikkat edilmesi gereken bazı noktalar şunlardır:

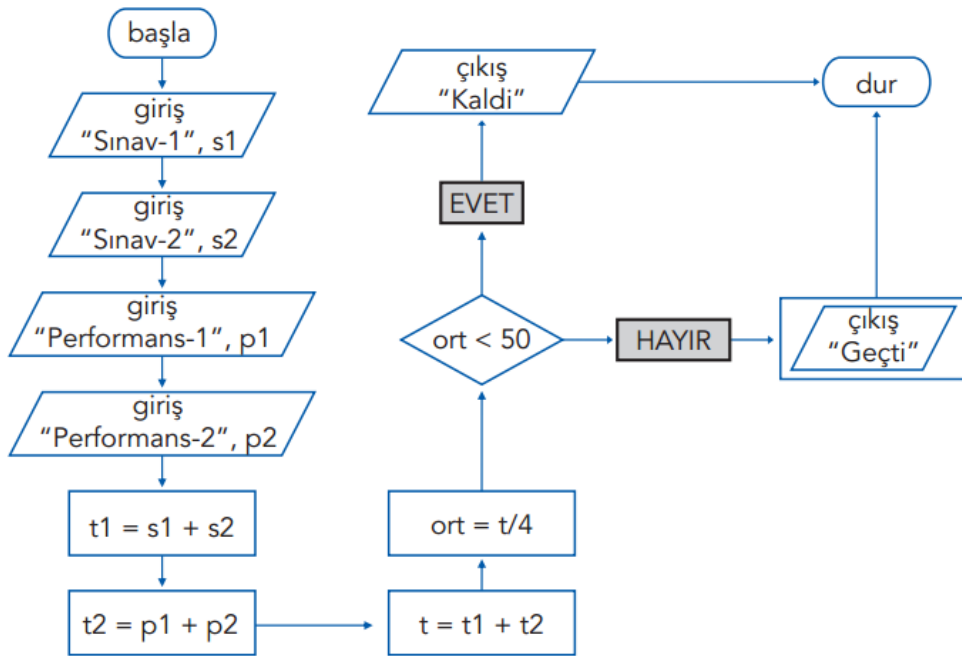
1. Yönergeler, simgelerin içine yazılmalıdır.
2. Hatırlatıcı bilgiler simgenin yanına yazılabilir. Böylece akış şeması ek açıklamalı bir şemaya dönüşür.
3. Bir akış şeması her zaman sayfanın başından başlar ve sonuna doğru gider. Eğer bir sayfaya sığmazsa bir ya da daha fazla bağlantı simgesi kullanılarak diğer sayfaya geçilebilir.
4. Akış şemasını çizmek için uygun yazılımlar kullanılırsa daha standart bir görünüm elde edilir.
5. Simgeler, içeriğindeki yazının rahatça okunabileceği kadar büyük yapılmalıdır.

ÇÖZÜMÜN PROGRAMLANMASI/KODLANMASI

Algoritmalar ve akış şemaları tamamlandıktan sonra istenilen bir programlama dili kullanılarak programın yazılması işlemine geçilir ki bu işleme "programlama" ya da "kodlama" adı verilir. Kodlama sonucunda programın ne kadar hatasız çalıştığı, algoritmanın etkililiğine bağlıdır.

Aşağıda iki yazılı ve iki performans puanı almış bir öğrencinin puan ortalamasını hesaplayarak, dersten geçip geçmediğini belirleyen akış şeması yer almaktadır.

Siz de benzer bir problemi çözüme kavuşturacak basamakları akış şemasıyla oluşturmayı deneyiniz.



ÖRNEK ALGORİTMA VE AKIŞ ŞEMALARI

Bir asansörün çağrı gelen kata nasıl gittiğini gösteren algoritma:

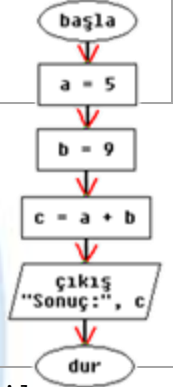
1. Başla
2. Çağrı yoksa 2. adıma git
3. Gidilecek katı oku
4. Eğer Gidilecek kat==Bulunulan kata, o zaman 6. adıma git
5. Eğer Gidilecek kat>bulunulan kattan, o zaman bir kat yukarı çık ve 4. adıma git
6. Eğer Bulunulan kat>Gidilecek kat bir kat aşağı in ve 5. adıma git
7. Kapıyı aç
8. Bitir

İsmi girilen kişiyi "Merhaba" diye selamlayan algoritma:

1. Başla
2. İsminizi giriniz, isim
3. Ekrana yaz "Merhaba "+isim
4. Bitir.

Önceden belirlenmiş iki sayının toplamını ekrana yazdıran algoritma:

1. Başla
2. a=5
3. b=9
4. c=a+b
5. Ekrana yaz c
6. Bitir

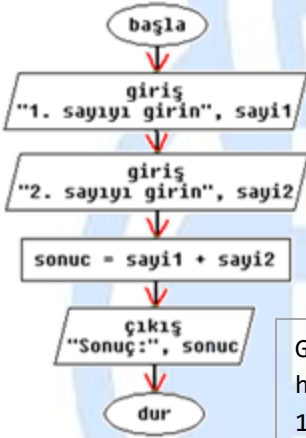


Klavyeden girilen iki sayıyı toplayıp ekrana yazdıran algoritma

1. Başla
2. Birinci sayıyı giriniz, sayi1
3. İkinci sayıyı giriniz, sayi2
4. toplam=sayi1+sayi2
5. Ekrana yaz toplam
6. Bitir

Kenar uzunlukları girilen dikdörtgenin alanını hesaplayan algoritma

1. Başla
2. Kısa kenarı gir, a
3. Uzun kenarı gir, b
4. Alan=a*b
5. Ekrana yaz, Alan
6. Bitir



Girilen 3 sayının ortalamasını hesaplayıp ekrana yazdıran algoritma

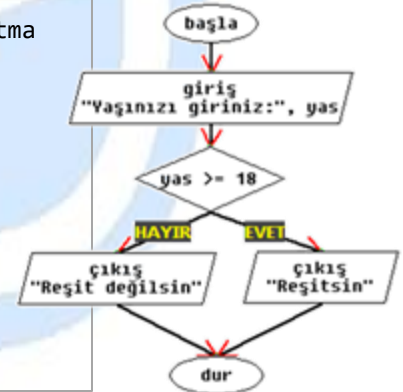
1. Başla
2. Birinci sayıyı giriniz, sayi1
3. İkinci sayıyı giriniz, sayi2
4. Üçüncü sayıyı giriniz, sayi3
5. ortalama=(sayi1+sayi2+sayi3)/3
6. Ekrana yaz, ortalama
7. Bitir

Yaşı girilen kişinin reşit olup olmadığı ekrana yazan algoritma ve akış şeması:

1. Başla
2. Yaşınızı giriniz, yas
3. Eğer yas>=18 o zaman ekrana yaz "Reşitsiniz" değilse ekrana yaz "Reşit değilsiniz"
4. Bitir

Kenar uzunlukları girilen dikdörtgenin alanını hesaplayan algoritma (kenar uzunluğu olarak 0 veya daha küçük bir sayı girilemez):

1. Başla
2. Kısa kenarı gir, a
3. Eğer a <= 0 ise Adım 2'ye git
4. Uzun kenarı gir, b
5. Eğer b <= 0 ise Adım 4'e git
6. Alan=a*b
7. Ekrana yaz, Alan
8. Bitir



Girilen sıcaklık derecesine göre suyun halini ekrana yazdıran algoritma.

1. Başla
2. Sıcaklık giriniz, sicaklik
3. Eğer sicaklik<=0 ise ekrana yaz "Katı"
4. Eğer sicaklik>0 ve sicaklik<100 ise ekrana yaz "Sıvı"
5. Eğer sicaklik>=100 ise ekrana yaz "Gaz"
6. Bitir.