

5. SINIF BİLİŞİM TEKNOLOJİLERİ VE YAZILIM ÇALIŞMA KÂĞIDI

PROGRAMLAMA İÇİN KULLANILAN KAVRAMLAR

1. Yazılım Nedir: Çeşitli görevleri gerçekleştirmek amacıyla hazırlanmış programlara **yazılım** adı verilir. Her yazılım bir **problemi** çözmek amacıyla geliştirilmiştir. **Problem**, çözülmesi gereken sorun ya da aşılması gereken engel anlamına gelir. Günlük hayatta sık sık problemlerle karşılaşırız. **Dersimizde;** günlük hayatımızda karşılaştığımız, çözüm aranması gereken ve çözümü için bilgi, mantık, deneyim ya da dikkat isteyen durumlara “**problem**” dendiğini görmüştük.

Karşılaştığınız bir problemi çözmek için ne yaparsınız?

1. Problemi iyi anlamak
2. Kısa ve anlaşılır bir çözüm yolu geliştirmek
3. Sonucun doğruluğunun tekrar tekrar test etmek

2.Programlama Nedir: Problem çözmeye; problemleri formüle edebilme, farklı ve yaratıcı çözüm yolları önerebilme, çözümü kesin ve doğru biçimde ifade edebilme becerisidir. Bilgi işlemsel düşünme aşağıdaki özellikleri barındıran bir problem çözmeye sürecidir.

- Problemleri bilgisayar veya başka araçlar yardımı ile çözebilir hâle getirme
- Mantıklı bir şekilde verileri düzenleme ve çözümleme
- Model ve benzetim desteği ile verileri sunma
- Algoritmik düşünme çerçevesinde çözümleri otomatikleştirme
- Kaynakları verimli bir şekilde kullanarak uygun çözümleri tanımlama, çözümleme ve uygulama
- Bulunan çözümü farklı problemlere transfer etme ve genelleştirme

3.Program Nedir: Program, yapılacak bir işlemi ya da hesaplamayı gerçekleştirmek için birbirini izleyen komut ya da yönergelerden oluşan yapıdır. İşlemler matematiksel ya da mantıksak olabilir.

4.Programlama Dili: Bir Problemin Algoritmik çözümünün Bilgisayara anlatılmasını sağlayan, son derece sıkı-sıkıya kuralları bulunan kurallar dizisidir.

5.Günlük Hayatta Problem Çözme: Günlük hayatımızda problemlerimizi çözmek için yaşantımızı etkileyen pek çok karar veririz. Bu kararlar yalnızca yaşantımızı etkilemekle kalmaz, bazen yaşam kalitemizi ve geleceğimizi bile etkileyebilir. Örneğin karşılaştığımız problemler, televizyonda hangi kanalı seyreteyim gibi basit de olabilir, hangi mesleği seçmeliyim gibi çok önemli de olabilir. Yanlış bir karar verilirse zaman ve kaynaklar boşa gidebilir, bu nedenle nasıl doğru karar verildiğini öğrenmek önemlidir.

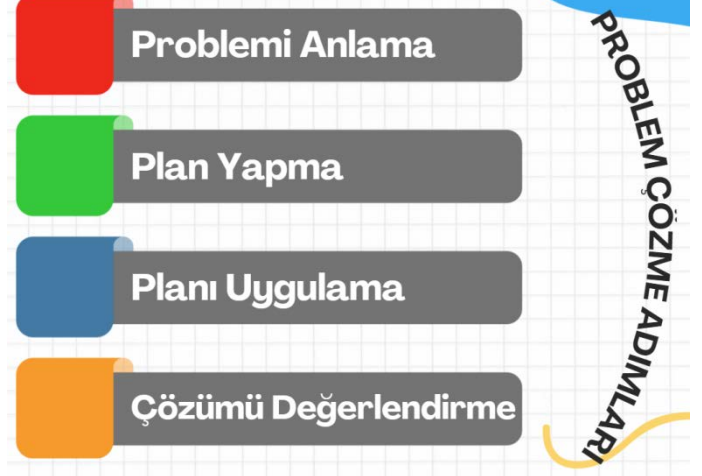
PROBLEM ÇÖZME SÜRECİ

1.Problem Çözme Teknikleri: Programlama sürecinde de problemin çözümüne yönelik yol ve yaklaşımları belirlemek gerekir ama öncelikle genel kural ve teknikleri bilmek yararlıdır.

- **Her Zaman Bir Planınız Olsun:** Belirsiz bir durumu yaşamak yerine her zaman bir planınız olmalıdır.
- **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.
- **Problemi Küçük Parçalara Ayırın:** Verilen problemi adımlara ya da bölümlere ayırarak, çözümü kolaylaştırır.
- **Önce Bildiklerinizden Yola Çıkın:** Programlama yaparken öncelikle bildiklerimiz ile başlamalı ve sonra yeni çözümler arayışına girmeliyiz.
- **Problemi Basitleştirin:** Çözmekte zorlandığınız bir problemle karşılaşırsanız problemin kapsamını daraltmayı deneyebilirsiniz.
- **Benzerlikleri Arayın:** Burada ele aldığımız benzerlik kavramı, çözülmesi istenen problemle önceden çözülen problem arasındaki olası örtüşme ya da yeni çözüme ilham verme olarak tanımlanabilir.

- **Deneme Yapın:** Bazen bir problemi çözenin en kolay yolu denemek ve sonuçlarını gözlemlemektir.
- **Asla Vazgeçmeyin:** Kararlılık, güven ve istek kaybaldığı zaman açık düşünemezsiniz, işlemler olması gerektiğinden uzun sürer ve gittikçe zorlaşır. Hatta öf ke ve kızgınlığa bile dönüşebilir. Böyle durumlarda en etkili çözüm ara vermektir.

2.Problem Çözme Adımları: Problem çözmeye sürecinde en iyi kararı verebilmek için izlenmesi gereken 4 adım vardır:



Veri Nedir: Ham yani işlenmemiş bilgi parçasına verilen addır. Veriler ölçüm, sayım, deney, gözlem ya da araştırma gibi yöntemlerle elde edilmektedir. Örneğin; havanın sıcaklığı, okul numaramız birer veridir. Birçok yönden, veriler dünyanın bir tanımı olarak düşünülebilir. Bu verileri duyanlarımızla algılarımız ve beyin bu veriyi işleyebilir. İnsanlar, dünyayı tanımak ve anlamlandırmak için verileri kullanırlar.

Bazen veriler her zaman aynı değerdendirler ve asla değişmezler. Biz bunlara **SABİT** diyoruz. Bazen de değerlerini değiştirirler bunlara da **DEĞİŞKEN** diyoruz.

Sabit Nedir: Programın her yerinde aynı değeri ifade eden değerlerdir.

Değişken Nedir: Her seferinde farklı değerler içerebilen yapıdır. Programlarda, İnternet sitelerinde boş kutucuklara girilen her bilgi değişkendir. Değişkenler, yapılan işlemler sonucu oluşturulan veya kullanıcı tarafından girilen verileri tutmak için kullanılır.

TÜR		AÇIKLAMA
SABİT	11	Her takım için sahaya çıkan oyuncu sayısı
SABİT	1	Bir takım için oyunda kaleci sayısı
SABİT	1	Sahadaki top sayısı
SABİT	4	Hakem sayısı
SABİT	Saha	Sahanın boyutları
DEĞİŞKEN	Skor	Atılan gol sayısı
DEĞİŞKEN	Seyirci	Maçı izleyenleri sayısı
DEĞİŞKEN	Takımlar	O hafta maç yapan takımların adları
DEĞİŞKEN	Faul	Verilen faul sayısı
DEĞİŞKEN		Topun bir takımda kalma oranı

OPERATÖRLER

Programlama yaparken **operatörlerden yararlandığımızı** görmüştük.

Operatör kavramı bir aracı, nesneyi ya da sayıyı işletmek/çalıştırmak anlamında kullanılır ve bilgisayar operatörü bilgisayarı işletme/çalıştırma görevini yerine getirmesine karşın, matematiksel operatörler matematiksel işlemlerin uygulanması görevini yerine getirir.

Matematiksel operatörler: Matematiksel işlemlerde bu operatörler 4 işlemi yapabileceğimiz toplama, çıkarma, çarpma, bölme ve eşittir işlemlerini gerçekleştirdiğimiz "+", "-", "x", "/", "=" sembollerle ifade edilir.

5. SINIF BİLİŞİM TEKNOLOJİLERİ VE YAZILIM ÇALIŞMA KÂĞIDI

Mantıksal Operatörler: Matematiksel operatörlerin yanında programlamada çokça kullanılan mantıksal operatörler vardır. Mantıksal operatörler “VE”, “VEYA” ve “DEĞİL”dir.

ALGORİTMA VE TÜRLERİ

Algoritma: Kodlamaya başlamadan önce oluşturacağımız yazılımın adım adım ne yapacağını tasarlamamız gerekir. İşte açık ve net ifadelerle problemin adım adım çözümünü gösteren bu taslağa **algoritma** adı verilir. Programlamanın ilk adımı algoritma oluşturmaktır.

Algoritma kurulurken dikkat edilecek hususlar şunlardır;

1. Mutlaka bir başlangıcı ve sonu olmalıdır. Sonu olmayan bir algoritmanın test edilmesinin mümkün olmadığı açıktır.
2. Bir algoritmadaki ifadeler kesin olmalıdır. Belirsizlikler algoritma ile ulaşılmak istenen sonucun elde edilememesine neden olur.
3. Algoritma sınırlı sayıda belirli kurallarla kurulmalıdır. Ve bu belirli kurallar takip edebilecekleri bir sıraya sahip olmalıdırlar.

Algoritma Özellikleri:

Tüm algoritmalar;

- ✓ **Başla** ile başlar, **Bitir** ile biter
- ✓ **Başla** ve **Bitir** arasındaki işlemler sırasıyla gerçekleşir.
- ✓ İşlem basamaklarında değişiklik yapılması bize hatalı sonuç verecek ya da sonuca ulaştırmayacaktır.

Ayrar yapma algoritması	Üç Öğrencinin yaş ortalamasını bulan ve ekrana yazdıran programın algoritması
Adım 1: Başla Adım 2: Yoğurdu kaba koy Adım 3: Su ekle Adım 4: Çırp Adım 5: Tuz koy Adım 6: Bardağa doldur Adım 7: Bitir	Adım 1: Başla Adım 2: Birinci öğrencinin yaşını gir Adım 3: İkinci öğrencinin yaşını gir Adım 4: Üçüncü öğrencinin yaşını gir Adım 5: Girilen sayıları toplayarak 3'e böl Adım 6: Sonucu ekrana yazdır Adım 7: Bitir.

Akış Şeması ile Gösterim: Bir algoritmanın görsel şekiller ve sembollerle ifade edilmiş haline “Akış Şemaları” adı verilir. Algoritma doğal dille yazıldığında herkes tarafından anlaşılabilir ya da başka anlamlar çıkarılabilir. Ancak akış çizgelerinde her bir şekil standart bir anlam taşıdığı için farklı yorumlanması mümkün değildir.

Başla ve Bitir		Akış şemasına başlarken ve bitirirken kullanılır.
Giriş İşlemleri		Kullanıcıdan veri(bilgi) almak için kullanılır. İsim, yaş, cinsiyet veya sayı gibi.
Karşılaştırma Karar Verme(Eğer)		Karşılaştırma yapmak için kullanılır. Yaşı 6'dan büyük ise, Ankara'da oturuyor ise gibi. Evet ve Hayır olmak üzere iki kola ayrılır.
Hesaplama ve Değişkene Değer Atama		Matematiksel işlemler için kullanılır. Değişkenlere değer atamak için kullanılır. İki sayıyı topla, toplamları böl, ortalama al gibi.
Yön Okları		Algoritma işlemlerinin çalışma yönünü gösterir.

DERSTE GÖRDÜĞÜMÜZ ALGORİTMA ÖRNEKLERDEN BAZILARI

BİLGİSAYARI AÇMA ALGORİTMASI

1. Başla
2. Kasadaki açma düğmesine bas.
3. Ekranı aç.
4. Bilgisayar açılana kadar bekle.
5. Bitti.

KLAVYEDEN GİRİLEN İKİ SAYININ TOPLAMINI BULAN ALGORİTMA

1. Başla
2. 1.sayıyı gir
3. 2. sayıyı gir
4. Sayıları topla
5. Sonucu ekrana yaz
6. Bitir

TRAFİK IŞIKLARINI KONTROL EDİYORSUN. YEŞİL YANIYORSA GEÇİYORSUN. KIRMIZI YANIYORSA BEKLİYORSUN. BU OLAYIN ALGORİTMASI

1. Başla
2. Trafik ışığı yeşil mi?
3. Evet ise karşıya geç. Adım 5'e git
4. Hayır ise bekle
5. Bitir.

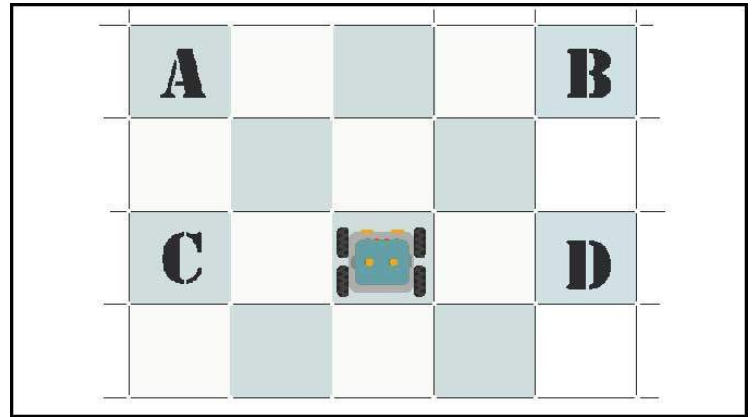
BİLGİSAYARDA OYUN AÇIP OYNAMA ALGORİTMASI

1. Başla
2. Bilgisayarı aç
3. Oyunun simgesine tıkla
4. Oyunu oyna
5. Bitir

ÖRNEK

Akış şemalarını algoritmamızın adımlarını tanımlamak için kullandık. Şimdi de bir aracı/nesneyi istediğimiz yere göndermek için nasıl bir akış şemaları hazırlayabileceğimize bakalım.

Bu aslında bilgisayarda karşılaştığımız pek çok oyun ve animasyonun da temelini oluşturuyor. **Şimdi şekilde yer alan aracın C noktasına ulaşması için algoritma ve akış şemasını oluşturalım.**



Bu akış şemasına göre C noktasına ulaşması için gerekli olan akış şemasını bir düşünelim.

Algoritma (Çözüm)

1. Başla
2. Sola dön
3. İlerle
4. İlerle
5. Bitir

Akış Şeması (Çözüm)



Not: Ayrıca derste görmüş oynanmış olduğunuz, EBA Labirent, EBA Cody gibi oyunlardan sorumluyuz.